Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2020, vol. 25, no. 2, pp. 87–104
https://doi.org/10.2478/acss-2020-0011
https://content.sciendo.com

A Systematic Comparative Analysis of Clustering Techniques

Satinder Bal Gupta[1*], Rajkumar Yadav[2], Shivani Gupta[3]
[1–3]*Indira Gandhi University, Meerpur, Rewari, India*

Abstract **– Clustering has now become a very important tool to manage the data in many areas such as pattern recognition, machine learning, information retrieval etc. The database is increasing day by day and thus it is required to maintain the data in such a manner that useful information can easily be extracted and used accordingly. In this process, clustering plays an important role as it forms clusters of the data on the basis of similarity in data. There are more than hundred clustering methods and algorithms that can be used for mining the data but all these algorithms do not provide models for their clusters and thus it becomes difficult to categorise all of them. This paper describes the most commonly used and popular clustering techniques and also compares them on the basis of their merits, demerits and time complexity.**

Keywords **– Clustering, *c*-means, data mining, fuzzy, *k*-means, partitioning.**

I. Introduction

Clustering can be described as a method that helps to handle the colossal amount of data stored in the database in a very effective manner [1]. It is a technique based on unsupervised learning method that helps in representing the data accurately by transforming it in a compact form. This is done by separating the data into different classes or categories which are called clusters. Different groups are made that separate a similar type of data from dissimilar data. The data present in one cluster are of the same type [2]. The similarity between different data points is found on the basis of distance measures. Clustering techniques used to separate the data items can be one of the two types: *Hard Clustering*: It is a type of clustering technique in which data items are divided into clusters, but a data item can be a part of only one cluster. The overlapping of clusters is not allowed in this technique. *Fuzzy Clustering*: It is also known as soft clustering because this technique allows overlapping of clusters which means that a data item can be a part of more than one cluster and this is done on the basis of similarity between the data items [3]. Partitional and hierarchical clustering algorithms belong to hard type of clustering technique. In partitioned clustering type, the clustered dataset results in discrete partitions. While in case of hierarchical clustering algorithms, the clustered datasets result in a tree-like structure. Hard clustering can be applied to larger datasets and is a simple technique with only two probabilities of either 0 or 1. However, this technique sometimes gets trapped in local optima and is sensitive to outliers. It is not efficient and does not provide accurate results in case of high dimensionality datasets. Thus, in such cases fuzzy clustering is used as it provides accurate results and its efficiency is also very good. It also handles noisy data, and computation becomes easy with the use of this type of clustering. For dividing the data into different clusters, distance is calculated and this depends on the type of variables present in the dataset, such as Euclidean distance, Manhattan distance, Minkowski distance, Hamming distance and so on. Euclidean distance helps in finding the distance between numeric type variables. Manhattan or Minkowski distance is used to find out the value of summation of difference in the coordinates of data points. Hamming distance finds distance between non-numerical data values. Different clustering algorithms are used for mining the data where a large amount of data is carefully examined to extract useful information so that it can help users in making decisions. The main purpose of clustering is to explore the data. The clustering techniques can be very effectively used in statistical data analysis in different fields such as machine learning, pattern recognition, education, bio-informatics and so on. Different clustering algorithms help in grouping the data points with similar patterns into clusters so that the data points belonging to one cluster are almost similar to each other. In case of the educational sector, clustering algorithms help the teachers to divide the students into clusters on the basis of their abilities and taking into consideration the course material. It helps the teachers to know about the difficulties of students so as to improve their skills by making changes in the teaching methods. The clustering algorithm when applied to a large amount of data helps to find out the hidden and unidentified patterns in the form of classes or groups called clusters. This paper focuses on describing in detail different clustering techniques used to group the data along with their merits, demerits and complexities. Different applications which make use of these clustering techniques in order to organise the data are also discussed [4].

II. Literature Review

A lot of work has already been done in this field by hundreds of researchers. Some of the work has been reviewed in this section by the authors.

[*] Corresponding author's email: satinderbal@igu.ac.in

Kaufman [1] did a study on some fuzzy clustering algorithms such as PAM, CLARA and agglomerative and divisive methods and others with their algorithms and programs. The purpose of fuzzy analysis was discussed which included divisive analysis, monothetic analysis etc. These techniques were found to be robust and consistent.

Fahad [2] discussed various clustering methods such as partitioning method grid-based method with its different algorithms. The conclusion made was that there did not exist any single algorithm that fulfilled all the needs and passed the evaluation criteria. All the discussed clustering algorithms had a stability problem. Therefore, it was suggested to develop a clustering algorithm that could solve these problems and the drawbacks of the discussed techniques should be removed.

Jain *et al.* [3] considered various techniques used for data mining that could be helpful in extracting useful information from the database and could be used in making useful decisions. They also analysed fuzzy clustering technique used to make clusters of the data on the basis of similarity. They applied *k*-means clustering algorithm to the collected dataset and found out that the algorithm was simple to implement and was computationally good due to the linear time complexity. It was concluded that clustering techniques could be used in different applications, including image segmentation, retrieval of information, object recognition and so on.

Khaing [4] discussed various clustering algorithms, which could help in the processing of big data along with their complexities. It was concluded that BIRCH, CLIQUE algorithms were effective algorithms for removing outliers. Arbitrary shaped clusters could be obtained using CURE and ROCK algorithms and non-convex shaped clusters could be obtained using a model-based method.

Bezdek [5] evaluated an FCM algorithm. This algorithm could be applied to analyse the problems of geostatistical data. The paper discussed the FORTRAN-IV coding of FCM algorithm, which could be used for generating fuzzy partitions for the numerical dataset. In order to control the sensitivity to noise, an adjustable weighing factor was used.

Cannon *et al.* [6] compared two versions of FCM algorithms: LFCM and AFCM. The approximate FCM was proposed for the computation of Euclidean distances and exponentiation. AFCM could be used to increase the processing of FCM. They compared LFCM (literal coding of FCM algorithm) and AFCM (table-driven approach) and found out that about 1/6 less computer time was required by AFCM to get accurate results as compared to LFCM. AFCM run faster than LFCM.

Hung *et al.* [7] presented a *k*-means clustering algorithm in which convergence could be achieved by doing expensive computations related to distance of centroids. It was concluded that the algorithm could converge easily. The clustering algorithm presented took less time in processing than the original *k*-means algorithm.

Bezdek [8] modified the FCM algorithm proposed by Dunn in 1973. This algorithm could be effectively used in making clusters of the data and in this process a data point having membership values was assigned to a cluster. It was found out that this algorithm was useful for pattern recognition.

Gustafson *et al.* [9] proposed Gustafson and Kessel algorithm, which was an extension of FCM algorithm. They used fuzzy covariance for making clusters of the data and found out that the use of fuzzy covariance helped to produce more accurate clusters.

Oyelade [10] used a *k*-means clustering algorithm so as to forecast the performance of students at higher education institutions. The Euclidean distance method was used for finding out the similarity between the collected data. The students' results were analysed to make the prediction. It was concluded that using this algorithm with a deterministic model was of great help to monitor and enhance the semester performance of students. It also helped to improve the decisions made by the academic planners in monitoring the performance of the candidates in different semesters so as to improve the academic results in the future.

Jumaa *et al.* [11] discussed techniques used to protect the sensitive knowledge while applying clustering data mining algorithms to the datasets, such as additive noise technique, data swapping technique, data copying technique. They also proposed a system for hiding the sensitive clusters, in which adaptive noise was added to the original database. The clustering algorithm used was *k*-means clustering. JAVA programming language and WEKA tool were used in the proposed algorithm. It was concluded that the new proposed algorithm could protect the sensitive clusters with a low information loss ratio and high privacy ratio.

Atiyah *et al.* [12] proposed a *KC*-means algorithm by combining two algorithms – *K*-means and *C*-means clustering algorithms during two different stages. The *K*-means algorithm was applied to the dataset at the first stage to find the centres of groups and *C*-means algorithm was applied during the second stage to the centres obtained previously. It was concluded that the algorithm formed was more accurate and clusters formed using this algorithm were found to be more precise. The clusters were found to be more distinct.

Kaufman [13] used a *k*-medoid modelling technique in order to divide the data points into clusters. The method was found to be more robust and based on partitioning around medoids method for making clusters. It was found out that the process of clustering was not affected by the order in which the objects were presented. This algorithm could minimise the average dissimilarity of the objects present in the dataset to the nearest medoid.

III. Working Process of Cluster Formation and Evaluation

The process of clustering involves dividing the data point stored in the database in the form of clusters. In fuzzy clustering, each data item is associated with a membership value that helps to determine the strength of the data item. The data items with similar strength are grouped together in a cluster. After the formation of clusters, these clusters get analysed for the purpose of extracting important and useful information from them. Fig. 1 shows the formation and evaluation of clusters.

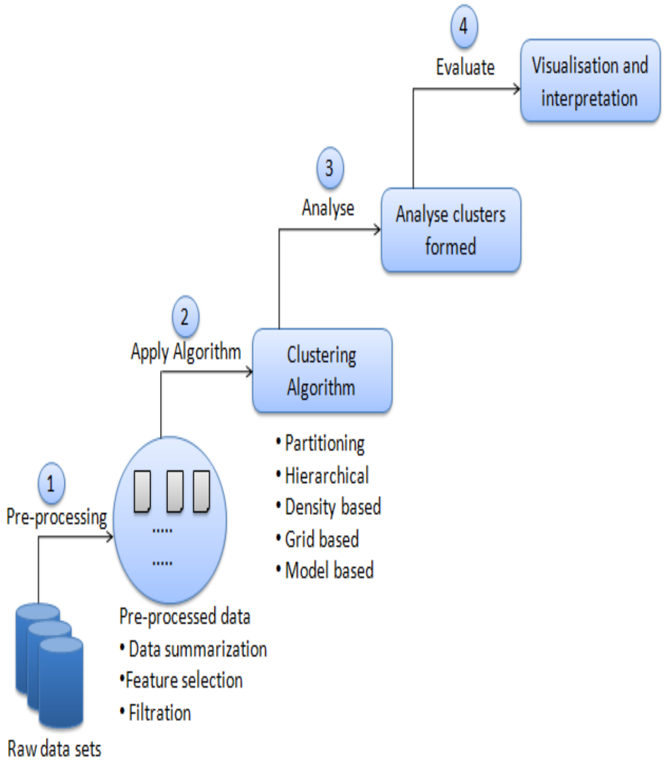

Fig.1. Formation and evaluation of clusters [3].

Fig. 1 (adapted from [3]) shows the process through which clusters are formed and evaluated in order to get the required result. The steps involved are as follows:

Step 1. *Data Pre-Processing:* Firstly, the data are collected and then processing of the data is done, which involves summarisation of data, feature selection and filtration of data. Data summarisation means to store the data in a way that reduces the size of data and represents the data in a concise form. Feature selection involves selecting the features from the dataset to which clustering algorithms are to be applied. Filtration involves application of filters to the data in order to reduce the differences in the attributes.

Step 2. *Application of Algorithms:* Different clustering techniques are discussed later that are used for dividing the data points into cluster.

Step 3. *Analysis:* The clusters thus formed are analysed to get the required information from the dataset.

Step 4. *Evaluation:* After this evaluation of the result obtained is performed, the data can be represented using visualization tools and interpretation can be done easily.

IV. Clustering Methods and Techniques

There are different types of clustering techniques on the basis of cluster formation methods [2] (see Fig. 2).

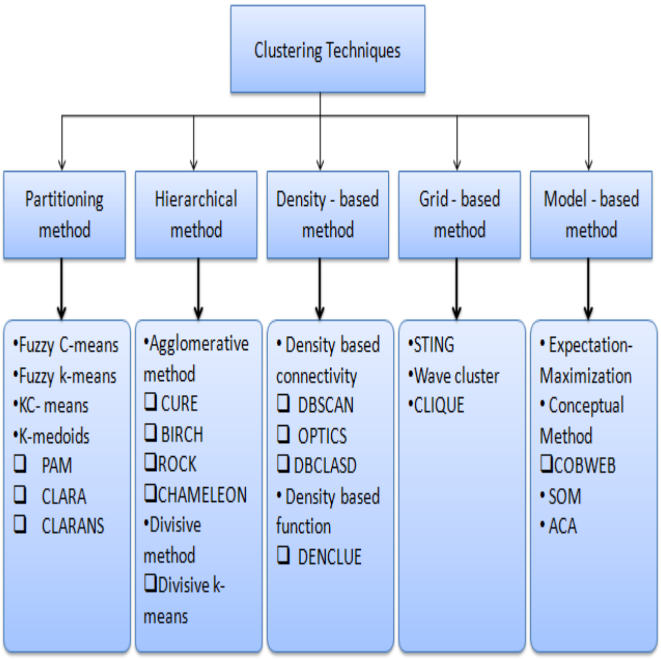

Fig. 2. Various clustering techniques.

Different clustering methods [2], [4] (as shown in Fig. 2) can be categorised into five types, namely:

(i) partitioning method (on the basis of centroids);
(ii) hierarchical method (on the basis of connectivity);
(iii) density-based method (on the basis of density);
(iv) grid-based method (on the basis of grid structure);
(v) model-based method (on the basis of distribution).

All of these clustering methods and various techniques used in them are briefly explained in the following sections.

A. Partitioning Method

This method divides the data points present in the database into *n* number of partitions where each partition can be called a cluster. All the data points present in one cluster have some similarity. This method involves the following four algorithms:

1) Fuzzy C-Means Algorithm

In this technique [5], datasets are divided into *c* clusters. FCM algorithm is based on finding the centroid of clusters and the distance of data points from the different centroids so that the cluster with the closest distance can be allocated to the data point [6]. This distance is known as Euclidean distance [7], [8]. The FCM algorithm can be shown as follows. Fig. 3 shows the steps required to perform the FCM algorithm:

Step 1. The data items are assigned the membership values in the range [0,1]. μ_{jk} parameter is used to assign cluster *j* to the element x_k based on the membership value of the element.

Here,

$\mu_{jk} = 0$ represents non-membership;

$\mu_{jk} = 1$ represents full membership;

$0 < \mu_{jk} < 1$ represents partial membership.

There is a condition that while assigning the membership values the addition of membership values of all the data points must be equal to 1, i.e.,

$$\sum_{i=1}^{k} \mu_{ij} = 1 \text{ , for all } j = 1 \ldots n.$$

Step 2. The cluster centres are found out by using the formula:

$$C_j = \frac{\sum_k [\mu_j(x_k)]^m x_k}{\sum_k [\mu_j(x_k)]^m},$$

where x_k = kth data element;

C_j = the centre of cluster j;

m = the fuzzification parameter;

μ_j = the membership value.

The fuzzification parameter is used to find out the degree of fuzziness in the clusters. The value ranges from 1 to n. The low value of m means the element belongs to only one cluster and a larger value of m means that the data element belongs to more clusters and there is cluster overlapping.

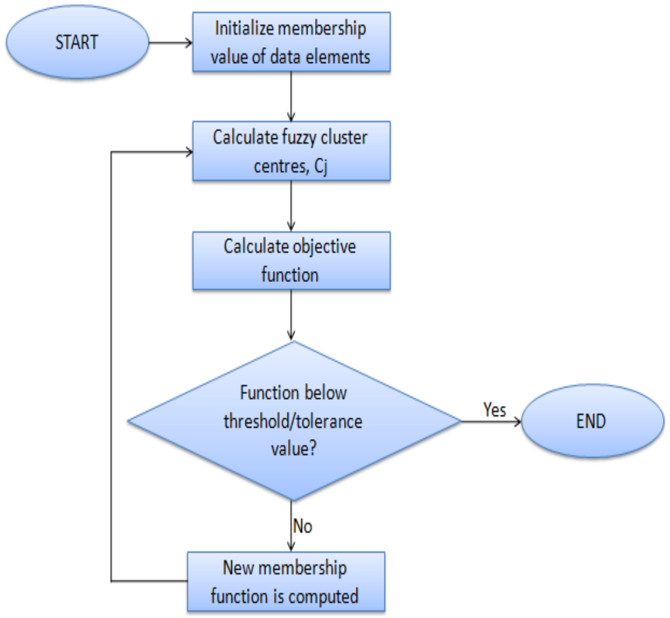
START
Initialize membership value of data elements
Calculate fuzzy cluster centres, Cj
Calculate objective function
Function below threshold/tolerance value?
Yes
END
No
New membership function is computed

Fig. 3. FCM algorithm.

Step 3. The objective function that needs to be minimised is:

$$J_{FCM}(U,V) = \left\{ \sum_{j=1}^{l} \sum_{k=1}^{n} (\mu_{jk})^m \left\| x_k - c_j \right\|^2 \right\},$$

where μ_{jk} belongs to 0 to 1 and $\sum_{j=1}^{l} \mu_{jk} = 1$, for all values of k

c_j is the centre of cluster.

Step 4. The membership values of data elements are calculated as follows:

$$\mu_{jk} = \frac{\left(\frac{1}{d_{jk}}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{p} \left(\frac{1}{d_{ik}}\right)^{\frac{1}{m-1}}},$$

where μ_{jk} = the membership of x_k in the jth cluster;

d_{jk} = the distance of x_k from the jth cluster;

m = the parameter for fuzzification;

p = number of clusters;

d_{ik} = distance of x_k from cluster i.

Step 5. Termination criteria: The iteration stops if

$$\left\| U^{k+1} - U^k \right\| < \epsilon,$$

where ϵ belongs to 0 to 1 and k = iteration steps.

Extensions to Fuzzy *C*-Means Algorithm:

a) Gustafson Kessel Fuzzy Clustering Algorithm (GKFCA)

The GKFCA is an extension to an FCM algorithm with a small variation. The clusters can have different ellipsoidal shapes [9]. In this algorithm, the objective function used is:

$$J_{GKFCM}(U,V) = \left\{ \sum_{j=1}^{l} \sum_{k=1}^{n} (\mu_{jk})^m \left\| x_k - c_j \right\|^2 A_j \right\},$$

where A_j = the symmetric matrix and $\|A_j\|$ = the constant.

b) Fuzzy C-Elliptotypes Clustering

Clusters can have a shape of line/plane. The Euclidean distance is calculated taking into consideration the eigen vector directions [8]. The distance calculated is a combination of two distance measures:

$$d^2(x_i, P_j) = \alpha \cdot d^2 V_{ji} + (1-\alpha) d^2 E_{ji},$$

$d^2(x_i, P_j) = \alpha \cdot d^2 V_{ji} + (1-\alpha) d^2 E_{ji}$ where d^2E_{ji} = the Euclidean distance, and

$$d^2 V_{ji} = \left\| x_j - p_j \right\|^2 - \sum_{k=1}^{r} \left((x_i - p_j) \cdot e_{jk} \right),$$

were e_{jk} = kth eigen vector of matrix C_j of cluster j.

2) Fuzzy K-means Algorithm

This algorithm is used to group n different data points into k clusters chosen randomly [10].

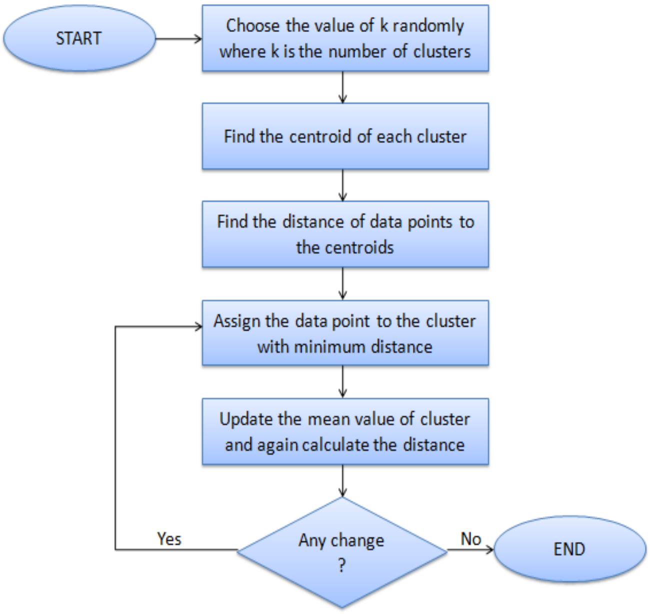
START
Choose the value of k randomly where k is the number of clusters
Find the centroid of each cluster
Find the distance of data points to the centroids
Assign the data point to the cluster with minimum distance
Update the mean value of cluster and again calculate the distance
Any change ?
Yes
No
END

Fig. 4. Fuzzy *k*-means algorithm.

The objective function used in this algorithm is:

$$E = \sum_{j=1}^{l} \sum_{k=1}^{n} \left\| x_{ik} - c_j \right\|^2 ,$$

where n = different data points present in cluster i;
x_{ik} = k^{th} data point of cluster i;
c_j = centre of cluster j.

This function is called the sum of squares error, which finds the distance to the nearest cluster. The steps of this algorithm are shown in Fig. 4. On the basis of the distance of data points from the centroid of cluster, the data points are assigned to different clusters [11].

3) Fuzzy KC-Means Algorithm

It is a combination of *K*-Means and *C*-Means algorithms [12]. The steps of this algorithm are shown in Fig. 5.

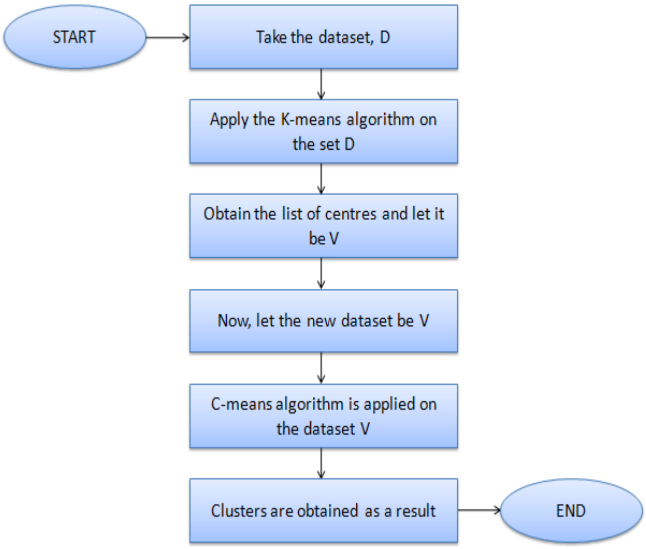
START
Take the dataset, D
Apply the K-means algorithm on the set D
Obtain the list of centres and let it be V
Now, let the new dataset be V
C-means algorithm is applied on the dataset V
Clusters are obtained as a result
END

Fig. 5. KCM algorithm.

In KCM algorithm, firstly *k*-means algorithm is applied to the initial data set and then the centre of different clusters is found out. Then, a *c*-means algorithm is applied to find the groups of dataset in the form of clusters.

4) K-Medoids Algorithm

In this algorithm, a medoid value is used to group data points instead of a mean value [13]. Medoid is considered to be the object that is located almost at the centre of the cluster [14]. Fig. 6 shows the algorithm steps, in which the input is *k* and *d*, where *k* denotes the number of clusters and *d* denotes the complete dataset having *n* different objects. Every cluster has a centred object in it. The distance of other point is found out with that centred object and the smallest distance point is made to be the centroid of cluster.

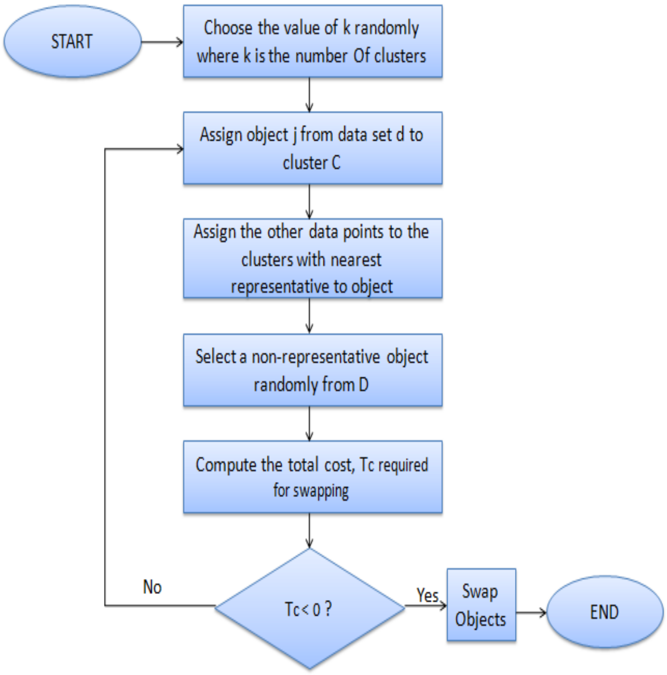
START
Choose the value of k randomly where k is the number Of clusters
Assign object j from data set d to cluster C
Assign the other data points to the clusters with nearest representative to object
Select a non-representative object randomly from D
Compute the total cost, Tc required for swapping
Tc < 0 ?
No
Yes
Swap Objects
END

Fig. 6. *K*-medoids algorithm.

Various types of k-medoids clustering are PAM, CLARA and CLARANS. These are discussed briefly below.

a) PAM (Partitioning Around Medoids)

The PAM technique is much alike to a *k*-medoids algorithm. It consists of two algorithms – BUILD and SWAP so as to obtain the optimal result [14]. The PAM algorithm comprises two different phases:

BUILD: In this phase, initially a set is formed by selecting some objects from the dataset.

SWAP: In this phase, the selected objects are swapped with unselected objects and this helps enhance the quality of clusters.

In this algorithm, initially *k* points are selected randomly as the medoids from *n* data points that are stored in the dataset. Then, each data point is associated with the closest medoid. After this process, swapping takes place and the cost of swapping is computed. Then, the minimum swapping cost is chosen. The above process is repeated till no change occurs in the medoid.

b) CLARA (Clustering Large Applications)

It is an extension of PAM. It divides the sample of data points into sub-samples and then applies PAM repeatedly to the sub-samples.

Fig. 7 shows the algorithm, in which firstly multiple subsets of the same size are chosen from the dataset collected. Then, the PAM algorithm is applied to each subset and medoid of each subset is chosen. After this procedure, the data points are assigned to the clusters by choosing the closest medoid. Finally, the sum of dissimilarities of observations is calculated so as to measure the goodness of clustering [15].

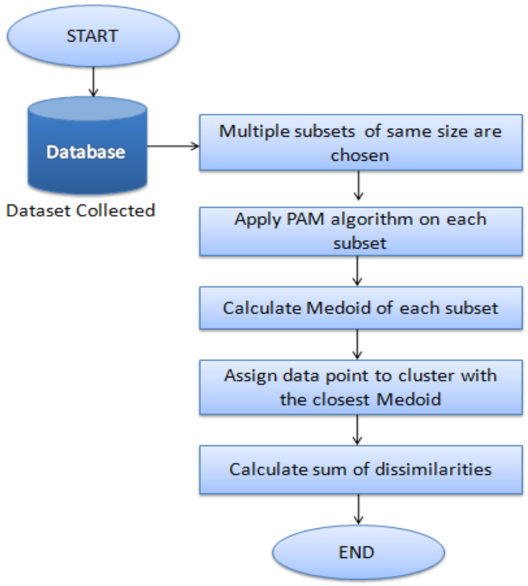
START
Database
Dataset Collected
Multiple subsets of same size are chosen
Apply PAM algorithm on each subset
Calculate Medoid of each subset
Assign data point to cluster with the closest Medoid
Calculate sum of dissimilarities
END

Fig. 7. CLARA algorithm.

c) CLARANS (Clustering Large Applications on the basis of Randomized Search)

It is a combination of PAM and sampling technique. It is used for huge datasets. In this algorithm, samples are drawn randomly while searching in every step. Less distance between the neighbouring nodes results in the increasing efficiency [16], [17]. In this algorithm, two parameters are used –maxneighbour and numlocal. The maxneighbour indicates the highest number of neighbours that are explored and the numlocal indicates the number of local minima that is obtained. The nodes with lower cost are found and compared with the current node and then the node with lower cost is declared as a minimum local value. This process keeps on repeating till the numlocal has been found.

B. Hierarchical Method

It is a type of clustering method that helps divide the dataset into clusters in the form of a hierarchy. The hierarchy of clusters so formed is called as dendrogram. The clusters are formed using either of the two approaches: top-down or bottom-up approach [18]. It is of two types:

1) Agglomerative Method

It uses a bottom-up approach. It is also known as AGNIS, i.e., Agglomerative Analysis.

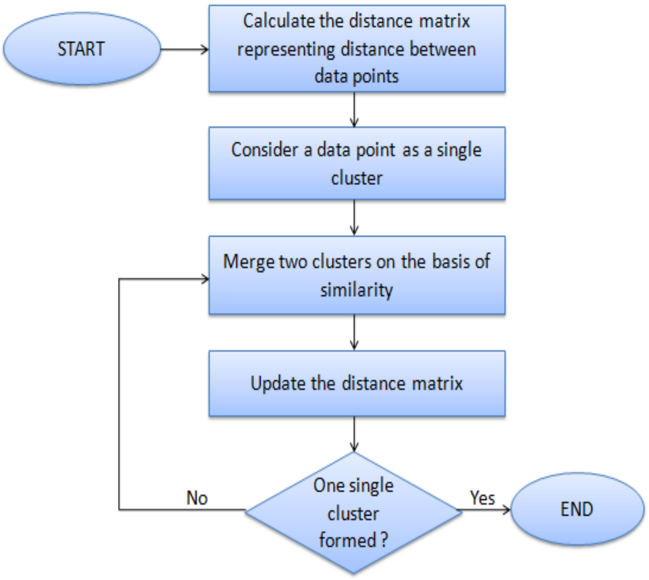
START
Calculate the distance matrix representing distance between data points
Consider a data point as a single cluster
Merge two clusters on the basis of similarity
Update the distance matrix
One single cluster formed ?
No
Yes
END

Fig. 8. Agglomerative hierarchical clustering algorithm.

Fig. 8 shows the steps of a basic agglomerative hierarchical algorithm in which, firstly, a distance matrix of the data points is calculated. Then, a data point present in the dataset is taken to be an individual cluster. The two individual clusters are amalgamated to form one single cluster and this process keeps on repeating till the termination condition is reached, i.e., *k* number of clusters are formed [19]. The following three algorithms: CURE, BIRCH, CHAMELEON used this hierarchical method to form clusters. These are explained below:

a) CURE (Clustering Using REpresentative)

Fig. 9 shows the steps required to divide the dataset into clusters using a CURE algorithm. Firstly, the samples are drawn randomly from the dataset. Secondly, the selected sample is partitioned. Thirdly, clustering is applied to the partitions. Fourthly, outliers are eliminated and the clustering process is applied again. Finally, the labels are allocated to the different data points. This process of making samples in Step 2 and partitioning of the samples in Step 3 help speed up the process of making clusters [20].

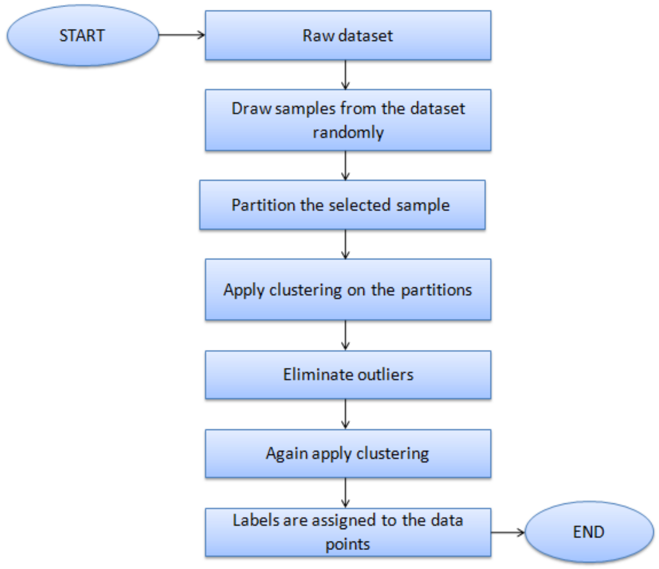
START
Raw dataset
Draw samples from the dataset randomly
Partition the selected sample
Apply clustering on the partitions
Eliminate outliers
Again apply clustering
Labels are assigned to the data points
END

Fig. 9. CURE algorithm.

b) BIRCH (Balanced Iterative Reducing and Clustering using Hierarchy)

The steps involved in this algorithm are as follows:

Step 1. A CF (Clustering Feature) tree is constructed by examining the whole database. It is a height balanced tree that is used to calculate the centroid and distances required to merge the clusters. It is a memory-based algorithm that stores the required information.

Step 2. In this step, the leaf nodes of the tree are divided by applying the clustering algorithm [21].

c) ROCK (Robust Clustering Using Links)

This algorithm makes use of a link strategy to form clusters of data points. The bottom-up approach is used to combine the links and form clusters. The algorithm can be shown as follows:

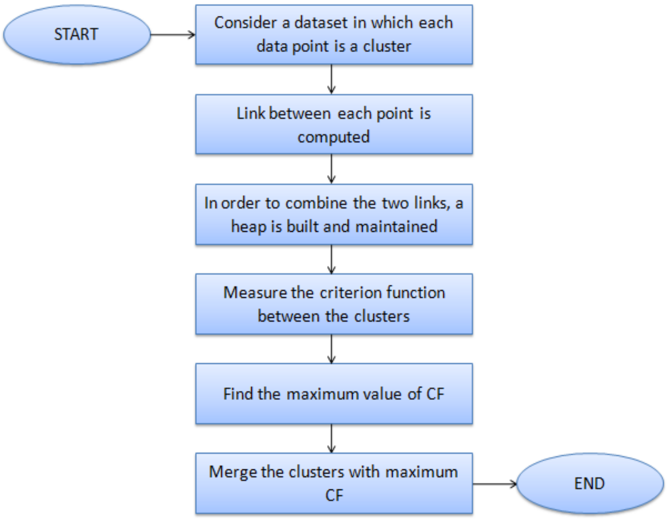
START
Consider a dataset in which each data point is a cluster
Link between each point is computed
In order to combine the two links, a heap is built and maintained
Measure the criterion function between the clusters
Find the maximum value of CF
Merge the clusters with maximum CF
END

Fig. 10. ROCK algorithm.

Fig. 10 shows the steps of ROCK algorithm in which, firstly, every data point present in the dataset is taken as a different cluster. Then, link between each of the two points is computed. Then, a heap is built in order to combine the links. The criterion function is measured between the clusters, which help merge the clusters. The clusters with a maximum value of CF (Criterion Function) are merged [22], [23].

d) CHAMELEON

This algorithm uses a sparse graph consisting of nodes and edges. Nodes denote the data points present in the dataset and edges represent the similarity between the data points [18], [19].

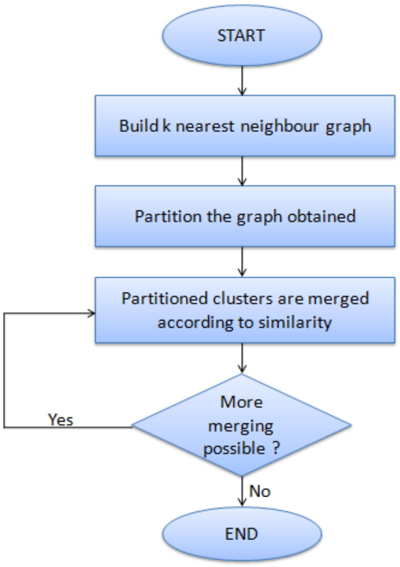
START
Build k nearest neighbour graph
Partition the graph obtained
Partitioned clusters are merged according to similarity
Yes
More merging possible ?
No
END

Fig. 11. Chameleon algorithm.

Fig. 11 explains the steps included in this algorithm, in which the graph is built from the dataset and the partitioning is performed on the graph. Firstly, the *k* nearest neighbour graph is built from the dataset collected. Then, the graph is partitioned. After this procedure the partitioned clusters are merged on the basis of similarity and this process keeps on repeating till the merging is possible [24], [25].

2) Divisive Method

It uses a top-down approach. It is also called DIANA, i.e., Divisive Analysis.

Fig. 12 shows the basic divisive hierarchical algorithm in which, firstly, the complete dataset is taken to be a single cluster. The cluster is divided into sub-clusters and this process is repeated till *k* numbers of possible clusters are formed [26]. It includes the following algorithm:

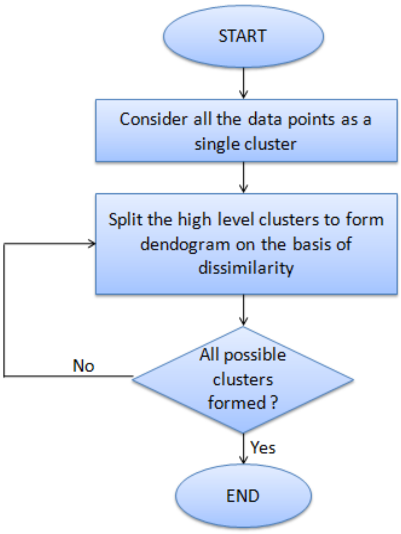
START
Consider all the data points as a single cluster
Split the high level clusters to form dendogram on the basis of dissimilarity
All possible clusters formed ?
No
Yes
END

Fig. 12. Divisive hierarchical algorithm.

a) Divisive Hierarchical K-Means Algorithm

This algorithm helps to find *k* number of clusters. Fig. 13 shows the extended version of *k*-means algorithm in which, firstly, the dataset *D* and the *k* clusters are taken and the whole dataset is considered to be one single cluster and the value of variable storing the number of clusters formed, i.e., clusterNumber is initialised to 1. Then, the two data points are selected randomly from the dataset say, *x* and *y*. Then, *k*-means algorithm is applied in order to get two clusters. The value of clusterNumber is updated to 2. Now, the value of *k* is compared to the clusterNumber value and if clusterNumber < *k,* then SSE (Sum of Squared Error) is calculated. SSE is used to find out which cluster should be selected for division [27]. The cluster with the maximum value of SSE is picked for bisection.

The SSE of clusters is calculated as follows:

$$SSE = \sum_{j=1}^{p} \left(x_j - m\right)^2 ,$$

where *m* represents a mean that is calculated using the formula:

$$m = \frac{\sum_{i=1}^{p} x_i}{p},$$

where *p* represents the number of elements present in the dataset.

This process is repeated till *k* clusters are formed [28].

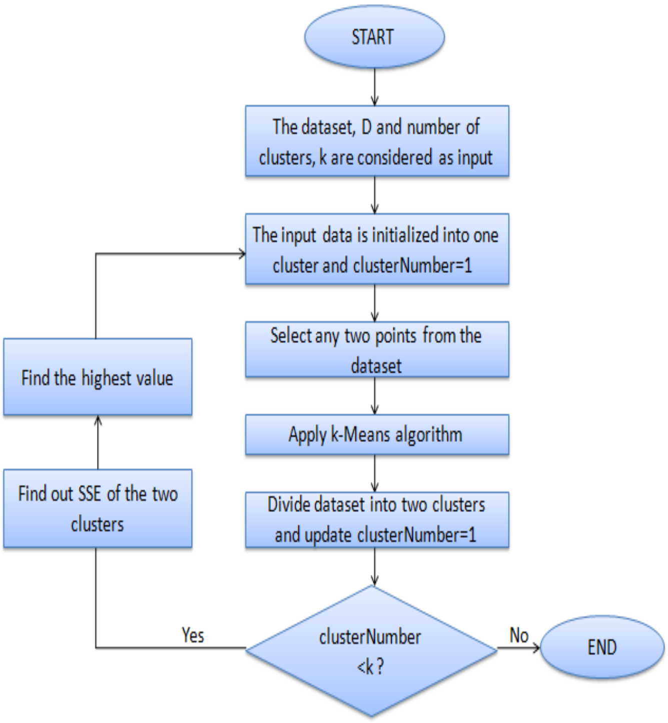
START
The dataset, D and number of clusters, k are considered as input
The input data is initialized into one cluster and clusterNumber=1
Select any two points from the dataset
Apply k-Means algorithm
Divide dataset into two clusters and update clusterNumber=1
clusterNumber <k ?
Yes
No
END
Find out SSE of the two clusters
Find the highest value

Fig. 13. Divisive *k*-means algorithm.

C. Density-Based Method

This method is applied to each data point stored in the initial dataset. In this algorithm, the density of data points is used for finding the clusters [29]. Clusters formed are considered high density regions separated from the other data members, which are considered to be low density points [20]. It is further categorised into the following two sub-parts:

1) Density-Based Connectivity

This method measures both the density and the connectivity of data points and this is done according to the local distribution of neighbour data points [30]. It includes the following algorithms:

a) DBSCAN (Density Based Spatial Clustering of Application with Noise)

The DBSCAN helps find clusters with different shapes, such as 'S' shape and oval shape [31]. There is no need to define the number of clusters in the beginning, as clusters are formed according to the density of data points. The data points are separated into three parts:

Core Points: These refer to the points that are present inside the cluster.

Border Points: These refer to the points that are present in the neighbourhood of the core points.

Noise Points: These points refer to all the other points except core and border points [32].

Fig. 14 shows the steps of DBSCAN algorithm in which, firstly, the graph is formed with the help of data points present in the database. Then, an edge is created from point *a* to another point *b* present in the neighbourhood. After this procedure, if there are no more core points from a node, then the process is terminated; otherwise, a node that can be reached from the core point is selected. The above process is repeated till a cluster is formed from the core points [33].

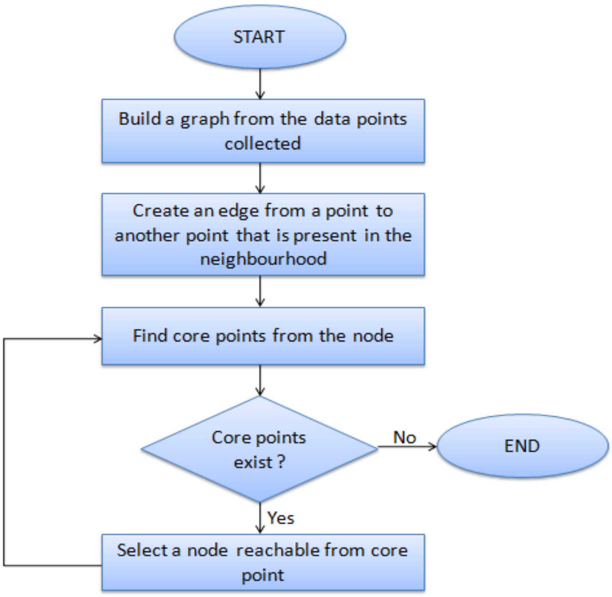
START
Build a graph from the data points collected
Create an edge from a point to another point that is present in the neighbourhood
Find core points from the node
Core points exist ?
No
END
Yes
Select a node reachable from core point

Fig. 14. DBSCAN algorithm.

b) OPTICS (Ordering Point to Identify the Cluster Structure)

It is an extended version of DBSCAN algorithm. It is based on the notion of density-based cluster ordering. Data points are divided into high density and low density clusters. The priority is given to high density clusters [34]. It stores two types of information about the data points present in the clusters formed: core distance and reachability distance [35].

Core distance: This refers to the minimum value of the radius that is needed for classifying the given point as a core point.

Reachability distance: The reachability distance of a data point *n* from any other data point *m* can be defined as the smallest distance from data point *n*, where *n* is the core point. The reachability distance is always greater than or equal to the core distance and it cannot be smaller than the core distance.

In this algorithm, a point is chosen from the dataset and its neighbours are found. The core distance of the data point is found and the reachability distance of all the neighbours is computed from the core point. If the reachability distance of the point is undefined, then it is replaced with the computed distance. If the reachability distance for that point already exists, then its value is compared with the newly computed reachability distance and if the new distance obtained is less than the old distance, then the value is replaced. The above steps are repeated for all other data points until clusters are formed.

c) DBCLASD (Distribution-Based Clustering of Large Spatial Databases)

The DBCLASD is based on the connectivity method. It does not require input parameters as they are determined automatically. It uses a dynamic approach in determining the shape of clusters. It begins with a single cluster initially and then adds neighbouring points to the cluster if the point fulfils the condition of the expected distribution distance [36].

Fig. 15 shows the DBCLASD algorithm in which, firstly, the set *C* of candidates is formed that is based on the query. After this process, the Expected Distribution Condition (EDC) of set *C* is checked which decides whether a point will be added to the list of unsuccessful candidates or will remain in the cluster. The EDC represents the nearest neighbour distance of data point *x* represented as NNdist(*x*). NNdist(*x*) has an expected distribution along with some confidence level. The points present in cluster *C* are connected in the form of grid-like structure.

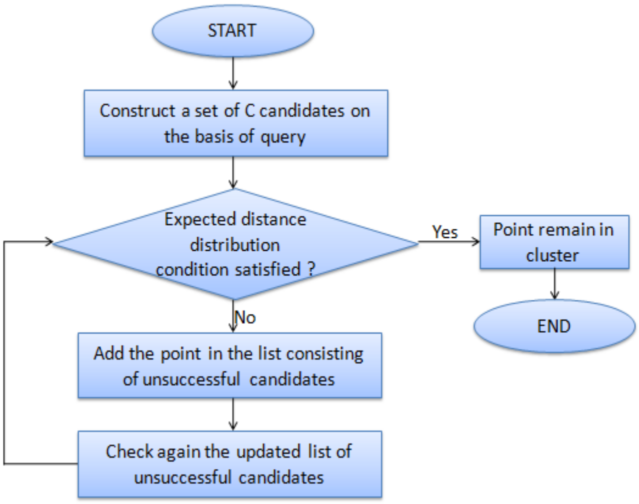
START
Construct a set of C candidates on the basis of query
Expected distance distribution condition satisfied ?
Yes
Point remain in cluster
END
No
Add the point in the list consisting of unsuccessful candidates
Check again the updated list of unsuccessful candidates

Fig. 15. DBCLASD algorithm.

2) Density-Based Functions

The density function of a data point represents the summation of all the influence functions calculated for each of the data point in the database. If there are *n* data points, then the density function of a data point *y* can be calculated as follows:

$$f_{\mathrm{DF}}(y)=\sum_{j=1}^{n} f_{\mathrm{DF}}^{y_j}(y)=f_{\mathrm{DF}}^{y_1}(y)+f_{\mathrm{DF}}^{y_2}(y)+\ldots+f_{\mathrm{DF}}^{y_n}(y),$$

where *y* is the data point, DF is the density function.

It includes the following algorithm:

a) DENCLUE (Density-Based Clustering)

It uses the concept of density function and influence of data point. The impact of data points on their neighbourhood points is calculated using the influence function [37]. The density function finds out the summation of the influence of each and every data point [38]. This algorithm mainly comprises two steps: The first step involves representing the data points available in the database in the form of a hyper-rectangle structure, which helps in speeding up the computation of density function. In the second step, the clusters are identified by connecting highly populated cells to the populated cells in their neighbourhood.

Fig. 16 shows the detailed algorithm in which, firstly, the dataset is presented in the form of a hyper-rectangle like structure. Then, the highest density cell is found on the basis of a mean value. This is done by calculating influence between the points. The aggregate sum of these influence points is the density function. It can be calculated with the help of Gaussian function. The Gaussian function between two points x_1, x_2 can be applied as follows:

$$f_{\text{Gauss}}\left(x_1, x_2\right) = e^{-\frac{d(x_1,x_2)^2}{2\sigma^2}},$$

where $d(x_1, x_2)$ represents the Euclidean distance between the two given data points, σ represents the radius of the neighbouring structure, which contains x_1.

The density function is then applied to the data points as follows:

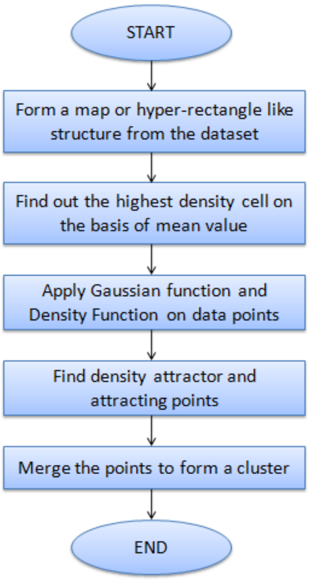
START
Form a map or hyper-rectangle like structure from the dataset
Find out the highest density cell on the basis of mean value
Apply Gaussian function and Density Function on data points
Find density attractor and attracting points
Merge the points to form a cluster
END

Fig. 16. DENCLUE algorithm.

$$g_{\text{DF}}\left(x\right) = \sum_{i=1}^{n} f_{\text{Gauss}}\left(x_1, x_i\right)$$

Then, the local maximum value of the density function for each data point is found out, which is the density attractor. This is found using a hill climbing method shown below:

$$y^2 = y^1 \text{ and } \; y^{i+1} = y^i + \delta \frac{\nabla f^D_{\text{Gauss}}\left(y^i\right)}{\left\|\nabla f^D_{\text{Gauss}}\left(y^i\right)\right\|},$$

where D represents the data point stored in the database.

After applying the above formula, the density attractor is found only if

$$f_D(y^i) < f_D(y^{i+1}),$$

where i is any cardinal value.

The points which are now selected are called attracted points and clusters are formed using these points and the density attractor.

D. Grid-Based Method

In this method, the object space is divided into a finite number of cells so as to obtain a grid-like structure [39], [40]

Fig. 17 shows the grid-based algorithm in which, firstly, a set of grid cells is defined. Then, the density of cells is calculated. The cells having density smaller than the specified threshold value are then eliminated. Finally, the contiguous cells are used to form clusters, i.e., the cells having high density are connected together so as to form a cluster [41]. It includes the following algorithms:

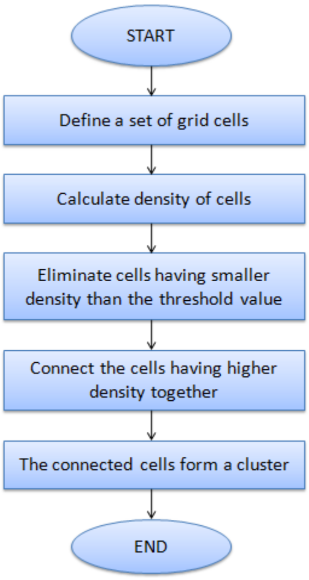
START
Define a set of grid cells
Calculate density of cells
Eliminate cells having smaller density than the threshold value
Connect the cells having higher density together
The connected cells form a cluster
END

Fig. 17. Grid-based algorithm.

1) STING (Statistical Information Grid)

In this method, the area is partitioned into rectangular shaped and forms a tree-like structure. The cells at a higher level break into cells and make a lower level. It finds out the statistical information that is stored in the cells of a tree-like structure. This statistical information helps give answers to the queries. It makes use of a top-down approach to provide an answer to queries [42]. In this algorithm, firstly, a layer from the grid is chosen. Each cell of the layer is considered and its confidence interval is calculated. It helps determine whether the cell is related to the query or not. The cells which are not related are removed and are not taken into consideration in the further process. When the current is examined completely, then the control moves to the next layer and this process keeps on repeating till the bottom layer is reached.

2) Wave Cluster

In this method, the objects are clustered using a wavelet transform method. The dense regions are identified from the data space by applying this method [43]. In this algorithm, the data are summarised in the form of multi-dimensional grid-like structure. The data objects are characterised in the n-dimensional feature space. Now, the wavelet-based transformation is used to find dense regions in feature space. This process is repeated to form clusters.

3) CLIQUE (Clustering InQUEst)

It combines density and grid-based approaches to form clusters. It first forms a grid structure of the data points and then finds out the high density cells [44]. The process of clustering using this algorithm involves the following steps:

a) The algorithm is used for partitioning the *N* dimensional data space into grids. Then the dense region is found by taking into consideration the given threshold value. A grid is considered dense if the value of data point exceeds the threshold value.

b) Apriori approach is used for generating clusters from the dense subspaces. Clique algorithm is used to generate minimum description for the clusters and this is done by finding out the maximum dense regions in subspaces and then from the maximal region, minimal cover for each cluster is found. This process is repeated till all the dimensions are covered.

E. Model-Based Method

It is based on the method probability distribution method in which clusters are formed with a high level of similarity and a low level of similarity. The similarity between the data points depends on the mean values and its key function is to reduce the squared error function [45]. Every object is associated with a density function and a probability or weight [46]. It involves the following methods:

1) Expectation-Maximization

In this method, each cluster is provided with parametric probability distribution. Different objects have mean values and weights associated, which are used to divide them into clusters [47], [48]. This model is used for latent variables whose value cannot be observed directly and is inferred from the value of the observed variables. It is also used in case if the data are incomplete or missing.

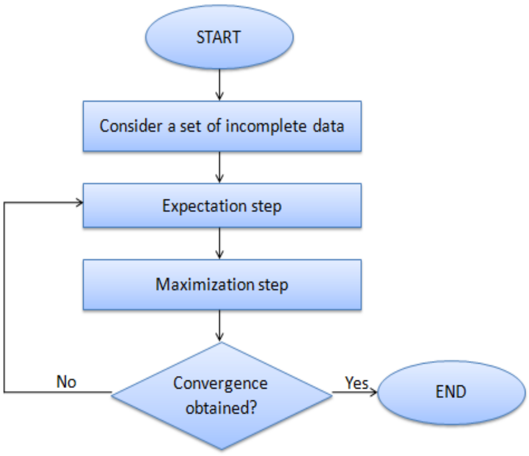

Fig. 18. Expectation-maximization algorithm.

Fig. 18 explains the EM algorithm, in which initially incomplete data are used by the system as an input. Then, E-step (Expectation step) is used, in which the observed data are used to guess the values of the incomplete data used as an input. The next step is the M-step (Maximization step), in which the value of the different parameters is updated on the basis of the complete data generated in E-step. These two steps keep on repeating until convergence is obtained.

2) Conceptual Method

It is an unsupervised method used in machine learning. It makes use of a decision tree while doing classification of objects in clusters. It includes the following algorithm:

a) COBWEB

It is a clustering algorithm that builds clusters of data points but the number of clusters formed is not pre-defined [49]. The clusters are defined with the help of conditional probability $P(B = X|C)$, where B = any attribute, X = value of attribute B and C = class to which the value of attribute belongs.

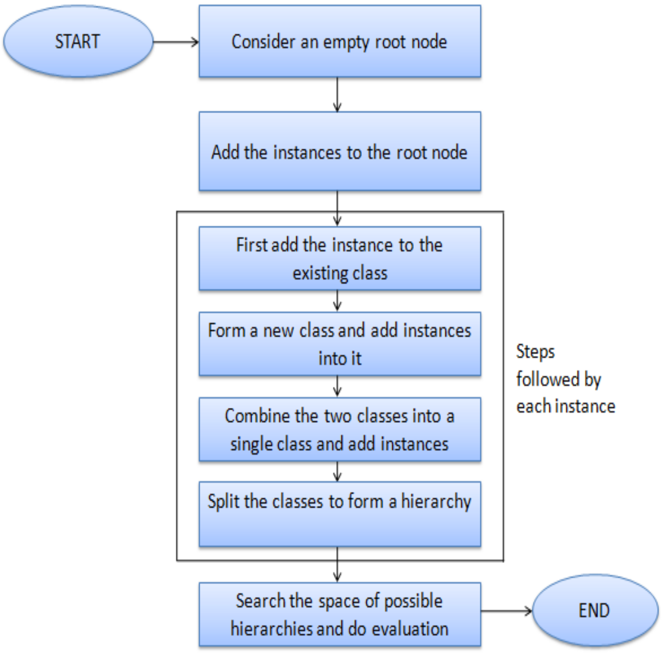
START
Consider an empty root node
Add the instances to the root node
First add the instance to the existing class
Form a new class and add instances into it
Combine the two classes into a single class and add instances
Split the classes to form a hierarchy
Steps followed by each instance
Search the space of possible hierarchies and do evaluation
END

Fig. 19. COBWEB algorithm.

Fig. 19 shows the COBWEB algorithm in which, firstly, a root node is considered that is empty. Then, the instances are added to the root node. Each instance then follows the following steps: Instance is first classified into the class that already exists. A new class is formed and instances are added to it. After this process, two classes are combined so as to form a single class and new instances are added to it. The new classes thus formed are split to form a hierarchy. Finally, the space of hierarchies possible is searched by applying the above steps and then the evaluation is done.

3) SOM (Self-Organising Map) Algorithm

SOM is an ANN that follows a “Winner-Takes-All” competitive learning method. It is an unsupervised learning method for adjusting the weight of neurons [50], [51].

SOM is a form of artificial neural network which helps in dimensionality reduction by reducing the data and this is done by representing the data in a spatially organised form and discovering the correlation between the data. SOM is used for the purpose of clustering and mapping that helps map the data in high dimensional form into data in low dimensional form so as to reduce the complex problems for easy interpretation. It comprises two different layers: the first layer is known as an input layer and the second layer is known as an output layer. Fig. 20 shows the steps of algorithm in which, firstly, data points are distributed randomly in a plane by choosing initial weights w_j. Then, a sample vector is chosen randomly from the initial training data set. Each of the data point is then examined so as to find out the data point whose weight is most similar to the weight of the chosen vector. The similarity is decided on the basis of the shortest distance, called Euclidean distance. The chosen data point is considered the winner and is referred to as the BMU (Best Match Unit). The weights of different data points are then updated and this process is repeated several times and clusters are formed at last.

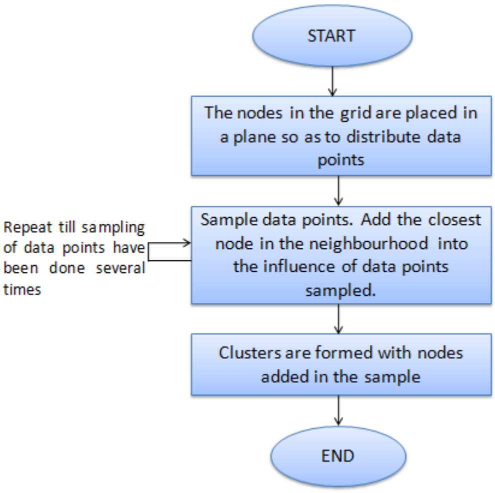
START
The nodes in the grid are placed in a plane so as to distribute data points
Sample data points. Add the closest node in the neighbourhood into the influence of data points sampled.
Repeat till sampling of data points have been done several times
Clusters are formed with nodes added in the sample
END

Fig. 20. SOM algorithm.

4) Advanced Clustering Algorithm (ACA)

This algorithm makes use of two data structures. One is used to store the cluster labels and the other is used to store the distance of data points. The algorithm can be shown as follows:

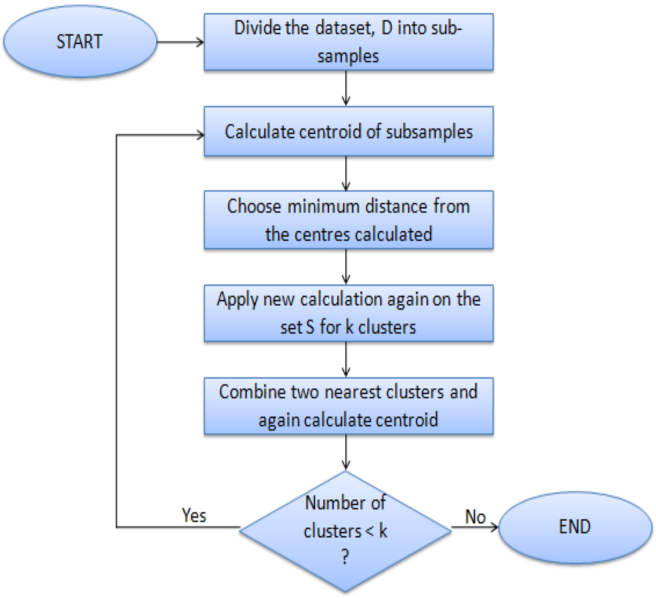
START
Divide the dataset, D into sub-samples
Calculate centroid of subsamples
Choose minimum distance from the centres calculated
Apply new calculation again on the set S for k clusters
Combine two nearest clusters and again calculate centroid
Number of clusters < k ?
Yes
No
END

Fig. 21. ACA algorithm.

Fig. 21 shows the ACA algorithm, in which input is the dataset *D* and number of clusters *k*. In this algorithm, firstly, the dataset is divided into sub-samples. Then centroid of these samples is calculated. After calculating the minimum distance clusters are amalgamated to form a single cluster. There are a number of iterations involved in forming clusters. The distance between data point and the centre of a new cluster formed is calculated and this distance is then compared to the distance of data point from the old centre. Now, if this newly measured distance is less than or equal to the previously calculated distance, then the data point will remain in the previously assigned cluster. In this way, data points are assigned to the clusters [52].

V. Comparative Analysis

In the previous section, various clustering techniques with algorithms have been discussed. The present section provides an analysis of various techniques [8], [10]–[12], [15], [16], [19]–[22], [24], [26], [27]. The different methods of clustering discussed above have their own merits and demerits which are shown in Table I. This table includes four columns, in which first column specifies the type of the clustering method. The second column tells about the algorithm and the researcher/scientist who proposed the algorithm and later on modified the algorithm, if any. The complexity of different algorithms is also mentioned in this column. In the third column, the different merits/advantages of the algorithm are mentioned and in the fourth column, the demerits/ disadvantages of the algorithms are provided. This table compares different algorithms of clustering and helps get a clear understanding of different algorithms discussed in the previous section.

TABLE I

COMPARISON OF VARIOUS CLUSTERING TECHNIQUES

No.	Clustering methods	Algorithms with complexity/Proposed by	Merits	Demerits
1.	**Partitioning Method**	*Fuzzy C-Means* **Proposed by:** Dunn (1973) [23] **Modified by:** Bezdek (1981) [8] **Complexity:** O(*n*) *n* denotes the number of data points.	➤ Better for overlapping datasets. ➤ Data points can be a part of more than one cluster. ➤ Efficient technique.	➤ Difficult to handle large dimensional datasets. ➤ The optimal solution may not always be found and there are chances of getting trapped in local optima. ➤ If the starting point is changed, then a different solution is obtained. ➤ Noisy data cannot be handled as it assigns a high membership value to the outliers.
		Gustafson Kessel FCM **Proposed by:** Gustafson and Kessel (1979) [9]	➤ Can adapt to changes in the shapes of clusters.	➤ Clusters should be of equal volume.
		Fuzzy C-Ellipto Types Clustering **Proposed by:** Bezdek (1981) [8]	➤ Overcomes limitation of G-K algorithm in which clusters had to be of equal volume.	➤ High computational cost. ➤ Parameters cannot be picked automatically.
		Fuzzy K-Means **Proposed by:** Lloyd (1957) **Published by**: Forgy (1965) [10] **Complexity:** O(kn) *n* denotes the number of data points and *k* is the number of clusters.	➤ Simple to understand and implementation is easy ➤ Efficient and fast for large datasets. ➤ Results can be easily interpreted. ➤ Produces dense clusters.	➤ Noise sensitive, i.e., it is not able to handle data containing an error. ➤ Finds locally optimal solutions. ➤ No. of clusters should be defined in the beginning. ➤ Not suited for clusters with different shapes. ➤ Works only for numerical attributes.
		Fuzzy KC-Means **Proposed by:** Atiyah *et al.* (2018) [12]	➤ Faster than *C*-Means Clustering with high accuracy as more accurate results are obtained using this algorithm. ➤ Suitable for large datasets. ➤ Processing time is less than for a CM algorithm to obtain clusters. Clusters can be obtained in less time.	➤ Noisy data cannot be handled completely but better than a *C*-means algorithm.
		K-medoid (PAM) **Proposed by:** Kaufman *et al.* (1990) [13] **Complexity:** O($k^3 \cdot n^2$) *n* denotes the number of data points and *k* is the number of clusters.	➤ Robust and fast technique as it converges in fixed number of steps. ➤ More efficient than *k*-means as medoid is less influenced by the outliers as compared to mean. ➤ Can handle noisy data. ➤ Efficient for small datasets.	➤ Not suitable for large datasets ➤ Processing is more costly or the run time cost is high. ➤ Different results can be obtained when algorithm is run multiple times on the same dataset as *k*-medoids in the beginning are chosen randomly.
		CLARA **Proposed by:** Kaufman *et al.* (1990) [15] **Complexity:** O($ks^2 + k(n-k)$) *n* denotes the number of data points, *k* is the number of clusters and *s* is the subsets	➤ Can deal with large datasets as it obtains samples from the dataset and applies PAM to different samples. ➤ Reduced computation time as compared to PAM by making improvement in the swapping method used in PAM. ➤ Storage problem is reduced.	➤ Data need to be scanned multiple times ➤ Efficiency is dependent on the sample size.
		CLARANS **Proposed by:** Ng *et al.* (2002) [16] **Complexity:** O(n^2) *n* denotes the number of data points	➤ Can detect outliers ➤ More effective than PAM and CLARA ➤ Efficient for different size of datasets.	➤ The quality of clusters is affected by the sampling done.
2.	**Hierarchical Method**	*Agglomerative Method* **Proposed by:** Sneath *et al.* (1973) [19] **Complexity:** O(n^3) but can be reduced to O(n^2) *n* denotes the number of data points	➤ Identifies small clusters ➤ Easy to implement ➤ The number of clusters needs not be specified initially.	➤ Backtracking is not possible as it is not possible to change the previous step – it means that once a value has been assigned to the cluster, it cannot be moved around. ➤ Not suitable for large datasets ➤ Less accurate as early decisions are made which cannot be changed again.

		CURE **Proposed by:** Guha *et al.* (1998) [20] **Complexity:** O(n^2log$n$) *n* denotes the number of data points		➢ More accurate technique ➢ Efficient technique and can be used for large datasets without causing any affect to the quality of clusters. ➢ The clusters of different shapes can be found.	➢ Run time complexity is high
		BIRCH **Proposed by:** Tian Zhang *et al.* (1996) [21] **Complexity:** O(n) *n* denotes the number of data points		➢ High dimensional datasets can be handled ➢ Backtracking is possible ➢ Works incrementally without any need of advanced dataset and it can be used to scan the data multiple times which improves the quality of clusters obtained. ➢ Linear scaling can be done ➢ Complete memory is utilised ➢ I/O cost is also minimised	➢ Non-numeric data cannot be handled
		ROCK **Proposed by:** Guha *et al.* (1999) [22] **Complexity:** O(n^2) *n* denotes the number of data points		➢ Huge dataset can be handled easily ➢ Effective technique for categorical datasets	➢ Static modelling technique, so it does not support incremental dataset
		CHAMELEON **Proposed by:** Karypis *et al.* (1999) [24] **Complexity:** O(n^2) *n* denotes the number of data points		➢ Uses dynamic modelling ➢ Can adapt to the properties of merged clusters	➢ Cannot handle high dimensional data
		Divisive Method **Proposed by:** Macnaughton-Smith *et al.* (1964) [26] **Complexity:** O(n) *n* denotes the number of data points		➢ Identifies large clusters easily ➢ More efficient ➢ More accurate as clusters obtained are of good quality	➢ More complex ➢ Computation is hard
		Divisive Hierarchical K-Means Algorithm **Proposed by:** Lamrous *et al.* (2007) [27] **Complexity:** O($k(nktd) + 6n$) *n* denotes the size of dataset, *k* is the cluster number, *t* denotes the count of cluster number, and *d* denotes the distance measure.		➢ Fast execution ➢ Can handle different attribute types ➢ Can work with arbitrary shapes of clusters	➢ Number of clusters must be defined in the beginning ➢ Sensitive to noisy data and outliers
3.	**Density-Based Method**	*Density-Based Connectivity*	*DBSCAN* **Proposed by:** Ester *et al.* (1996) [30] **Complexity:** O($n$log$n$) *n* denotes the number of data points	➢ Efficiently handles noisy data ➢ Different shapes clusters can be handled ➢ Faster technique ➢ The number of clusters does not need to be defined in the beginning	➢ Density of clusters may vary ➢ Cannot handle large dimensional data ➢ Faces difficulty in separating closely located clusters
			OPTICS **Proposed by:** Markus *et al.* (1999) [34] **Complexity:** O($n$log$n$) *n* denotes the number of data points	➢ Can work with data of varying density	➢ Cannot handle high dimensional data.
			DBCLASD **Proposed by:** Xu *et al.* (1998) [36] **Complexity:** O($3n^2$) n denotes the number of data points	➢ Efficient technique for large spatial databases. ➢ Input parameters are not required to be specified as they are generated automatically. ➢ Good technique for handling noisy data	➢ Slower technique than DBSCAN and DENCLUE ➢ High complexity as the computation time is high.
		Density-Based Functions	*DENCLUE* **Proposed by:** Hinnebug *et al.* (1998) [37] **Complexity:** O(log|D|) where *D* is the dataset given	➢ Makes use of mathematical function ➢ Can handle noisy data as erroneous data can be detected easily. ➢ Faster than DBSCAN in terms of processing.	➢ Requires a greater number of parameters

4.	**Grid-Based Method**	*STING* **Proposed by:** Wang *et al*. (1997) [42] **Complexity:** O(*n*) *n* denotes the number of data points	➢ Independent of query because of the presence of statistical information ➢ Provides the facility of parallel processing as multiple queries can be handled. ➢ Efficient technique ➢ Update can be done incrementally instead of computing all the information in the cell hierarchy again.	➢ Inaccurate clusters can be obtained ➢ Quality depends on the grid structure
		WAVE **Proposed by:** Sheikholesami *et al.* (2000) [43] **Complexity:** O(*n*) *n* denotes the number of data points	➢ Fast technique ➢ Clusters obtained are of high quality and the result is not affected by outliers ➢ Can work with high dimensional data ➢ Can remove outliers automatically ➢ There is no need to have knowledge regarding the number of clusters initially.	➢ Works with numerical data.
		CLIQUE **Proposed by**: Agrawal *et al*. (1998) [44] **Complexity:** O(n+k^2) *n* denotes the number of data points, and *k* is the number of clusters	➢ Scalable technique as the size of the input can be increased ➢ The order of input does not matter	➢ Costly technique as the cost of mining is high. ➢ Does not work if clusters are of varying densities as the threshold value is fixed in this algorithm. ➢ Low and high dimensionality data have the same threshold value
5.	**Model-Based Method**	*Expectation-Maximisation* **Proposed by:** Dempster *et al*. (1977) [47] **Complexity:** O(mn^3) *n* denotes the number of data points, *m* is the number of iterations	➢ Simple ➢ Implementation is easy	➢ Computation is expensive as it takes more time in forming clusters. ➢ Needs large datasets ➢ Rate of convergence is slow
		Conceptual Method	➢ Simple approach	➢ Estimating the quantity of clusters is hard when the databases are large.
		COBWEB **Proposed by:** Fisher (1987) [49] **Complexity:** O(n^2) *n* denotes the number of data points	➢ Simple method of conceptual learning	➢ Sensitive to the order of data records. ➢ Not suitable for large datasets. ➢ Assumes that the probability distribution done on the attributes is independent which is not always true as correlated attributes may be present. ➢ The storage and update of the clusters are expensive processes due to the probability distribution representation of the clusters.
		SOM **Proposed by:** Kohonen (1990) [50] **Complexity:** O(n^2m) *n* denotes the number of data points and *m* is the sub-samples	➢ Can handle noisy data and outliers ➢ Data can be easily understood and interpreted. ➢ The reduction in dimensionality makes it easy to observe similarity in the data. ➢ Fast processing speed in case there are a fewer number of neurons.	➢ Computation is expensive and complex as the complexity of this algorithm is high. ➢ Uses a heuristic approach ➢ Not a deterministic algorithm so a different result can be produced. ➢ The sufficient data are required to obtain meaningful clusters. ➢ The computing speed reduces as the number of neurons increases.
		ACA **Proposed by:** Toor (2014) [52] **Complexity:** O(*nk*) *n* denotes the number of data points and *k*, the number of clusters	➢ Efficient and simple algorithm ➢ Can work on a high dimensional dataset ➢ High accuracy	➢ More research needs to be done to know its demerits.

VI. Observation and Results

In this paper, various clustering techniques with algorithms have been discussed and it has been observed that FCM, a partition-based method of clustering, is an efficient technique and can be used for datasets that are overlapping. However, its main drawback is its possibility of getting trapped in local optima and it finds it difficult to handle large datasets. Its complexity is very low, i.e., O(n), here n is the number of data points. A variation of this algorithm is Gustafson and Kessel algorithm, in which clusters can be of different shapes but of equal volume. This problem was solved in Fuzzy *C*-Elliptotypes algorithm in which clusters could be of different volume. The FKM algorithm is an efficient algorithm for big datasets and is simple to understand and implement but is noise sensitive and works only for numerical attributes. In order to get advantages of *C*-means and *K*-means algorithms together, these two can be combined to form a *KC*-means algorithm which provides more accurate results than both the algorithms and the time of processing also gets reduced in this algorithm. One more algorithm that can be used in the partitioning method is k-medoid algorithm, which is also known as PAM. It is efficient for small datasets and can handle data, which contain error, but it is costly. Its two variations are CLARA and CLARANS. In CLARA, the computation time is reduced. However, CLARANS is much more effective than both CLARA and PAM. Agglomerative and divisive methods of clustering help divide the data points into clusters in the form of a hierarchy. These methods follow top-down and bottom-up approaches respectively in forming clusters. Different algorithms that use agglomerative methods are CURE, ROCK, BIRCH and CHAMELEON. BIRCH and CURE are agglomerative clustering algorithms that make use of the centroid of clusters for the purpose of labelling. The execution time of BIRCH is lower than that of CURE. However, these algorithms do not take into consideration the information about the inter-connectivity between the objects. For this purpose, the CHAMELEON algorithm is used. It solves the problem of inter-connectivity. It is an efficient method that makes use of dynamic modelling in order to obtain clusters. ROCK algorithm is also good for large datasets. The next method used to obtain clusters is the density-based method that includes DBSCAN, OPTICS, DBCLASD and DENCLUE algorithms. OPTICS, DBSCAN and DBCLASD algorithms are based on density-based connectivity. DBSCAN and OPTICS cannot handle high dimensional data. In case of DBSCAN, clusters of varying densities cause problems, which can be handled by the OPTICS algorithm. DBCLASD is efficient for large spatial databases. DENCLUE is based on a density-based function and has mathematical foundation. It can handle noisy data. It is faster than DBSCAN algorithm but requires large number of parameters. The next method used is grid-based method that includes STING, WAVE and CLIQUE algorithms. STING is a very efficient algorithm that provides the facility of parallel processing but sometimes produces inaccurate clusters. Wave clustering technique is very fast. It can work with high dimensional data easily. CLIQUE is a scalable but costly technique. The next method is model-based method that includes EM, COBWEB, SOM and ACA algorithms. EM algorithm is based on Expectation and Maximization parameters and is a simple algorithm to implement. COBWEB is a simple method of conceptual learning but cannot be applied to large datasets. SOM has a fast processing speed and is a complex algorithm. ACA algorithm overcomes hierarchical limitations of SOM algorithm and can work with a high dimensional dataset.

VII. Conclusion

Clustering can be considered one of the most significant techniques for knowledge discovery from the databases. Many techniques are available for this purpose. Some of the most commonly used clustering techniques have been discussed by the authors. The various algorithms are available to make clusters of the data available. Each algorithm has its own benefits and computation time. Some algorithms can handle high dimensional data very well, such as wave clustering, CLARANS etc., which can be used more in future as they are more effective techniques. Other algorithms discussed have their own merits and can be used to retrieve information from large databases in many fields.

References

[1] L. Kaufman, and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 2009.

[2] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis, "IEEE *Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, Sep. 2014. https://doi.org/10.1109/TETC.2014.2330519

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999. https://doi.org/10.1145/331499.331504

[4] D. T. T. Khaing, "Review the clustering algorithm in big data," *International Journal of Advance Research and Innovative Ideas in Education*, vol. 5, no. 4, pp. 1390–1403, 2019.

[5] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means algorithm," *Computers & Geosciences*, vol. 10, no. 2–3, pp. 191–203, Dec. 1984. https://doi.org/10.1016/0098-3004(84)90020-7

[6] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 248–255, Mar. 1986. https://doi.org/10.1109/TPAMI.1986.4767778

[7] M.-C. Hung, and D.-L. Yang, "An efficient fuzzy c-means clustering algorithm," in *2001 IEEE International Conference on Data Mining*, pp. 225–232. https://doi.org/10.1109/ICDM.2001.989523

[8] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[9] D. E. Gustafson, and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pp. 761–766. https://doi.org/10.1109/CDC.1978.268028

[10] O. J. Oyelade, O. O. Oladipupo, and I. C. Obagbuwa, "Application of k-means clustering algorithm for prediction of students' academic performance,"*International Journal of Computer Science and Information Security*, vol. 7, no. 1, pp. 292– 295, 2010.

[11] A. K.Jumaa, A. A. Abudalrahman, R. R. Aziz, and A. A.Shaltooki, "Protect sensitive knowledge in data mining clustering algorithm," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 15, pp. 3422–3431, 2017.

[12] I. A. Atiyah, A. Mohammadpour, and S. M. Taheri, "KC-Means: A fast fuzzy clustering," *Advances in Fuzzy Systems*, article number 2634861, 2018. https://doi.org/10.1155/2018/2634861

[13] L. Kaufman, and P. J.Rousseeuw, *Clustering by Means of Medoids.*Faculty of Mathematics and Informatics, 1987.

[14] H.-S. Park, and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, part 2, pp. 3336–3341, Mar. 2009. https://doi.org/10.1016/j.eswa.2008.01.039

[15] L. Kaufman, and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, 1990.

[16] R. T. Ng, and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, Sep./Oct. 2002. https://doi.org/10.1109/TKDE.2002.1033770

[17] E. Schubert, and P. Rousseeuw, "Faster *k*-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms," Lecture Notes in Computer Science, vol 11807. Springer, Cham, 2018. https://doi.org/10.1007/978-3-030-32047-8_16

[18] M. K. Rafsanjani, Z. A. Varzaneh, and N. E. Chukanlo, "A survey of hierarchical clustering algorithms, *"The Journal of Mathematics and Computer Science*, vol. 5, no. 3, pp. 229–240, 2012. https://doi.org/10.22436/jmcs.05.03.11

[19] P. H. A. Sneath, and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman and Company, 1973.

[20] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases, *"Information Systems*, vol. 26, no. 1, pp. 35–58, Mar. 2001. https://doi.org/10.1016/S0306-4379(01)00008-4

[21] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications, *"Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, Jun. 1997. https://doi.org/10.1023/A:1009783824328

[22] S. Guha, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," in *15th International Conference on Data Engineering*, IEEE, 1999, pp. 512–521. https://doi.org/10.1109/ICDE.1999.754967

[23] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters,"*Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, Jan. 1973. https://doi.org/10.1080/01969727308546046

[24] G. Karypis, and E.-H. Han, "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling," vol. 32, no. 8, pp. 68–75, Aug. 1999. https://doi.org/10.1109/2.781637

[25] X. Cao, T. Su, P. Wang, G. Wang, Z.Lv, and X. Li, "An optimized chameleon algorithm based on local features," in*10th International Conference on Machine Learning and Computing*, ACM, 2018, pp. 184–192. https://doi.org/10.1145/3195106.3195118

[26] P. Macnaughton-Smith, W. T. Williams, M. B. Dale, and L. G. Mockett, "Dissimilarity analysis: a new technique of hierarchical sub-division, *"Nature*, vol. 202, pp. 1034–1035, 1964. https://doi.org/10.1038/2021034a0

[27] S.Lamrous. and M.Taileb, "Divisive hierarchical k-means," in *International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, IEEE, 2006, p. 18. https://doi.org/10.1109/CIMCA.2006.89

[28] J. Di, and X. Gou, "Bisecting k-means algorithm based on k-valued self-determining and clustering center optimization," *Journal of Computers*, vol. 13, no. 6, pp. 588–595, Jun. 2018. https://doi.org/10.17706/jcp.13.6.588-595

[29] Y. El-Sonbaty, M. A. Ismail, and M. Farouk, "An efficient density based clustering algorithm for large databases," in *16th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2004, pp. 673–677. https://doi.org/10.1109/ICTAI.2004.27

[30] M. Ester, H.-P.Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[31] A. Merk, P. Cal, and M. Wozniak, "Distributed DBSCAN algorithm – Concept and experimental evaluation," in *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*. Advances in Intelligent Systems and Computing, vol 578, Springer, Cham. https://doi.org/10.1007/978-3-319-59162-9_49

[32] G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," in *Nirma University International Conference on Engineering*, IEEE, 2012, article number 6493211. https://doi.org/10.1109/NUICONE.2012.6493211

[33] L. Meng'Ao, M. Dongxue, G. Songyuan, and L. Shufen, "Research and improvement of DBSCAN cluster algorithm," in *7th International Conference on Information Technology in Medicine and Education*, IEEE, 2015, pp. 537–540. https://doi.org/10.1109/ITME.2015.100

[34] M.Ankerst, M. M. Breunig, H.-P.Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACMSIGMOD Record*, vol. 28, no. 2, pp. 49–60, Jun. 1999. https://doi.org/10.1145/304181.304187

[35] B. Shen, and Y.-S. Zhao, "Optimization and application of OPTICS algorithm on text clustering, *"Journal of Convergence Information Technology*, vol. 8, no. 11, pp. 375–383, Jun. 2013. https://doi.org/10.4156/JCIT.VOL8.ISSUE11.43

[36] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander, "A distribution-based clustering algorithm for mining in large spatial databases," in *14th International Conference on Data Engineering*, IEEE, 1998, pp. 324–331. https://doi.org/10.1109/ICDE.1998.655795

[37] A. Hinneburg, and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *4th International Conference on Knowledge Discovery and Data Mining*, 1998, pp. 58–65.

[38] H. Rehioui, A. Idrissi, M. Abourezq, and F. Zegrari, "DENCLUE-IM: A new approach for big data clustering," *Procedia Computer Science*, vol. 83, pp. 560–567, 2016. https://doi.org/10.1016/j.procs.2016.04.265

[39] D. Xu, and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol.2, pp. 165–193, 2015. https://doi.org/10.1007/s40745-015-0040-1

[40] M. R. Ilango, and V. Mohan, "A survey of grid based clustering algorithms," *International Journal of Engineering Science and Technology*, vol. 2, no. 8, pp. 3441–3446, 2010.

[41] Y. Lu, Y. Sun, G. Xu, and G. Liu, "A grid-based clustering algorithm for high-dimensional data streams," in Li X., Wang S., Dong Z.Y. (eds) Advanced Data Mining and Applications. ADMA 2005. Lecture Notes in Computer Science, vol 3584. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11527503_97

[42] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *23rd International Conference on Very Large Data Bases*, 1997, pp. 186–195.

[43] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A wavelet-based clustering approach for spatial data in very large databases," *The VLDB Journal*, vol. 8, pp. 289–304, Feb. 2000. https://doi.org/10.1007/s007780050009

[44] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *ACMSIGMOD Record*, vol. 27, no. 2, pp. 94–105, Jun. 1998. https://doi.org/10.1145/276305.276314

[45] G. Schoier, and G. Borruso, "On model based clustering in a spatial data mining context," in Murgante B. *et al.* (eds) Computational Science and Its Applications – ICCSA 2013. Lecture Notes in Computer Science, vol 7974. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39649-6_27

[46] M. Meila, and D. Heckerman, "An experimental comparison of model-based clustering methods," *Machine Learning*, vol. 42, pp. 9–29, 2001. https://doi.org/10.1023/A:1007648401407

[47] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm, *"Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[48] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, Nov 1996. https://doi.org/10.1109/79.543975

[49] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, pp. 139–172, 1987. https://doi.org/10.1023/A:1022852608280

[50] T. Kohonen, "The self-organizing map, *"Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990. https://doi.org/10.1109/5.58325

[51] T. Tateyama, S. Kawata, and H. Ohta, "A conditional clustering algorithm using self-organising map, "in *SICE 2003 Annual Conference*, IEEE, 2003, vol. 3, pp. 3259–3264.

[52] A. Toor, "An advanced clustering algorithm (ACA) for clustering large dataset to achieve high dimensionality, *"Global Journal of Computer Science and Technology: C Software and Data Engineering*, vol. 14, no. 2, pp. 71–74, 2014.

Satinder Bal Gupta obtained Doctoral degree in computer science from Kurukshetra University, Kurukshetra, Haryana, India in 2011.
He is currently an Associate Professor at the Department of CSE of Indira Gandhi University, Meerpur, Rewari, Haryana, India. He has published more than 30 papers in various international/national Journals. He has written more than 15 books. His research interest includes search engines, data mining, ad-hoc networks etc. Dr Satinder is a life member of ISTE and a member of the Indian Science Congress in 2019.
E-mail: satinderbal@igu.ac.in
ORCID iD: https://orcid.org/0000-0002-6056-1489

Rajkumar Yadav obtained his Doctoral degree in computer science & engineering from Maharshi Dayanand University, Rohtak, Haryana, India in 2011. He is currently an Associate Professor at the Department of CSE of Indira Gandhi University, Meerpur, Rewari, Haryana, India. He has published more than 50 papers in various international/national journals. He has completed the major research project granted by UGC, MHRD, Govt of India. He has research interest in information hiding techniques, water marking, finger printing, data mining etc. Dr Rajkumar is a life member of ISTE and Indian Science Congress.
E-mail: rajyadav76@rediffmail.com
ORCID iD: https://orcid.org/0000-0003-0605-8759

Shivani Gupta obtained Masters degree in computer science from Vaish College of Engineering, MDU Rohtak, Haryana, India in 2015. She received her Bachelor’s degree from R. N. College of Engineering, MDU Rohtak, Haryana, India in 2013. She is currently working as a faculty member at the Department of Computer Science and Engineering, Indira Gandhi University, Meerpur, Rewari, Haryana, India. She has published two research papers in international journals.
E-mail: shivanigupta646@gmail.com