Journal on Artificial Intelligence
DOI:10.32604/jai.2020.014944
Article

Tech Science Press

Clustering Algorithms: Taxonomy, Comparison, and Empirical Analysis in 2D Datasets

Samih M. Mostafa[1,2,*]

[1]Computer Science-Mathematics Department, Faculty of Science, South Valley University, Qena, 83523, Egypt
[2]Fellow of the Academy of Scientific Research and Technology (ASRT), Egypt
[*]Corresponding Author: Samih M. Mostafa. Email: samih_montser@sci.svu.edu.eg

Received: 29 October 2020; Accepted: 03 December 2020

Abstract: Because of the abundance of clustering methods, comparing between methods and determining which method is proper for a given dataset is crucial. Especially, the availability of huge experimental datasets and transactional and the emerging requirements for data mining and the like needs badly for clustering algorithms that can be applied in various domains. This paper presents essential notions of clustering and offers an overview of the significant features of the most common representative clustering algorithms of clustering categories presented in a comparative way. More specifically the study is based on the numerical type of the data that the algorithm supports, the shape of the clusters, and complexity. The experiments were done using nine clustering algorithms representing the common clustering categories on eight 2D clustered datasets differ in the clusters' shapes and density of the data points. Furthermore, the comparison was done from the point of view seven performance measures.

Keywords: Clustering algorithms; clustering taxonomy; clustering validation indices; partition-based clustering; hierarchical-based clustering; density-based clustering.

1 Introduction

Clustering means dividing the data into groups that are useful, meaningful, or both; the greater the difference between clusters and the greater the homogeneity (or similarity) within a cluster, the better the clustering result [1–3]. Tab. 1 describes the clustering steps and their descriptions. Clustering has long played a substantial role in a wide variety of areas: Biology, pattern recognition, machine learning, information retrieval, data mining, psychology, and other social sciences. The best definition of clustering depends on the required results and the data nature. Clustering is considered as a type of classification in that it generates cluster labels of the objects. Thus, clustering is known as unsupervised classification. Machine learning includes variant topics (e.g., feature selection [4], regression [5], and classification [6]). In classification, the objects in the dataset are assigned with specific classes [7].

The lack of precise definition of the notion "cluster" led to developing many clustering methods each of which based on different induction principle [8,9]. Clustering algorithms can be classified from various points of view. Fahad et al. [10] categorized clustering algorithms to partitioning-based, hierarchical-based, density-based, grid-based, and model-based. Grid-based algorithms are included in density-based [11]. Dervis et al. [12] and Fraley et al. [13] classified the clustering algorithms as hierarchical clustering and partitional clustering. Han et al. [14] categorized the methods into three categories: Model-based clustering, density-based methods, and grid-based methods. Clustering methods can be classified according to: Type of data (e.g., numerical or categorical), similarity between two objects, optimization of the clustering criterion, and fundamental concepts and theory on which the clustering analysis is based (e.g., statistics, fuzzy theory). From the point of view of partitioning technique, clustering methods can be classified into three essential types (Tab. 2): i) Partition-based clustering; the algorithm partitions the dataset into a set of (typically) separated clusters called partitions, each partition states a cluster which contains at least one object and each object should be categorized to one cluster. Partition-based clustering algorithms require the number of clusters to be set by the user. There are many methods using partition-based clustering such as K-means [14,15], K-medoids [16], K-modes [17], PAM [18], CLARA [19], CLARANS [20], FCM [21], and CluStream [22,23], ii) Hierarchical-based clustering; the algorithm proceeds sequentially by either splitting larger clusters, or by combining smaller groups into larger ones, a tree of clusters resulted from the algorithm is called dendrogram, agglomerative and divisive are two types of hierarchical-based clustering. Unlike agglomerative approach which uses bottom-up strategy (i.e., initially, each object represents a cluster/partition of its own, and then these clusters are successively combined until the wished structure of the cluster is obtained), divisive approach uses top-down strategy (i.e., initially, all objects belong to one cluster, this cluster is divided into sub clusters, and successively this procedure continues until the desired structure of the cluster is obtained). Division or migration of the clusters is performed according to similarity measure to optimize some criterion (e.g., a sum of squares). Agglomerative [24], ROCK [25], BIRCH [26], and Chameleon [27] are examples of algorithms that use hierarchical-based clustering, and iii) Density-based clustering; the algorithm groups the neighbors objects of the dataset into groups based on density conditions. The density-based clusters are separated from one another by low density regions and are often considered to be outliers [14,15]. The points that belong to a cluster are drawn from an identified probability distribution [28]. DBSCAN [29], and OPTICS [30] use density-based clustering technique.

Table 1: Clustering steps and their descriptions

Step	Description
1- Feature selection	▪ Clustering instances depends on set of features. The main goal is selecting the convergent features of the instances to be clustered.
2- Choice of clustering algorithm	▪ Choosing the algorithm that is more convenient for the data on hand. Clustering criterion and similarity measure are selected in tandem: - Optimizing the cost function (clustering criterion) and expecting the type of generated clusters. - Ensuring that the selected features share evenly to the computation of the closeness measure and no features dominate others.
3- Validation the results	▪ Verifying the correctness of the clustering algorithm results using appropriate techniques and criteria.
4- User decision	▪ Choosing the results obtained or starting from the beginning using different parameters or perhaps changing different algorithm.

The type of features allowed in the dataset determines the algorithm used in the clustering, for example, statistical algorithms are used for clustering numeric data, conceptual algorithms are used for clustering categorical data, fuzzy clustering algorithms allow object to be classified into all clusters with a 1 ranging from 0 to 1, this membership degree indicates the similarity of the object to the mean of the cluster. For an object, the sum of the memberships is equal to 1. Unlike fuzzy clustering algorithms (a.k.a. soft clustering), crisp clustering (a.k.a. hard clustering) allows object to be classified to one and only one cluster. From another point of view of determining the number of clusters, clustering algorithms can be classified into user-dependent; the algorithm requires the user to determine the number of clusters, and self-dependent; the algorithm itself determines the number of clusters, others are mixed; determining the number of clusters is optional. Fig. 1 provides the essential categorization of clustering techniques from the point of view of portioning, computing, and determining number of clusters.

Table 2: Three essential types of clustering categories

Category	Technique	General characteristics
Partition-based clustering	Distance based	– Find clusters of spherical shape – May use medoid or mean (etc.) to represent the center of the cluster – Effective for small/medium datasets
Hierarchical-based clustering	Top-down or down-top decomposition	– Clustering is a multiple levels decomposition (i.e., hierarchical) – Cannot correct erroneous splits or merges. – May consider object "linkages" or combine other techniques (e.g., microclustering)
Density-based clustering	low-density regions separating high-density regions of objects	– Can find arbitrarily shaped clusters – Minimum number of neighbors of points is required for each point. – May filter out outliers

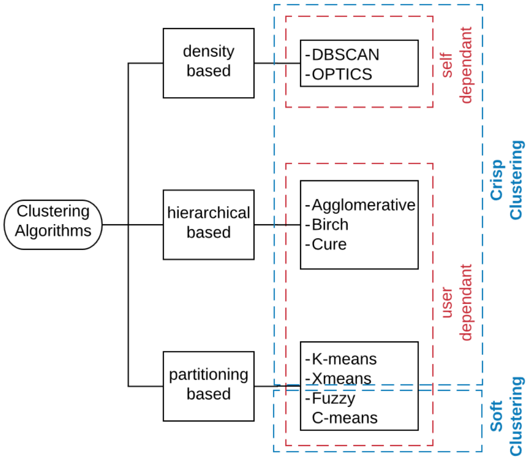

Figure 1: Clustering algorithms categorization and some examples of clustering algorithms

Motivation: In real-life applications, the main issue is to perform cluster analysis of the dataset. Some of real-life examples, where clustering data is very important, are: Gene expression analysis (clustering data that contain thousands of genes under hundreds of conditions), text documents (clustering documents are based on similarities of words in a subset of terms and the frequency of words), customer recommendation systems (clustering customers with some overlapping preferences depend on the type of product), etc.

Objective: This paper revises some of main methods to clustering algorithms comparison, presents the R and Python packages in the comparisons, and evaluates the clustering algorithms from the point of view of time, Adjusted Rand Index (ARI), Fowlkes-Mallows Index (FMI), Normalized Mutual Information (NMI), Homogeneity, Completeness, and V-Measure (Fig. 2). In addition, the author highlighted the set of best clustering algorithms for dataset.

Organization: The rest of this paper is divided as follows: Section 2 discusses the related work. Section 3 presents the benchmark datasets clustering algorithms. The chosen clustering algorithms are discussed in Section 4. The experimental implementation is discussed in Section 5. Fig. 3 shows the structure of this survey.

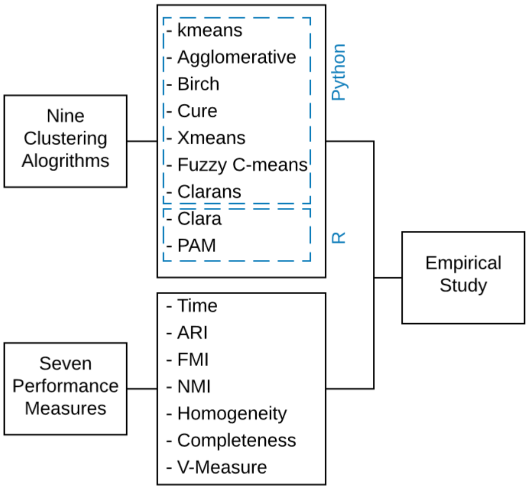

Figure 2: Clustering algorithms and validity measures used in this study

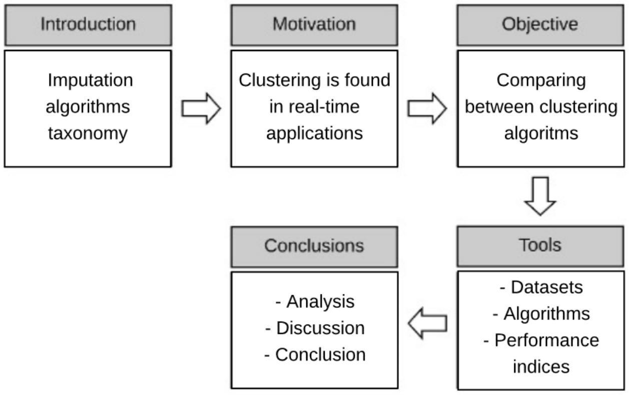

Figure 3: Organization of survey

2 Related Works

Gang et al. [31] used six clustering algorithms, eleven cluster validity indices, and three MCDM methods over bankruptcy risk and three real-life credit risk datasets to present a multiple criteria decision making (MCDM) based approach to rank the selection of widespread clustering algorithms. Their results showed that there is no algorithm achieves the best performance on all. Their results showed that no algorithm can achieve the best efficiency and performance on all indices for any dataset and, therefore, using more than one performance measure is mandatory.

Tomi et al. [32] performed a comparative analysis of clustering algorithms for text independent speaker verification. The comparison was done between K-means, expectation–maximization, random swap, pairwise nearest neighbor, split, split-and-merge, genetic algorithm, self-organizing map, and fuzzy C-means. The main conclusion they obtained is: The order of the model is the most important parameter, whereas the choice of the clustering method is critical only if small model size is used.

Marcilio et al. [33] provided analysis of 35 microarray cancer gene expression datasets (14 double-channel cDNA datasets and 21 single-channel Affymetrix chips datasets) using 7 different clustering approaches (complete linkage (CL), single linkage (SL), shared nearest neighbor-based clustering (SNN), spectral clustering (SPC), mixture of multivariate Gaussians (FMG), k-means (KM), and average linkage (AL)) and four proximity measures (Cosine (C), Pearson's Correlation coefficient (P), Euclidean Distance

(E), and Spearman's correlation coefficient (SP). From their experimental results, they inferred that the FMG showed the best performance followed by KM from the point of view of recovery of actual structure of the datasets, regardless of the measure used.

Ivan et al. [34] compared five clustering algorithms (CLICK, hierarchical clustering, k-means, dynamical clustering, and self-organizing maps (SOM)) for datasets of gene expression. The authors applied k-fold cross-validation process adapted to unsupervised approaches to evaluate the clustering algorithms. Some of the conclusions the authors got are: 1) Hierarchical clustering methods are affected to noisy and outliers, therefore, it has low accuracies for the Functional Classification (FC) datasets, 2) Dynamical clustering, k-means, and SOM give better accuracies in all experiments, 3) Selecting the parameters in SOM requires more complex experiments, this disadvantage does not exist in dynamical clustering and k-means, and 4) The clustering structure in SOM is more informative than dynamical clustering and k-means, because clustering structure in SOM depends on neighborhood relations returning topological map.

Lance et al. [35] presented a survey of various subspace clustering methods, and compared the two main methods to subspace clustering using accuracy tests and empirical scalability. The authors compared between representative top-down algorithm (FINDIT) and bottom-up algorithm (FINDIT). They concluded from their experiments that bottom-up approaches perform well in very high dimensionality datasets. They measured the running time of the two algorithms to measuring the scalability of the compared algorithms.

Ujjwal et al. [36] evaluated the performance of single linkage, hard K-Means, and a simulated annealing from the point of view of four clustering validity indices, namely Dunn’s index, a recently developed index I, Calinski-Harabasz index, and Davies-Bouldin index for both real-life and artificial datasets with clusters number varying from two to ten. When the appropriate clusters number is achieved, I attained its maximum value. I is considered to be more reliable and consistent in indicating the correct clusters number.

Sylvain et al. [37] compared four algorithms: Restricted Neighborhood Search Clustering (RNSC), Markov Clustering (MCL), Molecular Complex Detection (MCODE), and Super Paramagnetic Clustering (SPC). The authors sampled the parameter space, analyzed the sensitivity of the methods to the parameters, and selected optimal parameters. In various proportions, the authors randomly added/removed edges to/from the test graph to evaluating the robustness from the point of view of false negatives and false positives. The authors applied the algorithms to six datasets from the General Repository for Interaction Datasets (GRID).

Abla et al. [38] implemented and compared between four algorithms namely, agglomerative hierarchical, K-means, SOM, and DBSCAN on four sparse industrial datasets (Logistics, Automotive quality systems, Aircraft, and Customer’s requirements). The authors used Nbclust package to find the appropriate number of clusters. Eight validity indices (C-index, CH, Dunn, Gamma, BH, DB, tau, and Connectivity) were used for comparing between the compared algorithms. The authors highlighted efficient clustering algorithm. As a general conclusion, DBSCAN is recommended for noisy dataset, K-means for the big dataset, and SOM for small dataset.

C´assio et al. [39] presented a comparative study of three clustering algorithms: MR-Stream, CluStream, and STREAM on 90 synthetic datasets. The datasets were created from spatial point processes following Mixtures of Gaussians or Gaussian distributions. The experiments were executed in three scenarios: 1) High dimensional with concept drift, 2) Low dimensional, and 3) Low dimensional with concept drift. The authors concluded that CluStream performs better than the other algorithms from the point of view of the quality of clustering at a higher execution time cost.

Marina et al. [40] compared three model-based clustering algorithms (Expectation–Maximization (EM), agglomerative clustering, and Classification EM (CEM)) on high dimensional discrete features datasets. Each algorithm uses the same underlying model (e.g., naive-Bayes with a hidden root node). The authors found that EM algorithm significantly performs better than the other methods.

Tagaram [41] examined two clustering algorithms K-Medoids and K-Means. The author concluded that the execution time of K-Means is convenient efficient for small data sets; however, the number of clusters must be known in advance. On the other hand, K-Medoids performs better for big datasets. Dongkuan et al. [1] introduced the core and basic of commonly used clustering algorithm from 19 categories. The authors specified the source and analyzed the pros and cons of each algorithm. The authors took into account the essential elements required in the clustering process such as the similarity measurement or distance and evaluation indicators.

Pranav et al. [42] compared between eight clustering algorithms from different categories. These algorithms are: K-means, K-means++, Kernel K-means, Hierarchical clustering, Fuzzy CMeans, Model based, DBSCAN, and OPTICS on CURE-T2-4K and CLUTO-T8-8K datasets. The authors concluded that the compared algorithms could not precisely identify the clusters, the algorithms are sensitive to noise and outliers, computational complexity and high time.

3 Benchmark Datasets

The main problem one may face with clustering is deciding the number of clusters that fits the dataset. 2D-datasets are used in most algorithms' experimental evaluations to enable the reader to visually verify the validity of the obtained results (i.e., how well the clustering algorithm generated the clusters of the data set). It is clear that dataset visualization is a significant verification of the clustering results. In the case of large multidimensional datasets, effective dataset visualization would be difficult.

To compare the chosen clustering algorithms, eight clustered 2D-datasets (Tab. 3 summarizes the number of instances, number of features, and number of clusters) are used in the experiments, Fig. 4 presents the panels of the benchmark dataset used. In the first panel, the dataset namely Aggregation dataset consists of seven perceptually distinct collections of points. The features of the dataset create difficulties (e.g., uneven-sized clusters, narrow bridges between clusters, etc.). In the second panel, the dataset namely compounds consists of six groups varying in the shape, and densities.

In the left-upper part of the panel, the clusters are not really well separated because they touch. The separation between the clusters can be perceived by the low density where the clusters touch. In the left-bottom part of the panel, the points fall into two well separated nested distinct clusters, the right part presents two nested, not separated, areas of different point densities. The third panel presents 31 randomly placed 2-D Gaussian groups of 100 points each. The dataset in the fourth panel consists of two clusters not really well separated because they touch.

In the fifth panel, the dataset is clustered into two separated groups where the density varies within each cluster but the separation between the two clusters is still substantial. The dataset in the sixth panel consists of a circular group with an opening part near the bottom; the circular cluster includes two Gaussian distributed groups inside. The Spiral dataset in the seventh panel has two well separated nested distinct clusters in 2-circle dataset. The last panel contains 15 similar 2-D Gaussian distributions that are positioned in rings.

Table 3: Datasets used in the experiments

Dataset	#instances	#features	#clusters	References
Aggregation	788	2	7	[43,44]
Compound	399	2	6	[45,46]
D31	3100	2	31	[47,48]
Flame	240	2	2	[49,50]
Jain	373	2	2	[51,52]
Pathbased	300	2	3	[53,54]
Spiral	312	2	3	[54,55]
R15	600	2	15	[48,56]

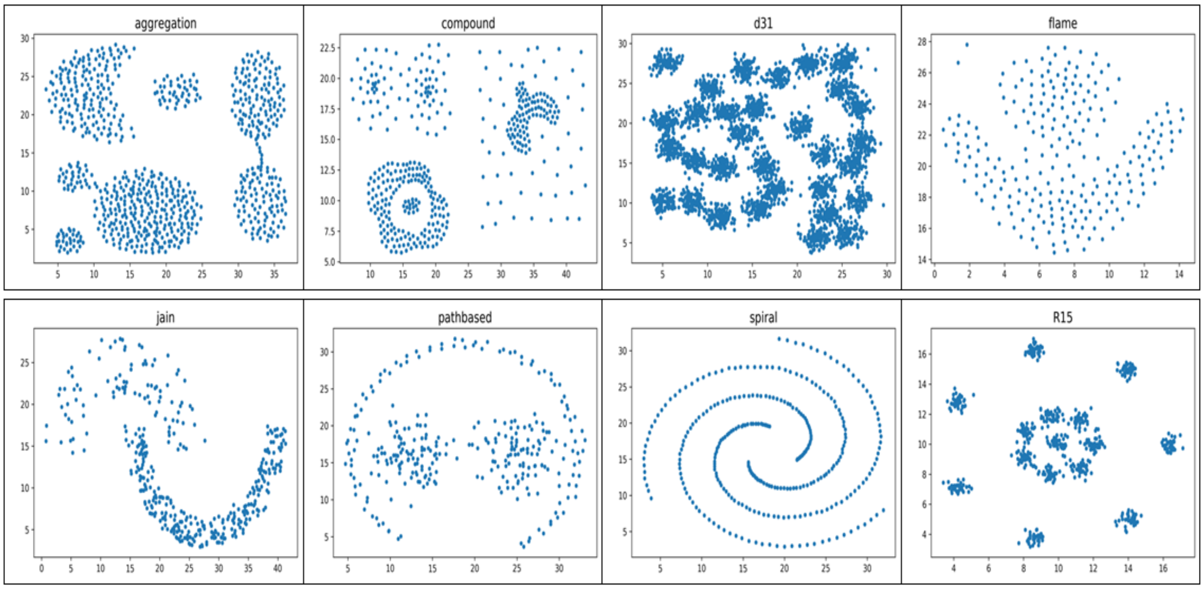

Figure 4: Benchmark datasets

4 Chosen Clustering Algorithms

Presenting a whole list of all clustering methods is hard; this is because of the intersection of research fields, the diversity of information, and the expansion of modern computer technology. So, from the three essential commonly used categorizations (density-based clustering, hierarchical-based clustering, and partition-based clustering), one or several algorithm(s) from each category will be discussed in detail. The chosen algorithms in this study are partitioning and hierarchical user-dependent algorithms, where, the datasets on hand are clustered data and the number of clusters is known. Here, the categories and their algorithms examples are indicated in Tab. 4.

Table 4: Clustering algorithms and their categories considered in the analysis. The first column presents the algorithm name. The second column presents the categories of the algorithms. The third column gives short descriptions of the algorithm. The fourth and fifth columns present the programming languages and function/library, respectively. The sixth column presents the time complexities. The last column is for the references. (n: The number of objects, k: The number of clusters, t: The number of iterations, and s: The size of the sample)

Algorithm name	Category	Short descriptions	language	Function	Complexity	References
K-means	Partitioning	– It is a centroid-based clustering algorithm. – It clusters data by separating samples in groups. – The centroid of the cluster is defined as the mean of all pints inside this cluster.	Python	KMeans/ Scikit learn	*O(nkt)*	[57–60]
X-means	Partitioning	– It is a K-means based algorithm. – It makes a local decision to determine which subset of the current centroids to be split to better fit the data. – The decision is based on Bayesian Information Criterion (BIC).	Python	xmeans/ pyclustering	*O(nkt)*	[61][62]

PAM (***P***_artitioning_ ***A***_round_ ***M***_edoids_)	Partitioning	– It is a medoids-based clustering algorithm. – It selects representative points *K* to form the initial clusters, and then repeatedly goes to better representatives of the cluster.	R	pam/ cluster	$O(k(n-k)^2t)$	[63][64]
Fuzzy C-means	Partitioning	– It is a membership-based clustering algorithm. – It assigns membership to each point of the data. – The membership degree is based on distance between the data point and each cluster center, and ranging between 0 and 1.	Python	FCM/ pyclustering	$O(n)$	[65–67]
CLARA (***C***_lustering_ ***LAR***_ge_ _Applications_)	Partitioning	– It is an implementation of PAM method in a subset of the dataset. – It selects a random sample of the dataset. – It builds clusters from multiple samples. – The best clustering is returned as the output.	R	clara/ cluster	$O(ks^2+k(n-k))$	[63][64]
CLARANS (***C***_lustering_ ***L***_arge_ ***A***_pplications based upon_ ***RAN***_domized_ ***S***_earch_)	Partitioning	– It is a medoids-based clustering algorithm. – It presents a trade-off between the effectiveness and the cost to obtain clustering. – It combines the sampling techniques with PAM.	Python	clarans/ pyclustering	$O(n^2)$	[21][63][68]
Agglomerative	hierarchical	– It is a "bottom-up" method. – It starts with each object as a separate cluster itself, and according to a distance measure, it successively merges groups	Python	agglomerativeClustering/ Scikit learn	$O(n^2* \log n)$	[9][69][70]
BIRCH (***B***_alanced_ ***I***_terative_ ***R***_educing and_ ***C***_lustering using_ ***H***_ierarchies_)	hierarchical	Partitions incoming data points in a dynamic and incremental way using a hierarchical data structure (i.e., CF-tree)	Python	birch/ Scikit learn	$O(n)$	[27][71][72]
Cure (***C***_lustering_ ***U***_sing_ ***RE***_presentatives_)	hierarchical	– It adopts a middle ground between all point extremes and the centroid based. – Well scattered points from each cluster are selected to represent the cluster, and then shrunk toward the cluster center.	Python	cure/ pyclustering	$O(n^2 \log n)$	[61][73][74]

4.1 K-means

K-means is the most commonly and simplest used clustering algorithm. It partitions the dataset into K clusters ($C_1, C_2, . . ., C_K$), represented by their means or centers to minimize some objective function (e.g., squared error function). The parameter k must be supplied by the user in advance. K-means algorithm may be considered as a gradient-decent procedure, which begins with an initial group of K cluster-centers and successively updates it to decreasing the error function. Fig. 5 shows how K-means clusters the datasets. The pseudo-code of K-means algorithm is as follows:

```
Algorithm 1          K-Means
Input                - Dataset
                     - number of clusters
Output               - K clusters

Step-1:      − Initialize K centers of the cluster
Step-2:      − while no change in centroids of the cluster do
                 − Calculate the mean of all  the objects belonging to that cluster
                         $\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q$
                     where $\mu_k$is the mean of cluster $k$ and
                     $N_k$ is the number of points belonging to that cluster
                 − Assign objects to the closest cluster centroid
                 − Update cluster centroids based on the assignment
             − end while
```

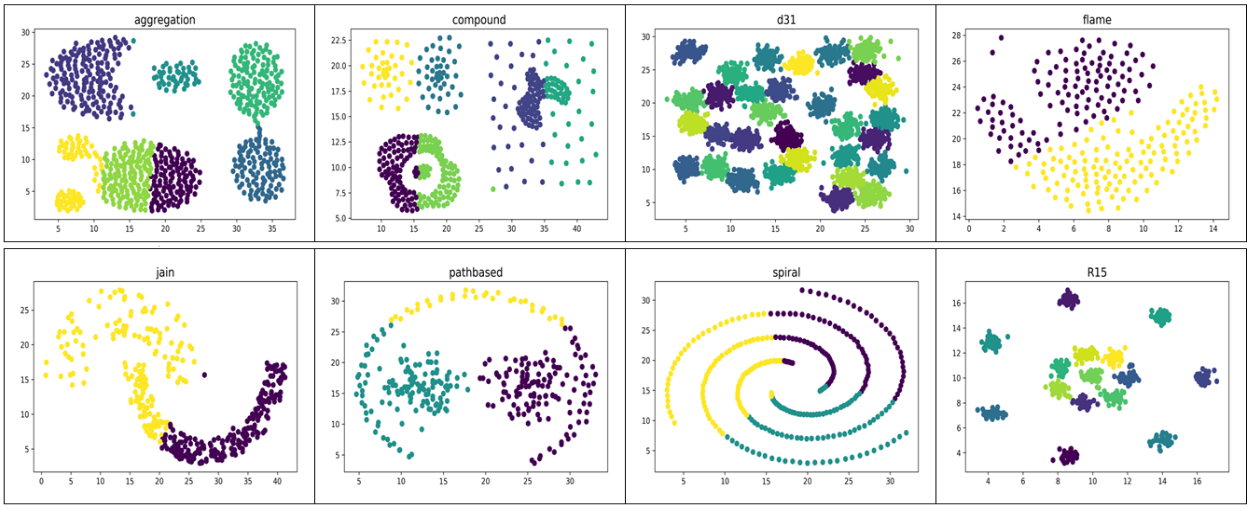

Figure 5: Clustering datasets using K-means

4.2 X-means

It is a powerful modulation of the popular k-means algorithm [75]. X-means takes an initial number and maximum number of clusters as parameters, X-means starts analysis from the initial number to select the optimal number of clusters based on Bayesian Information Criterion (BIC) computation [76,77]. Fig. 6 shows how X-means clusters the datasets. The pseudo-code of X-means algorithm is as follows:

```
Algorithm 2          X-means
Input                - Dataset
                     - initial number
                     - maximum number of clusters
Output               - X clusters

Step-1:      − For each centroid
                 − Consider their current locations
                 − Consider the bounding box geometry
                 − Calculates the membership of each centroid for a subset of points
                 − Record the centroid set and its score
                 − If score of new centroid better than previous one then
```

```
                    − Accept new record
                − **Else**
                    − Keep previous centroid
                − **End if**
        − **End for**

Step-2:     − **If** new number of clusters > maximum number of clusters **then**
                − Stop
                − Record the best score
            − **Else**
                − Go to Step-1
            − **End if**
```

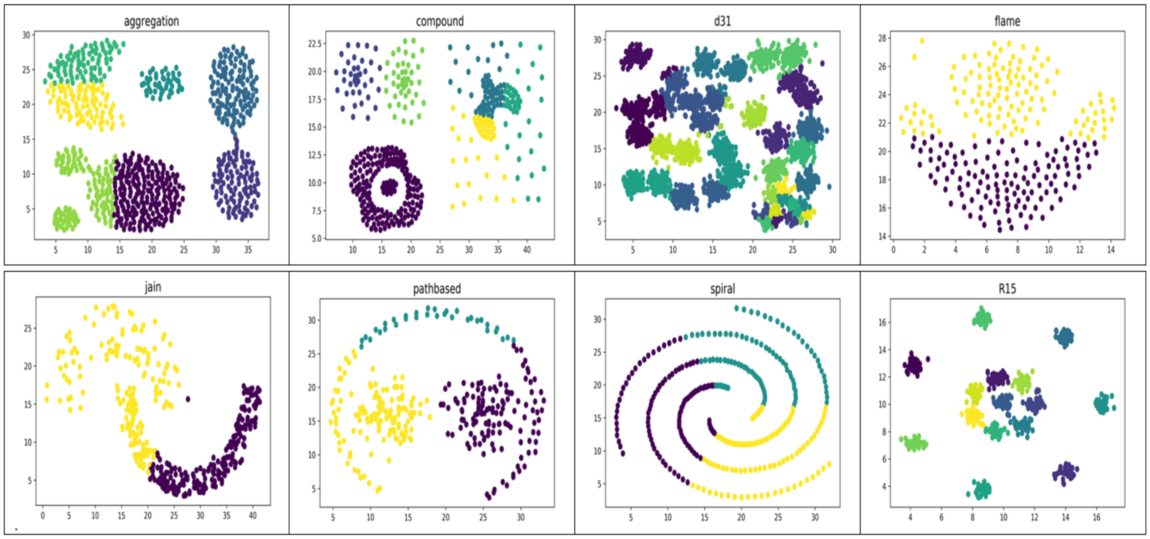

Figure 6: Clustering datasets using X-means

4.3 CLARANS

CLARANS does not use auxiliary structures and uses randomized search, increasing dimensionality has no significant effect CLARANS performance [21, 78, 79]. It does not inspect every neighbor of a node; however, CLARANS inspects a sample of the node's neighbors. Fig. 7 shows how CLARANS clusters the datasets. The pseudo-code of CLARANS algorithm is as follows:

```
Algorithm 3         CLARANS
Input               - X – Dataset
                    - K – number of clusters
                    - Numlocal – number of iterations
                    - Maxneighbor – number of adjacent nodes
Output              - K clusters

Step-1:     − Initialize i=1, mincost is a relatively large number.
Step-2:     − Set current as an arbitrary node
Step-3:     − Set j to 1
Step-4:     − Use random search to generate neighbor S of the current
            − Randomly check maxneighbor neighbors
            Based on a cost function, calculate the cost differential of these two nodes.
Step-5:     − If (cost of S < cost of current) then
                − Set current to S
```

```
                    – Go to *Step-3*
          – **Otherwise**
          – Increment *j* by 1
Step-6:   – **If** (*j* <= *maxneighborset*) **then**
                    – Go to *Step-4*
          – **Otherwise**
                    – Increment *j* by 1
Step-7:   – **If** (*j* > *maxneighbor*) **then**
                    – **If** (cost of *current* < cost of *mincost*) **then**
                         – Set *mincost* to cost of *current*
                    – Set *bestnode* to *current*
Step-8:   – Increment *I* by 1
          – **If** (*I* > *numlocal*)
                    – Output *bestnode* and stop.
          – **Otherwise**
                         – Go to *step-2*
```

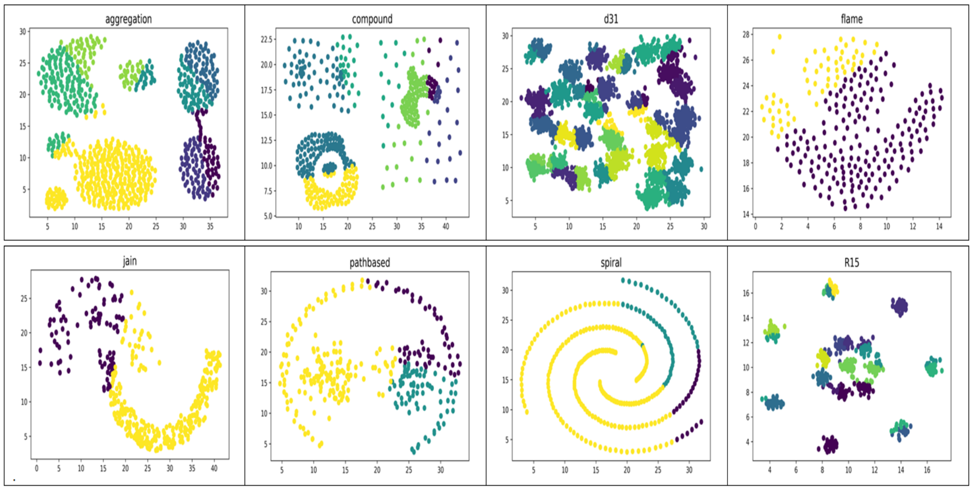

Figure 7: Clustering datasets using CLARANS

4.4 Fuzzy C-Means

FCM is a representative method of fuzzy clustering technique which is based on K-means to partition the dataset into clusters [10,80]. FCM algorithm is considered as a "soft" clustering method (i.e., the object is assigned to a cluster with a degree of belief) [81]. In each cluster, FCM finds the most characteristic point, named as the *centre*; the membership degree is computed for each object in the clusters, an object may belong 80% to one cluster and 20% to another [82]. Fig. 8 shows how FCM clusters the datasets. The pseudo-code of FCM algorithm is as follows:

```
Algorithm 4   FCM Algorithm
Input         – X - Dataset
              – C - number of clusters
              – t - convergence threshold (termination criterion)
Output        – U - membership matrix
Step-1:       – Randomly initialize matrix U

Step-2:       – Repeat
                    – Calculate the cluster centroids.
```

```
            − Compute dissimilarity between data points and
              centroids.
            − Compute a new $U$.
    − Until
            − The improvement between consecutive iterations is
              below $t$.
```

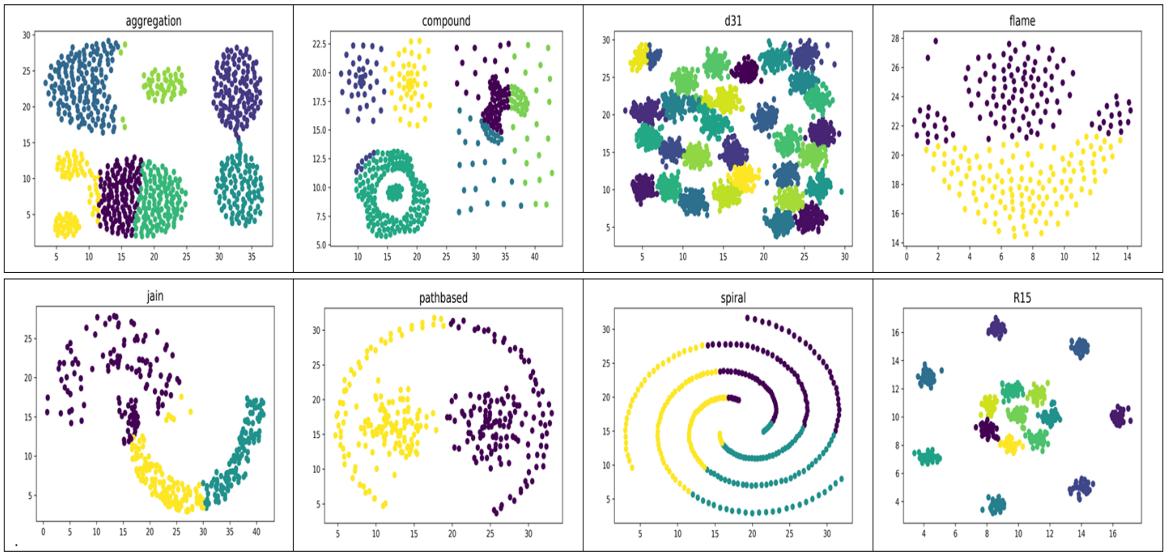

Figure 8: Clustering datasets using FCM

4.5 CLARA

CLARA is an extension to PAM to deal with large datasets. Instead of finding medoids for all the dataset, a small sample of the data is considered, which in turn reduces computing time. Then, CLARA applies PAM algorithm to generate the optimal set of medoids for that sample. In order to minimizing the sampling bias, the clustering and sampling processes are repeated for a pre-specified number of times. Fig. 9 shows how CLARA clusters the datasets.

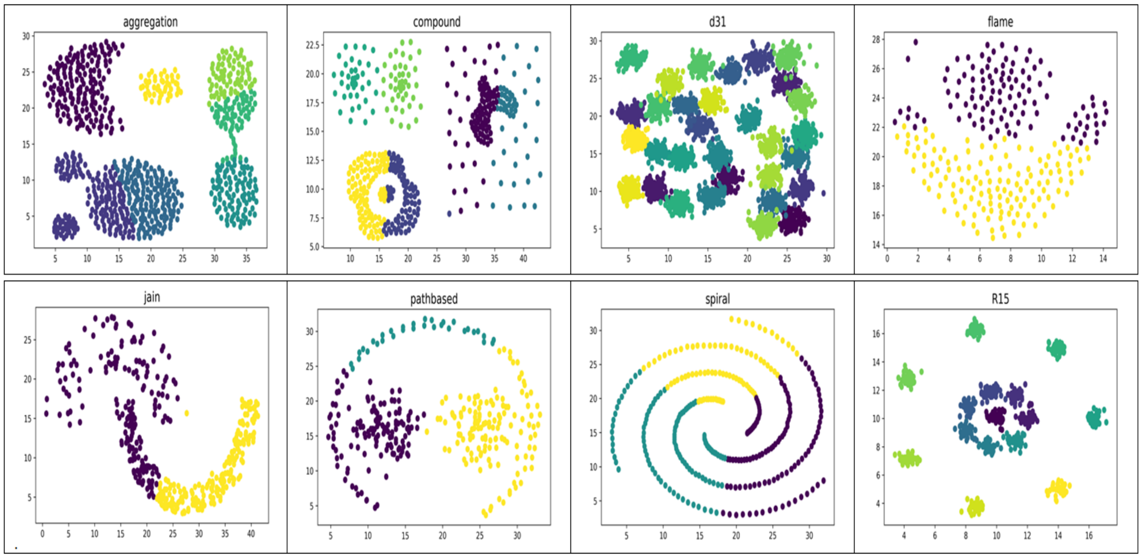

Figure 9: Clustering datasets using CLARA

The pseudo-code of CLARA algorithm is as follows:

```
Algorithm 5    CLARA
Input          - X - Dataset
               - C - number of clusters
Output         C clustered data
Step-1:        - Randomly create multiple subsets with fixed size
Step-2:        - Compute PAM algorithm on each subset
               - Select the corresponding number of representative objects (medoids).
               - Assign each object of the dataset to the closest medoid.
Step-3:        - Measure of the goodness of the clustering (e.g., mean of the dissimilarities of the objects to their
                 closest medoid).
Step-4:        - Repeat
                     - Step-1.
               - Until
                     - The goodness in Step-3 is found.
```

4.6 PAM

PAM algorithm is based on the search for number of representative observations (medoids) among the observations of the data set. As their name evoked, these observations should represent the various aspects of the data structure. After finding a set of k medoids, the k clusters are built by assigning each observation of the dataset to the nearest medoid. Fig. 10 shows how PAM clusters the datasets. The PAM algorithm consists of two phases, BUILD and SWAP, as follows:

```
Algorithm 6    PAM
Input          - X - Dataset
               - K - number of clusters
Output         - K clustered data
               Phase 1: BUILD (finds the K objects)
Step-1:        - Repeat
                     - Consider an object i (not yet been selected)
Step-2:              - Consider a non-selected object j
Step-3:              - Calculate the difference between dissimilarity d(j , i) with object i. and the
                       dissimilarity D_j with the most similar previously selected object:
                           $C_{ji} = \max\,(D_j - d(j, i), 0)$
                     - If this difference > 0, object j contributes on the decision of selecting object i.
Step-4:              - Calculate the gain obtained from the selection of object i in Step-3:
                           $\sum_j C_{ji}$
Step-5:              - The not yet selected object i should be selected to:
                           $\underset{i}{maximizes} \sum_j C_{ji}$
Step-6:        - Until
                     - K objects have been found
               Phase 2: Swap (improves the set of medoids and the clustering yielded by this set)
Step-7:        - Consider all pairs of objects (h, i), which object h has not been selected and object i has been
                 selected.
               - Consider a non-selected object j and calculate the contribution $C_{jih}$ to the swap
Step-8:        - Calculate the summation of the contributions
                           $T_{ih} = \sum_j C_{jih}$
```

```
Step-9:      – Select the pair ($h$, $i$) which $\underset{i,h}{minimizes}\ T_{ih}$
Step-10:     – If ($T_{ih}$< 0) then
                 – Go to Step-7
             – Otherwise
                 – The objective value cannot be decreased
                 – Stop
             – End if
```

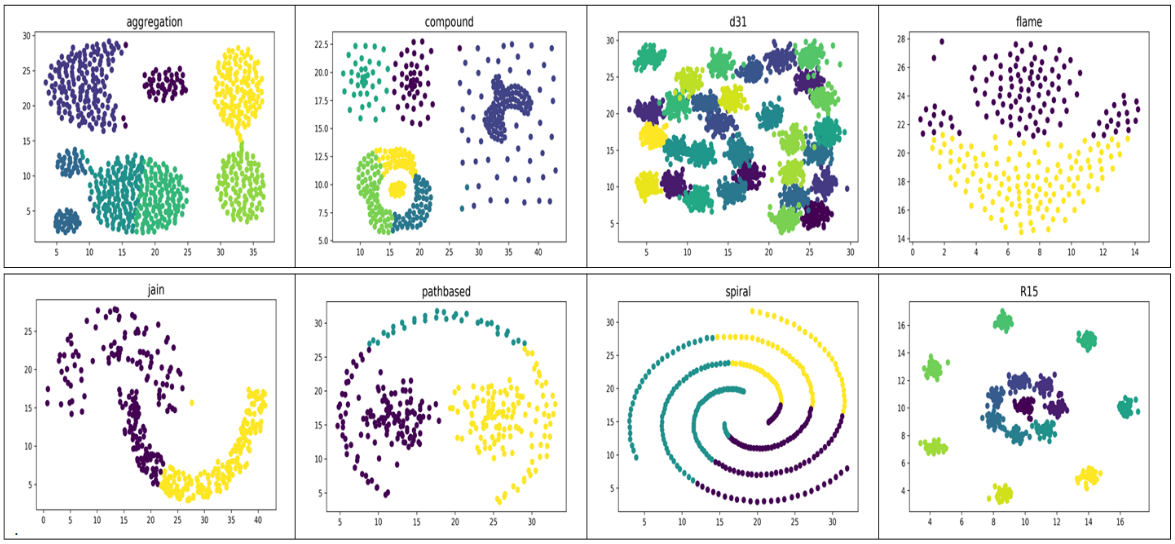

Figure 10: Clustering datasets using PAM

4.7 Agglomerative

In the agglomerative hierarchical clustering, each object represents a cluster; each data point is treated as a singleton cluster. The desired cluster structure is obtained by successively merging the clusters according to fixed rules [82,83]. Fig. 11 shows how agglomerative clusters the datasets. The pseudo-code of agglomerative algorithm is as follows:

```
Algorithm 7   Agglomerative
Input         - X – Dataset
              - $C$ - number of clusters
Output        - $C$ clustered data
Step-1:       – Start with a sample $x_i$, $i = 1, \dots, n$, considered as $n$ singleton clusters
Step-2:       – Define a dissimilarity $d$ for all pairs of disjoint
Step-3:       – Among all possible pairs of singletons; find the minimum dissimilarity $d(x_i, x_j)$ and join the $x_i$ and $x_j$ singletons.
Step-4:       – There are $n-1$ clusters $C_{1,1}, \dots, C_{1,n-1}$. Find
                  $(j^*, j'^*) = \underset{j \neq j'}{argmin}\ d(C_{1,j}, C_{1,j'})$
              – Merge $C_{1,j^*}, C_{1,j'^*}$
Step-5:       – If (the desired number of clusters is obtained) then
                  – Stop.
              – Otherwise
                  – Go to Step-4.
              – End if
```

4.8 BIRCH

BIRCH builds a dendrogram which known as clustering feature (CF) tree. It uses tree structures to partition objects hierarchically where the low-level node can be considered as "microclusters". BIRCH then applies other clustering methods to do macroclustering on the microclusters. CF-tree is based on two parameters: threshold T and branching factor B (maximum number of the children per nonleaf node). When a data point is encountered, the algorithm starts from the root and chooses the closest node at each level. If the closest leaf group for this point is finally identified, the algorithm tests whether this point belongs to the elected cluster or not. If not, a new cluster is generated with a diameter greater than the predefined T (maximum diameter of the sub-clusters stored at leaf nodes of the CF-tree). T and B control the resulting size of the tree. Using a clustering feature, useful statistics of a cluster can easily be derived, for example:

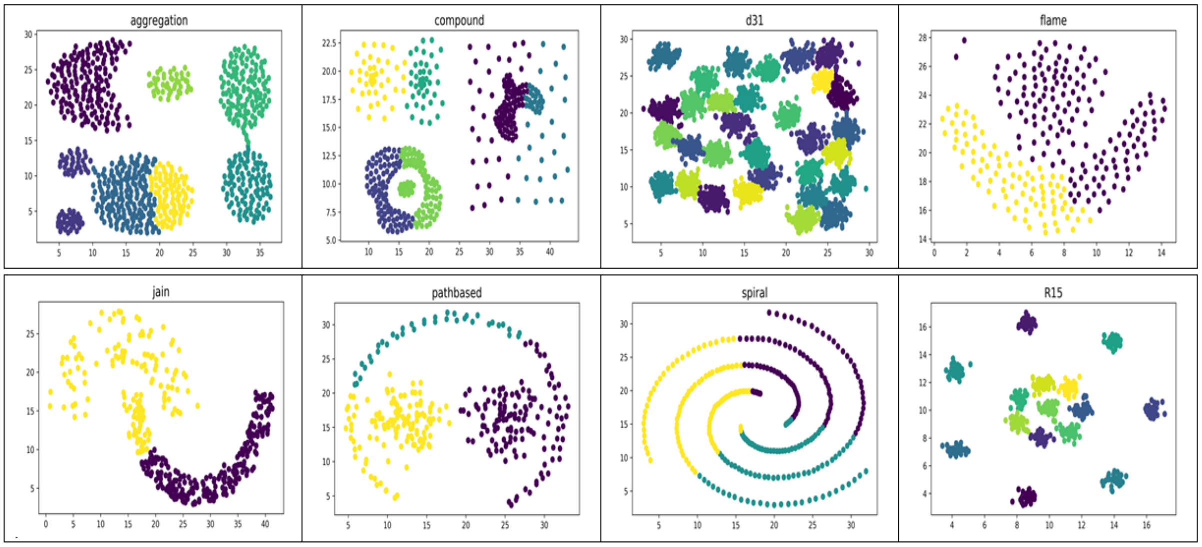

Figure 11: Clustering datasets using agglomerative

- Cluster's centroid, x_0:

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{LS}{n} \qquad (1)$$

- Radius, R:

$$R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_0)^2}{n}} = \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}} \qquad (2)$$

- Diameter, D:

$$D = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(x_i - x_j\right)^2}{n(n-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}} \qquad (3)$$

Fig. 12 shows how BIRCH clusters the datasets. The pseudo-code of BIRCH algorithm is as follows:

```
Algorithm 8    BIRCH

Input          – X – Dataset
               – C – Number of clusters
               – T – Maximum diameter of a cluster R
               – Branching factor
Output         – C clustered data
               – Clustering feature (CF): 3-D vector of information about the clusters of objects, CF = $\langle n, LS, SS\rangle$, LS:
                 The linear sum of n points ($\sum_{i=1}^{n} x_i$), SS: Square sum of data points ($\sum_{i=1}^{n} x_i^2$)

Step-1:        – Load the data into memory: An initial CF-tree is created
Step-2:        – Condense data: A larger T is used to rebuild the CF-tree.
Step-3:        – Global clustering: Apply the existing clustering method on CF leaves.
Step-4:        – From Step-3 reassign new data points to the closest centroid and perform additional passes through the
                 dataset.
```

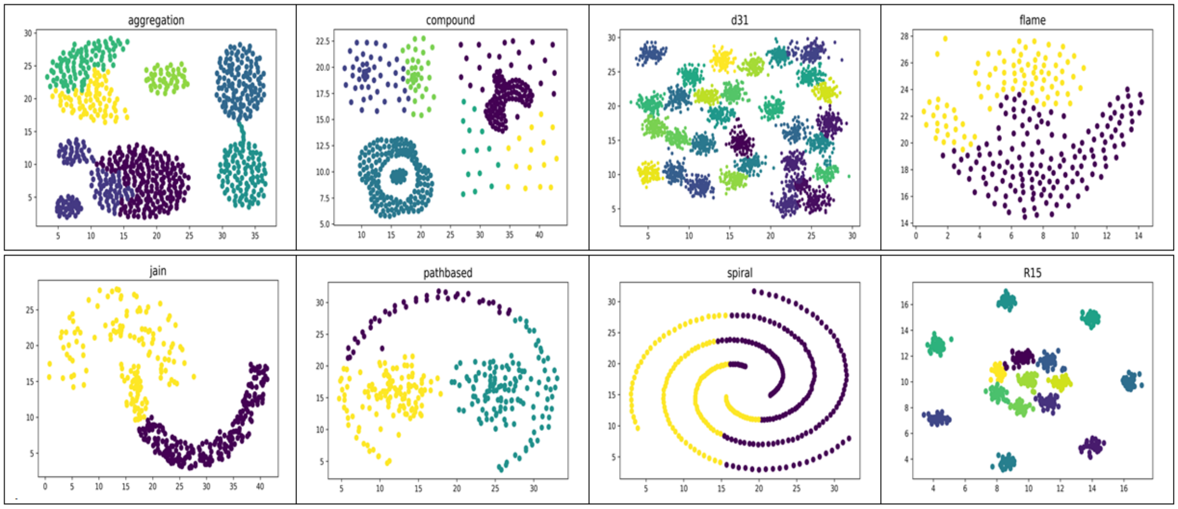

Figure 12: Clustering datasets using BIRCH

4.9 CUREs

CURE is more robust to outliers, and identifies the clusters that have wide variances in size and non-spherical shapes. This can be achieved by representing each cluster by a fixed number of points. Scattered points from the cluster are selected to generate these points. CURE applies a specified fraction to shrink the points toward the cluster's center. Having an abundance of representative points per the cluster helps CURE to modify well to the geometry of the non-spherical shapes, also the shrinking helps to weaken the effects of outliers. CURE employs a combination of partitioning and random sampling. CURE partitions a random sample selected from the dataset and partially clusters each partition, then, it clusters the partial clusters in a second pass to introduce the desired clusters. CURE algorithm is divided into initialization and completion phases. The clustering steps using Cure are described in Fig. 13.

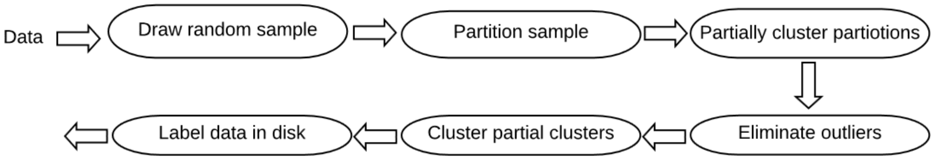

Figure 13: Overview of CURE

Fig. 14 shows how CURE clusters the datasets. The pseudo-code of CURE algorithm is as follows:

```
Algorithm 9    CURE
Input          - X – Dataset
               - $K$ – Number of clusters
Output         - $K$ clustered data
                 Phase 1: Initialization
Step-1:        − Draw a random sample from the dataset.
Step-2:        − Partition this sample into $p$ equal-sized partitions.
Step-3:        − In each partition, cluster the points into $\frac{m}{pq}$ clusters, where $m$ is the number of points, $q$ is the desired
                 reduction of the points in a partition, using CURE's clustering algorithm to obtain the total of $\frac{m}{q}$ clusters.
Step-4:        − Eliminate outliers (first stage of outlier elimination).
Step-5:        − Use CURE's clustering algorithm to cluster the $\frac{m}{q}$ clusters found in Step-3 until only $K$ clusters remain.
Step-6:        − Eliminate outliers (second stage of outlier elimination).
                 Phase 2: Completion
Step-7:        Remaining data points will be assigned to the nearest cluster to get a complete clustering.
```

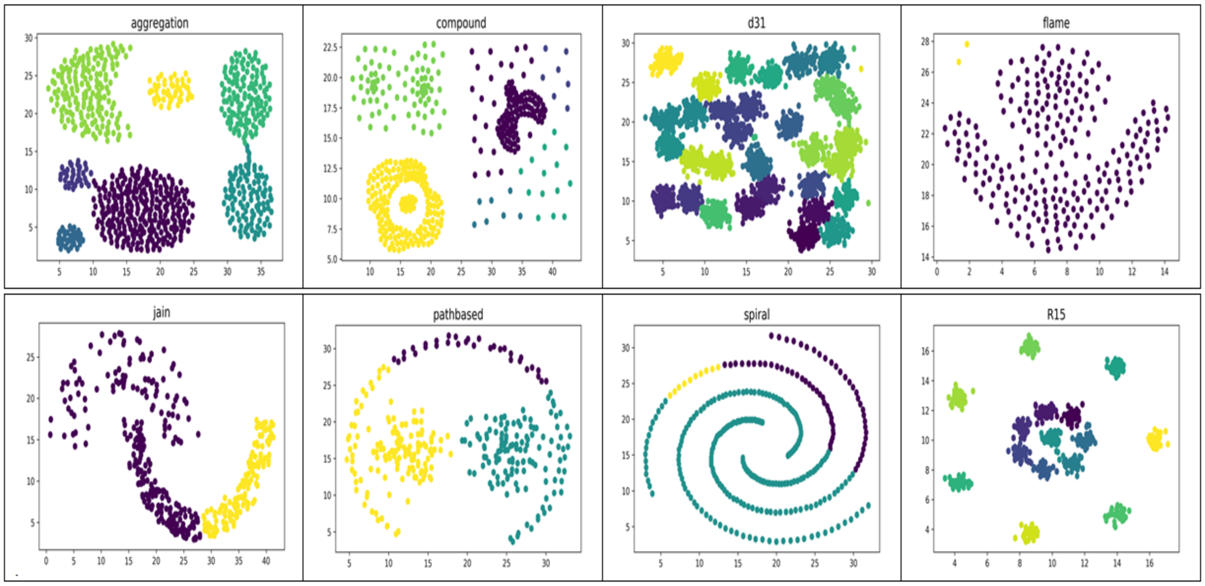

Figure 14: Clustering datasets using CURE

5 Experimental Implementation

5.1 Clustering Validation Indices

Several performance indices are used for cluster evaluation [84]. Indices measure the correspondence between two clusters of the same dataset and are based on how the pairs of objects are clustered [85]. Seven common indices are discussed in the following subsections.

5.1.1 Adjusted Rand Index

The Rand Index (RI) [86,87] is used in comparing an induced clustering structure (C1) a given clustering structure (C1) with an induced clustering structure (C2). Assume that a is the number of pairs of cases that are assigned to the same group in C1 and in the same group in C2; b be the number of pairs of cases that are in the same group in C1, but not in the same group in C2; c be the number of cases that are in the same group in C2, but not in the same group in C1; and d be the number of pairs of cases assigned to different clusters in C1 and C2. The quantities d and a can be considered as agreements, and c and b as disagreements. The RI is defined as:

$$RI = \frac{a+d}{a+b+c+d} \tag{4}$$

RI lies between 0 and 1. When the two clusters agree perfectly, the RI is 1. A problem with RI is that the expected value of two random groups does not take a constant value (e.g., zero) [88]. Adjusted Rand Index (ARI) is suggested to overcome this disadvantage [89,90]. ARI is an improvement of RI and can be computed by

$$ARI = \frac{\binom{n}{2}(a+d)-[(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2-[(a+b)(a+c)+(c+d)(b+d)]} \tag{5}$$

With maximum value 1 and expected value zero. Where n is the number of objects. ARI measures not only the correct separation of objects belonging to different clusters but also the relation between the objects of the same cluster. ARI pays attention to the relation between objects than to the relation between each object and its target label. ARI evaluates the ability of the method to separate the objects belonging to different clusters [85].

5.1.2 Fowlkes-Mallows Index

Fowlkes-Mallows Index (FMI) [91] measures the similarity of two clusters of a set of points. A higher value for the FMI means a greater similarity between the clusters. FMI is defined as the geometric mean between the recall and precision:

$$FM = \sqrt{\frac{TP}{TP+FP} . \frac{TP}{TP+FN}} \tag{6}$$

where FP is the number of False Positive, TP is the number of True Positive, and FN is the number of False Negative. FMI lies between 0 and 1; a high value means a good similarity between the clusters. The validity of FMI can be tested by comparing unrelated two clusterings which the value of FMI approaches zero making FMI is an accurate representation for unrelated data. If the dataset contains noise, the value of FMI decreases. Thus, FMI is a reliable index for measuring the similarity between two clusters.

5.1.3 Normalized Mutual Information

Normalized Mutual Information (NMI) measures the consistency between two clusters. Let two clusters $C^a = \{C_1^a, C_2^a, \dots, C_{k_a}^a\}$ and $C^b = \{C_1^b, C_2^b, \dots, C_{k_b}^b\}$ with k_a and k_b clusters, respectively, the NMI is defined as [92]:

$$NMI\left(c^a, c^b\right) = \frac{-2\sum_{i=1}^{k_a}\sum_{j=1}^{k_b} n_{ij}^{ab} \log\left(\frac{n_{ij}^{ab}\, n}{n_i^a\, n_j^b}\right)}{\sum_{i=1}^{k_a} n_i^a \log\left(\frac{n_i^a}{n}\right) + \sum_{j=1}^{k_b} n_j^b \log\left(\frac{n_j^b}{n}\right)} \tag{7}$$

where *n* is the total number of the patterns in the dataset, n_i^a represents the number of the patterns in cluster $C_i^a \in P^a$, n_{ij}^{ab} denotes the number of the shared patterns between clusters $C_i^a$and C_j^b, $C_i^a \in C^a$ and $C_j^b \in P^b$.

5.1.4 Homogeneity, Completeness, and V-measure

Homogeneity analysis [93] is a well-known *technique* for optimizing the homogeneity of the variables forms of simplification and manipulation. Historically, the idea of the homogeneity is related to the idea in which different features may measure the same thing. Thus, the number of features can be reduced or a lower weight for such feature a feature is put to get fair cases comparing other features [94,95]. Homogeneity and completeness scores are formally given by:

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{8}$$

where $H(C|K)$ is the classes' conditional entropy and is given by:

$$H(C|K) = \sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n}.\ \log\left(\frac{n_{c,k}}{n_k}\right) \tag{9}$$

and $H(C)$ is the classes' entropy and is given by:

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n}\ .\log\left(\frac{n_c}{n}\right) \tag{10}$$

where n is the total number of the instances, $n_{c,k}$ is the number of instances from the class c assigned to the cluster k, n_k and n_c are the number of instances belonging to class k and cluster c respectively. The entropy of clusters $H(K)$ and the conditional entropy of clusters given class $H(K|C)$ are defined in a symmetric manner. The clustering completeness is satisfied when all the data points which are elements of a given class are members of the same cluster. Completeness is symmetrical to the homogeneity. A clustering result satisfies completeness if all the data-points that are members of a given class are elements of the same cluster. Completeness is given by:

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{11}$$

V-measure [96,97] is based upon homogeneity and completeness, and defined as the harmonic mean of completeness and homogeneity:

$$v = 2\ .\ \frac{h.c}{h+c} \tag{12}$$

The specifications of the computer used to carry out the experiments are: Intel Core i5-2400 (3.10 GHz) processor, 16 GB memory, 1 TB HDD, R (version 3.5.2) and Python (version 3.7) programming language, and Gnu/Linux Fedora 28 OS. Tab. 4 shows the performance measures comparison. Tab. 5 shows the clustering validation indices comparison from the point of view of time, ARI, FMI, NMI, Homogeneity, Completeness, and V-Measure.

Table 5: Clustering validation indices comparison. The first column is the dataset name, the second column is the algorithm used, and from the third column to the ninth column are the performance indices (bold font indicates better value)

Dataset	Algorithm	Time (lower is better)	ARI (higher is better)	FMI (higher is better)	NMI (higher is better)	Homogeneity (higher is better)	Completeness (higher is better)	V-Measure (higher is better)
Aggregation	K-means	0.1468	0.7624	0.8159	0.8805	0.9287	0.8347	0.8792
	Agglomerative	0.0953	0.8133	0.8559	0.9214	0.9647	0.8799	0.9204
	BIRCH	0.5176	0.7334	0.7900	0.8475	0.8863	0.8103	0.8466
	Cure	0.0563	**0.9935**	**0.9949**	**0.9896**	**0.9898**	**0.9893**	**0.9896**
	X-means	0.0579	0.7321	0.7890	0.8510	0.8905	0.8132	0.8501
	FCM	0.0649	0.7333	0.7895	0.8395	0.8734	0.8070	0.8389
	CLARANS	157.2657	0.6085	0.6883	0.7817	0.8107	0.7538	0.7812
	CLARA	**0.0030**	0.6793	0.7457	0.8271	0.8610	0.7946	0.8265
	PAM	0.4070	0.7745	0.8255	0.8890	0.9361	0.8442	0.8878
compound	K-means	0.1071	0.4975	0.6181	0.6718	0.7428	0.6075	0.6684
	Agglomerative	0.1460	0.5178	0.6344	0.6994	0.7706	0.6348	0.6961
	BIRCH	0.0367	0.7833	0.8449	0.8080	**0.8101**	0.8059	0.8080
	Cure	0.0110	**0.8355**	**0.8861**	**0.8420**	0.8071	**0.8783**	**0.8412**
	X-means	0.5594	0.7218	0.7958	0.7414	0.7716	0.7124	0.7408
	FCM	0.0066	0.4952	0.6162	0.6623	0.7336	0.5979	0.6588
	CLARANS	117.6188	0.6651	0.7562	0.6428	0.6655	0.6210	0.6424
	CLARA	**0.0030**	0.5074	0.6261	0.6724	0.7417	0.6095	0.6692
	PAM	0.0750	0.4752	0.6030	0.6974	0.7594	0.6405	0.6949

d31	K-means	0.0095	0.9529	0.9544	0.9672	0.9672	0.9673	0.9672
	Agglomerative	**0.0068**	0.9201	0.9227	0.9508	0.9506	0.9511	0.9508
	BIRCH	0.0082	0.8836	0.8874	0.9404	0.9380	0.9428	0.9404
	Cure	0.0206	0.7751	0.7968	0.9279	0.8906	0.9668	0.9272
	X-means	0.1290	0.5004	0.5681	0.8181	0.7452	0.8982	0.8146
	FCM	0.0617	0.8528	0.8581	0.9348	0.9305	0.9392	0.9348
	CLARANS	44.8495	0.5673	0.5928	0.8239	0.7990	0.8495	0.8235
	CLARA	0.0750	0.0750	0.0750	0.0750	0.0750	0.0750	0.0750
	PAM	54.5050	**0.9993**	**0.9994**	**0.9995**	**0.9995**	**0.9995**	**0.9995**
flame	K-means	0.5295	0.4534	0.7364	0.3989	0.4101	0.3881	0.3988
	Agglomerative	0.0338	0.1872	0.6237	0.3297	0.3291	0.3303	0.3297
	BIRCH	0.0561	0.3851	0.7139	0.2780	0.2784	0.2775	0.2780
	Cure	0.0922	0.0128	0.7300	0.0479	0.0130	0.1767	0.0242
	X-means	0.0470	0.4649	0.7417	0.4269	0.4391	0.4150	0.4267
	FCM	0.0441	0.4880	0.7530	0.4422	0.4547	0.4301	0.4420
	CLARANS	17.8446	0.1871	0.6497	0.1137	0.1074	0.1204	0.1135
	CLARA	**0.0010**	**0.9964**	**0.9966**	**0.9971**	**0.9971**	**0.9971**	**0.9971**
	PAM	0.0050	0.5116	0.7646	0.4582	0.4708	0.4459	0.4580
jain	K-means	0.2240	0.3241	0.7005	0.3706	0.4070	0.3375	0.3690
	Agglomerative	0.0189	0.5146	0.7904	0.5068	0.5492	0.4678	0.5052
	BIRCH	0.5982	0.5016	0.7048	0.4968	0.5379	0.4387	0.4799
	Cure	0.0111	0.0279	0.5893	0.2235	0.2377	0.2102	0.2231
	X-means	0.0184	0.3241	0.7005	0.3706	0.4070	0.3375	0.3690
	FCM	0.0146	0.2658	0.6101	0.3888	0.5380	0.2809	0.3691
	CLARANS	327.2452	**0.5498**	**0.8558**	0.4701	0.4060	0.5442	0.4651
	CLARA	**0.0010**	0.4752	0.6030	**0.6974**	**0.7594**	**0.6405**	**0.6949**
	PAM	0.0160	0.2607	0.6712	0.3348	0.3682	0.3045	0.3333
pathbased	K-means	0.0142	0.4613	0.6617	0.5470	0.5128	0.5834	0.5458
	Agglomerative	0.0334	0.4847	0.6738	0.5671	0.5374	0.5985	0.5663
	BIRCH	0.0418	0.4789	0.6695	0.5519	0.5240	0.5813	0.5512
	Cure	0.0326	0.4572	0.6598	0.5437	0.5085	0.5814	0.5425
	X-means	0.1676	0.4618	0.6620	0.5475	0.5133	0.5839	0.5463
	FCM	0.0142	0.3990	0.6519	0.5039	0.4008	0.6337	0.4910
	CLARANS	1.4503	0.0276	0.5118	0.1648	0.1133	0.2398	0.1539
	CLARA	**0.0010**	**0.9993**	**0.9994**	**0.9995**	**0.9995**	**0.9995**	**0.9995**
	PAM	0.0090	0.4582	0.6604	0.5445	0.5092	0.5822	0.5433
spiral	K-means	0.0124	-0.0057	0.3277	0.0007	0.0007	0.0007	0.0007
	Agglomerative	0.0278	-0.0057	0.3277	0.0007	0.0007	0.0007	0.0007
	BIRCH	0.0181	0.0130	0.4196	0.0155	0.0122	0.0197	0.0150
	Cure	0.0370	0.0680	0.4951	0.1672	0.1278	0.2188	0.1614
	X-means	0.0661	-0.0060	0.3276	0.0004	0.0004	0.0004	0.0004
	FCM	0.0351	-0.0060	0.3274	0.0005	0.0005	0.0005	0.0005
	CLARANS	39.8868	0.0064	0.3705	0.0136	0.0128	0.0144	0.0136
	CLARA	**0.0020**	**0.2607**	**0.6712**	**0.3348**	**0.3682**	**0.3045**	**0.3333**
	PAM	0.0170	-0.0031	0.3341	0.0029	0.0028	0.0029	0.0029

R15	K-means	0.7554	**0.9928**	**0.9932**	**0.9942**	**0.9942**	**0.9942**	**0.9942**
	Agglomerative	0.0204	**0.9928**	**0.9932**	**0.9942**	**0.9942**	**0.9942**	**0.9942**
	BIRCH	0.0327	0.9717	0.9735	0.9810	0.9808	0.9811	0.9810
	Cure	0.0212	0.9821	0.9833	0.9873	0.9872	0.9873	0.9873
	X-means	0.0452	**0.9928**	**0.9932**	**0.9942**	**0.9942**	**0.9942**	**0.9942**
	FCM	0.0342	0.8853	0.8936	0.9496	0.9432	0.9560	0.9496
	CLARANS	0.1610	0.6351	0.6698	0.8340	0.8078	0.8610	0.8335
	CLARA	**0.0150**	0.4582	0.6604	0.5445	0.5092	0.5822	0.5433
	PAM	0.0160	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

5.2 Experimental Results and Discussion

This section discusses and compares between the clustering validation measures. The clustering algorithm is considered to be efficient if it achieves higher ARI, FMI, NMI, Homogeneity, Completeness, and V-Measure in less time.

5.2.1 Time and Performance Indices Analysis

For Aggregation, compound, flame, jain, pathbased, spiral, and R15 datasets, CLARA has the least time, and behaves somewhat similar to k-means in d31 dataset. However, its clustering performances are low for all datasets. CLARANS has the worst time; however, the clustering performances are not the best. For Aggregation and compound datasets, cure gives the best clustering performances and its time behaves somewhat similar to the time of CLARA t has the lowest time. For d31 dataset, PAM has worst time and clustering performances are very close to 1. For flame dataset, CLARA has the least time and its clustering performances close to 1. For jain dataset, CLARANS has the worst time, however, it gives better ARI and FMI, CLARA has the best time and gives better NMI, Homogeneity, Completeness, and V-Measure. For pathbased and spiral datasets, CLARA gives the best clustering performances with the least time. For R15, K-means, X-means, and Agglomerative give the same clustering performances, the time of each X-means and Agglomerative is similar somewhat to the time of CLARA that has the least time, and K-means has the longest time.

5.2.2 Discussion

The partitional algorithms, K-means, X-means, CLARA, CLARANS, PAM, and FCM are applicable to the numerical datasets. In addition, the partitional algorithms are unable to handle the outliers and noise. Moreover, these algorithms are based on a specific assumption of portioning the dataset. Thus, the algorithms need to determine the number of the clusters in advance. CLARANS needs more parameters such as the maximum number of neighbors of the node. The clustering process creates set of some representative points of the extracted clusters. Depending on the algorithm, these points may be the medoids or the centers.

The objective of K-Means is to minimize the distance of the representative point of a cluster from the objects within this cluster, while PAM is a medoid-based. The clustering criterion of CLARANS and CLARA is medoid-based, which means that the objective of them is to minimize the distance of the representative point (i.e., medoid) of a cluster from the objects within this cluster. CLARANS and CLARA apply clustering of the dataset; therefore, they may treat larger datasets than PAM. More specifically, CLARA considers multiple samples of the dataset on hand and applies PAM on each sample. Then, the output is the best clustering.

The main issue with this approach is that the efficiency depends on the size of the sample. Also, the samples of a dataset influence the clustering results. Thus, if a sample is biased, a good clustering resulted based on some samples will not necessarily represent the good clustering of the whole dataset. CLARANS is a mixture of CLARA and PAM. A key difference between PAM and CLARANS is that CLARANS searches a subset of the dataset in order to define clusters. Unlike CLARA which has a fixed sample at every stage, CLARANS draws the subsets with some randomness in each stage of the search. CLARANS is more scalable and efficient than both PAM and CLARA.

The algorithms described above consider that the object may belong to only one cluster, therefore, they are considered to be crisp clustering algorithms. In the real-life cases, it is hard to define the boundaries of a cluster. FMC, which is based on K-means clustering criterion, is a representative method of the fuzzy clustering which is an uncertainty-based technique. FCM assigns the object to a cluster with a degree of belief. Unlike K-means which is given a specific number of the clusters *K* and depends on once-only placement of the initial centroids, X-means had to search for this number in the range [2…K] and gradually adds new centroids in the areas where they are added.

One issue with X-means is that the splitting criterion is an experimental choice, for example, BIC performs well for some test-sets and applications and Minimum Description Length (MDL) [98] and Akaike Information Criterion (AIC) [99] perform well for others. Hierarchical clustering methods are more efficient in handling outliers and noise than partitional algorithms. BIRCH uses CF-tree hierarchical data structure for multiphase clustering. In BIRCH, a good clustering is yielded from a single scan. To improve the quality, one or more additional scans can be used. BIRCH handles only numerical data, and may create different clusters it is order-sensitive (i.e., different clusters may be generated for different orders of the same data).

When redistributing the objects in the final phase, BIRCH uses only the centroid of the cluster, thus, if the clusters do not have uniform shape and size, BIRCH does not perform well. On the other hand, CURE utilizes a combination of partitioning and random sampling to handle large datasets. CURE identifies clusters having wide variances in size and non-spherical shapes using multiple points to represent each cluster. These representative points of the cluster are generated by selecting the well-scattered points from the cluster of interest and shrinking these selected points toward the centre of this cluster by a specified fraction.

On the other hand, CURE is sensitive to some parameters (e.g., shrink factor used for handling outliers, number of partitions, and number of representative points). Thus, the selection of these parameters influences the quality of the clustering results. Agglomerative algorithm produces a series of clustering schemes of decreasing the number of clusters at east step. Merging the two closest clusters into one at each step results a new scheme [100].

5.2.3 Conclusion

Cluster analysis is one of the most important tasks in various research fields. The major aim of clustering is to extract and identify significant groups of objects in underlying dataset. Thus, clustering criterion is the basis of the clustering so that the data points in the same cluster are closer to each other than data points in other clusters. Since clustering is applied in various fields, a number of clustering algorithms and techniques have been proposed. The main characteristics of clustering algorithms have been presented in this paper.

Moreover, the classifications of the algorithms belonging to different categories have been discussed, and the representative algorithms of each category have been presented. The author concluded the discussion on the clustering algorithms by a comparative presentation. The benchmark datasets are clustered and the number of clusters is known in advance, therefore, the clustering algorithms used do not determine the number of clusters but take the number of clusters as a parameter. Moreover, an important issue the author discussed is the cluster validity. The comparison was done from the point of view seven performance measures.

Empirically, the author conducted extensive experiments where the most representative approaches from each of the categories have been compared using various datasets. The author measured the effectiveness of the candidate algorithms through a number of tests. In addition, the author highlighted the set of algorithms that are the best performing.

Funding Statement: This project was supported financially by the Academy of Scientific Research and Technology (ASRT), Egypt, Grant No. (6475), (ASRT) is the 2nd affiliation of this research.

Conflicts of Interest: The author declares that there is no conflict of interest regarding the publication of this article.

References

[1] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[2] P. Lasek and J. Gryz, "Density-based clustering with constraints," *Computer Science and Information Systems*, vol. 16, no. 2, pp. 469–489, 2019.

[3] J. Hämäläinen, S. Jauhiainen and T. Kärkkäinen, "Comparison of internal clustering validation indices for prototype-based clustering," *Algorithms*, vol. 10, no. 3, pp. 105, 2017.

[4] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine," *Artificial Intelligence,* vol. 97, no. 1–2, pp. 245–271, 1997.

[5] Y. Wang, Y. Fan, P. Bhatt and C. Davatzikos, "High-dimensional pattern regression using machine learning: from medical images to continuous clinical variables," *Neuroimage*, vol. 50, no. 4, pp. 1519–1535, 2010.

[6] I. H. Witten, E. Frank and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, *Third Edition*. Burlington, Mass.: Morgan Kaufmann Publishers. 2011. [Online]. Available: http://library.oum.edu.my/oumlib/content/catalog/718245.

[7] M. Z. Rodriguez, H. C. Cesar, C. Dalcimar, M. B. Odemir, R. A. Diego *et al.,* "Clustering algorithms: A comparative approach," *PLoS One*, vol. 14, no. 1, pp. 1–34, 2019.

[8] S. Theodoridis and K. Koutroumbas, "Pattern recognition," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 376, 2008.

[9] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA, 2010. [Online]. Available: https://doi.org/10.1007/b107408.

[10] F. Adil, A. Najlaa, T. Zahir, A. Abdullah, K. Ibrahim *et al.,* "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, 2014.

[11] L. Liu and M. T. Özsu, *Encyclopedia of Database Systems*. Boston, Springer. 2009. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9.

[12] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 652–657, 2011.

[13] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? Answers via model-based cluster analysis," *Computer Journal*, vol. 41, no. 8, pp. 578–588, 1998.

[14] J. W. Han, M. Kamber and J. Pei, *Data Mining: Data Mining Concepts and Techniques*. San Fransisco, Morgan Kaufmann. 2012. [Online]. Available: https://doi.org/10.1016/C2009-0-61819-5.

[15] W. Kwedlo, "A clustering method combining differential evolution with the k-means algorithm," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1613–1621, 2011.

[16] L. Morissette and S. Chartier, "The k-means clustering technique: general considerations and implementation in mathematica," *Tutorials in Quantitative Methods for Psychology*, vol. 9, no. 1, pp. 15–24, 2013.

[17] X. Jin and J. Han, "K-medoids clustering," in *Encyclopedia of Machine Learning and Data Mining*. Springer, Boston, MA, pp. 697–700, 2017.

[18] Z. Huang and M. K. Ng, "A note on k-modes clustering," *Journal of Classification*, vol. 20, no. 2, pp. 257–261, 2003.

[19] A. P. Reynolds, G. Richards and V. J. Rayward-Smith, "The application of k-medoids and pam to the clustering of rules," in *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, Exeter, UK, vol. 3177, pp. 173–178, 2004.

[20] P. T. Nguyen, K. Eckert, A. Ragone and T. Di Noia, "Modification to k-medoids and CLARA for effective document clustering," in *Int. Sym. on Methodologies for Intelligent Systems 2017, Lecture Notes in Computer*

Science, Warsaw, Poland, vol. 10352, pp. 481–491, 2017.

[21] 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, 2002.

[22] J. C. Bezdek, R. I. Ehrlich and W. E. Full, “FCM: The fuzzy c-means clustering algorithm,” *Computers and Geoences*, vol. 10, no. 2, pp. 191–203, 1984.

[23] C. C. Aggarwal, “A survey of stream clustering algorithms,” in *Data Clustering Algorithms and Application*, New York, Chapman and Hall, pp. 231–258, 2013.

[24] C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, “A framework for clustering evolving data streams,” in *Proc. 2003 VLDB Conf.*, Berlin, Germany, pp. 81–92, 2003.

[25] R. W. Sembiring, J. M. Zain and A. Embong, “A comparative agglomerative hierarchical clustering method to cluster implemented course,” *Journal of Computers*, vol. 2, no. 12, pp. 1–6, 2010.

[26] S. Guha, R. Rastogi and K. Shim, “Rock: A robust clustering algorithm for categorical attributes,” *Information Systems*, vol. 25, no. 5, pp. 345–366, 1999.

[27] T. Zhang, R. Ramakrishnan and M. Livny, “BIRCH: An efficient data clustering method for very large databases,” *Acm Sigmod Record*, vol. 25, no. 2, pp. 103–114, 1996.

[28] R. Xu and D. C. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[29] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proc. of the Second Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, Portland, Oregon, USA, vol. 96, no. 34, pp. 226–231, 1996.

[30] M. Ankerst, M. M. Breunig, H. Kriegel and J. Sander, “Optics: Ordering points to identify the clustering structure,” in *ACM Sigmod Int. Conf. on Management of Data*, Philadelphia, PA, USA, pp. 49–60, 1999.

[31] G. Kou, Y. Peng and G. Wang, “Evaluation of clustering algorithms for financial risk analysis using MCDM methods,” *Information Science*, vol. 275, pp. 1–12, 2014.

[32] T. Kinnunen, I. Sidoroff, M. Tuononen and P. Fränti, “Comparison of clustering methods: A case study of text-independent speaker modeling,” *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1604–1617, 2011.

[33] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir and A. Schliep, “Clustering cancer gene expression data: a comparative study,” *BMC Bioinformatics*, vol. 9, pp. 1–14, 2008.

[34] I. G. Costa, F. de A. T. de Carvalho and M. C. P. de Souto, “Comparative analysis of clustering methods for gene expression time course data,” *Genetics and Molecular Biology*, vol. 27, no. 4, pp. 623–631, 2004.

[35] L. Parsons, E. Haque and H. Liu, “Evaluating subspace clustering algorithms,” in *Workshop on Clustering High Dimensional Data and Its Applications, SIAM Int. Conf. on Data Mining*, Lake Buena Vista, Florida, pp. 48–56.

[36] U. Maulik and S. Bandyopadhyay, “Performance evaluation of some clustering algorithms and validity indices,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[37] S. Brohée and J. van Helden, “Evaluation of clustering algorithms for protein-protein interaction networks,” *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–19, 2006.

[38] A. C. Benabdellah, A. Benghabrit and I. Bouhaddou, “A survey of clustering algorithms for an industrial context,” *Procedia Computer science*, vol. 148, pp. 291–302, 2019.

[39] C. M. M. Pereira and R. F. de Mello, “A comparison of clustering algorithms for data streams,” in *Int. Conf. on Integrated Computing Technology*, Springer, Berlin, Heidelb, pp. 59–74, 2011.

[40] M. Meilă and D. Heckerman, “An experimental comparison of model-based clustering methods,” *Machine Learning*, vol. 42, no. 1–2, pp. 9–29, 2001.

[41] T. S. Madhulatha, “Comparison between k-means and k-medoids clustering algorithms,” in *Advances in Computing and Information*, Chennai, India, vol. 198, pp. 472–481, 2011.

[42] P. Nerurkar, A. Shirke, M. Chandane and S. Bhirud, “Empirical analysis of data clustering algorithms,” *Procedia Computer Science*, vol. 125, pp. 770–779, 2018.

[43] P. Fränti and S. Sieranoja, *Clustering basic Benchmark-Aggregation*. 2019. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/Aggregation.txt.

[44] A. Gionis, H. Mannila and P. Tsaparas, “Clustering aggregation,” *ACM Transactions on Knowledge Discovery*

from Data, vol. 1, no. 1, pp. 1–30, 2007.

[45] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-Compound*. 2019. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/Compound.txt.

[46] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers*, vol. C–20, no. 1, pp. 68–86, 1971.

[47] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-D31*. 2019. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/D31.txt.

[48] C. J. Veenman, M. J. T. Reinders and E. Backer, "A maximum variance cluster algorithm," *Pattern Analysis & Machine Intelligence IEEE Transactions on*, vol. 24, no. 9, pp. 1–19, 2002.

[49] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-Flame*. 2020. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/flame.txt.

[50] L. Fu and E. Medico, "Flame, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinformatics*, vol. 8, no. 3, pp. 1–15, 2007.

[51] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-Jain*. 2020. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/jain.txt.

[52] A. K. Jain and M. H. C. Law, "Data clustering: A user's dilemma," in *Int. Conf. on Pattern Recognition*, Kolkata, India, vol. 3776, pp. 1–10, 2005.

[53] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-Pathbased*. 2020. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/pathbased.txt.

[54] H. Chang and D. Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[55] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-Spiral*. 2019. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/spiral.txt.

[56] P. Fränti and S. Sieranoja, *Clustering Basic Benchmark-R15*. 2019. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/R15.txt.

[57] S. Developers, *Sklearn Cluster K-means*. 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[58] F. Farnstrom, J. Lewis and C. Elkan, "Scalability for clustering algorithms revisited," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 51–57, 2000.

[59] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[60] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of Berkeley Sym. on Mathematical Statistical and Probability*, Berkeley, vol. 1, pp. 281–296, 1967.

[61] A. Novikov, *Pyclustering xmeans*. 2019. [Online]. Available: https://pyclustering.github.io/docs/0.9.0/html/d2/d8b/namespacepyclustering_1_1cluster_1_1xmeans.html.

[62] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Pro. of 17th Int. Conf. on Machine Learning*, Stanford University, California, USA, pp. 727–734, 2000.

[63] E. Schubert and P. J. Rousseeuw, "Faster k-medoids clustering: improving the pam, clara, and CLARANS algorithms," in *International Conference on Similarity Search and Applications*, Newark, NJ, USA, vol. 11807, pp. 171–187, 2019.

[64] J. E. Gentle, L. Kaufman and P. J. Rousseuw, "Finding groups in data: an introduction to cluster analysis," *Biometrics*, vol. 47, no. 2, pp. 788, 1990.

[65] A. Novikov, *Pyclustering Fuzzy c-means*. 2019. [Online]. Available: https://pyclustering.github.io/docs/0.9.0/html/de/df0/namespacepyclustering_1_1cluster_1_1fcm.html.

[66] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, US, Springer, Boston, MA. 1981. [Online]. Available:

https://link.springer.com/book/10.1007%2F978-1-4757-0450-1#authorsandaffiliationsbook.

[67] J. C. Bezdek, "Fuzzy mathematics in pattern classification," Ph.D. dissertation, Cornell University, Ithaca, 1973.

[68] R. T. Ng and J. Han, *Efficient and Effective Clustering Method for Spatial Data Mining*. Canada, University of British Columbia. 1994. [Online]. Available: https://dl.acm.org/doi/book/10.5555/901953.

[69] S. Developers, *Sklearn Cluster Kmeans Agglomerative Clustering*. 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html.

[70] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, no. 1, pp. 7–24, 1984.

[71] S. Developers, *Sklearn Cluster BIRCH.* 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html.

[72] 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, 1997.

[73] A. Novikov, *Pyclustering BIRCH*, 2019. [Online]. Available: https://pyclustering.github.io/docs/0.9.0/html/dc/d83/namespacepyclustering_1_1cluster_1_1birch.html.

[74] S. Guha, R. Rastogi and K. S. Cure, "Cure: An efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

[75] B. Krawczyk and M. Woźniak, "Pruning ensembles of one-class classifiers with x-means clustering," in *Asian Conf. on Intelligent Information and Database Systems*, Bali, Indonesia, pp. 484–493, 2015.

[76] K. Hirose, S. Kawano, S. Konishi and M. Ichikawa, "Bayesian information criterion and selection of the number of factors in factor analysis models," *Journal of Data Science*, vol. 9, pp. 243–259, 2011.

[77] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[78] H. Shili and L. Ben Romdhane, "IF-CLARANS: Intuitionistic fuzzy algorithm for big data clustering," in *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Cádiz, Spain, vol. 854, pp. 39–50, 2018.

[79] S. Ghosh, S. Mitra and Center, "Incorporating fuzziness to CLARANS," in *Int. Conf. on Pattern Recognition and Machine Intelligence*, New Delhi, India, vol. 5909, pp. 122–127, 2009.

[80] G. Duan, W. Hu and Z. Zhang, "A novel multilayer data clustering framework based on feature selection and modified k-means algorithm," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 4, pp. 81–90, 2016.

[81] R. L. Cannon, J. V. Dave and J. C. Bezdek, "Efficient implementation of distinct the fuzzy c-means clustering algorinthms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 248–255, 1986.

[82] B. Clarke, E. Fokoue and H. H. Zhang, *Principles and Theory for Data Mining and Machine Learning*, Springer, New York, NY, 2009. [Online]. Available: https://citations.springer.com/book?doi=10.1007/978-0-387-98135-2.

[83] H. Jingshan, D. Jiangbo, M. B. Glen, E. Karen, Z. He *et al.,* "Omit: Dynamic, semi-automated ontology development for the microRNA domain," *PLoS One*, vol. 9, no. 7, pp. 1–16, 2014.

[84] S. Wagner and D. Wagner, "Comparing clusterings–An overview," *Analysis*, vol. 4769, no. 1907, pp. 1–19, 2007.

[85] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *Artificial Neural Networks–ICANN 2009*, Limassol, Cyprus, vol. 2, pp. 175–184, 2009.

[86] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850.

[87] R. K. Brouwer, "Extending the rand, adjusted rand and jaccard indices to fuzzy partitions," *Journal of Intelligent Information Systems*, vol. 32, no. 3, pp. 213–235, 2009.

[88] L. Rokach and O. Maimon, "Clustering methods," in *Data Mining and Knowledge Discovery Handbook*, Springer, Berlin, Heidelberg, pp. 321–352, 2005.

[89] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[90] K. Y. Yeung and W. L. Ruzzo, "Details of the adjusted rand index and clustering algorithms suppllement to the paper "an empirical study on principal component analysis for clustering gene expression data" (to appear in

bioinformatics)," *Science*, vol. 17, no. 9, pp. 763–774, 2001.

[91] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, vol. 78, no. 383, pp. 553–569, 1983.

[92] A. Fred and A. Lourenço, "Cluster ensemble methods: from single clusterings to combined solutions," in *Supervised and Unsupervised Ensemble Methods and their Applications*, Springer, Berlin, Heidelberg, vol. 126, pp. 3–30, 2008.

[93] J. de Leeuw, "Nonlinear multivariate analysis," *American Journal of Evaluation*, vol. 11, no. 2, pp. 155–157, 1990.

[94] M. Sato-ilk, "On evaluation of clustering using homogeneity analysis," in *IEEE Int. Conf. on Systems*, Nashville, TN, USA, vol. 5, pp. 3588–3593, 2000.

[95] M. Sato-Ilic, "On clustering based on homogeneity," in *IFSA World Congress and Nafips International Conference IEEE*, Vancouver, BC, Canada, vol. 5, pp. 2505–2510, 2001.

[96] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 410–420, 2007.

[97] E. Amigo´, J. Gonzalo, J. Artiles and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[98] J. Rissanen, "Modelling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[99] F. De Ridder, R. Pintelon, J. Schoukens and D. P. Gillikin, "Modified AIC and MDL model selection criteria for short data records," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 1, pp. 144–150, 2005.

[100] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "On clustering validation techniques–springer," *Journal of Intelligent Information Systems*, vol. 17, no. 2–3, pp. 107–145, 2001.