Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039923
Article

Fully Automated Density-Based Clustering Method

Bilal Bataineh* **and Ahmad A. Alzahrani**

Information Systems Department, College of Computers and Information Systems, Makkah, Saudi Arabia
*Corresponding Author: Bilal Bataineh. Email: bmbataineh@uqu.edu.sa
Received: 24 February 2023; Accepted: 01 June 2023; Published: 30 August 2023

Abstract: Cluster analysis is a crucial technique in unsupervised machine learning, pattern recognition, and data analysis. However, current clustering algorithms suffer from the need for manual determination of parameter values, low accuracy, and inconsistent performance concerning data size and structure. To address these challenges, a novel clustering algorithm called the fully automated density-based clustering method (FADBC) is proposed. The FADBC method consists of two stages: parameter selection and cluster extraction. In the first stage, a proposed method extracts optimal parameters for the dataset, including the epsilon size and a minimum number of points thresholds. These parameters are then used in a density-based technique to scan each point in the dataset and evaluate neighborhood densities to find clusters. The proposed method was evaluated on different benchmark datasets and metrics, and the experimental results demonstrate its competitive performance without requiring manual inputs. The results show that the FADBC method outperforms well-known clustering methods such as the agglomerative hierarchical method, k-means, spectral clustering, DBSCAN, FCDCSD, Gaussian mixtures, and density-based spatial clustering methods. It can handle any kind of data set well and perform excellently.

Keywords: Automated clustering; data mining; density-based clustering; unsupervised machine learning

1 Introduction

The abundance of data and information has led to an increased focus on identifying potential structures and unknown knowledge within the data. Clustering, a fundamental technique in unsupervised machine learning, is used to group unlabeled data into different clusters to highlight similarities among them [1–4]. That identify natural structures or patterns within a dataset, which can help in better understanding the data and making informed decisions. These clusters find applications in various fields such as finance, business, linguistics, healthcare, energy, community detection, medical image segmentation, security, big data analytics, and more [5–9]. Furthermore, clustering is widely employed in machine learning fields such as pattern recognition, data mining, analysis, and image processing [10–14].

Previously, various clustering algorithms have been developed based on different parameters, with each based on different parameters such as partitioning clustering, distribution model-based clustering, hierarchical clustering, and density-based clustering [15,16]. However, as indicated by the literature, the accuracy and efficacy of these methods vary based on the dataset's size and structure [17]. Moreover, most algorithms can only generate clusters in round shapes, which results in unsatisfactory clustering outcomes for datasets with many inconsistencies, outliers, and noise. Thus, finding an algorithm that can deliver reliable performance across different data types is challenging [18–20]. Additionally, these methods have a common shortcoming in that they require predetermined parameters such as the number of clusters, which need to be specified manually [3,21– 25].

This work introduces a novel density-based clustering method that adapts to the data and does not require any manual input to extract the optimal clusters. The proposed method can handle data sets of any size and structure and can produce clusters of any shape. It is also robust to outliers, noise, and varying densities. The method comprises two stages: Parameter selection and cluster extraction. In the proposed parameter selection, a statistical method is used to compute the optimal epsilon size (*Eps*) and the minimum number of points (*MinPts*) values based on the data set's size and structure. Here, *Eps* is the radius of a circular area around a specified point where the covered points are used to check the density. While *MinPts* is the threshold value of the number of points in the *Eps* area to be consider that the points belong to the same density. In the proposed clusters extraction stage, a run-based pass method is applied to label data points based on their ambient density. The proposed method's performance is evaluated using benchmark data sets and matrices and compared with other well-known clustering methods through visual and statistical evaluation experiments.

The paper is organized as follows; the next Section provides an overview of the state-of-the-art clustering algorithms, Section 3 presents the proposed density-based clustering method in detail, Section 4 discusses the experiments conducted and the results obtained to evaluate the proposed method, and finally, Section 5 provides the conclusion of the paper.

2 State of the Art

As per many literature reviews works on clustering, clustering algorithms can typically be classified into four categories: partitioning, hierarchical, model-based, and density-based clustering [15,16,26]. Each of these categories has been applied in various fields and has demonstrated good accuracy under certain data set circumstances.

The category of partitioning clustering includes several algorithms such as K-means [27], k-medoid [28], and mean shift [29] clustering. This approach involves dividing the data points into distinct clusters based on a particular partition function. Initially, the approach starts with an initial partition of the data and then uses an iterative control strategy to optimize the segmentation. Each cluster is usually represented by the center of gravity of the mass (as in k-means) or by one of the objects of mass near its center (as in k-medoid). However, the number of clusters must be specified manually, and processing initial values, noise, and outliers can be challenging.

The hierarchical approach of clustering is exemplified by the agglomerative clustering method [30], AGFC [31], and affinity Propagation [32]. This approach performs divisions or mergers based on the similarities and dissimilarities between the data points. The dataset is iteratively split and merged until a tree structure is generated, where the root represents the original dataset, and the branches represent the produced clusters [33,34]. This approach is easy to implement and can handle relational dataset points on a large scale. Agglomerative hierarchical clustering can reveal data patterns [1,4,9].

However, the number of clusters must be specified, and the output cannot be modified after splitting or merging operations. Additionally, the clustering results lack interpretation.

Model-based clustering assumes that data points are generated from a probabilistic model and tries to find the model that best matches the data [3,24]. This approach estimates the parameters of the model (like the number of clusters and probability distribution) from the data. The gaussian mixture model is the most used model for this type of clustering [35]. The advantage of Model-based clustering is that it does not require any prior knowledge of the number of clusters in the data. Moreover, the probabilistic nature of this method allows for uncertainty in clustering results, which can be valuable in outlier detection or density estimation applications [24]. However, the main challenge with this method is its computational complexity, which makes it difficult to use on large datasets. Furthermore, this method is not suitable for all types of datasets, such as large datasets, datasets with non-Gaussian distributions or non-linear structures, and datasets with high levels of noise or outliers.

Density-based clustering is a popular method of grouping data points based on the density of the points in space [26]. The approach involves identifying clusters of points where the density within each cluster is higher than the density outside the cluster. Examples of density-based clustering algorithms include DBSCAN [36], FCDCSD [7], and OPTICS [37]. Density-based clustering does not require prior knowledge of the number of clusters and can recognize clusters of varying shapes and sizes, making it effective in handling noise and outliers. However, the user needs to provide input values for the minimum number of points and the size of Epsilon.

In the state of the art, several clustering methods have been proposed based on prior approaches. Table 1 provides a summary of the up-to-date and well-known clustering methods, including their algorithmic approach, considering the strengths and weaknesses of each method.

Table 1: The summary of the state-of-the-art of clustering methods

Method	Adopted approach	Required parameters	Advantages	Disadvantage
FCDCSD [7]	Density-based	Epsilon size, Minimum points	Ability to handle non-spherical clusters, robustness to noise, and outliers.	Difficulty in choosing the appropriate parameters
K-means [27]	Partitioning	Number of clusters	Popular, fast, and simple	Requires specification of the number of clusters, handles spherical clusters only, Sensitive to the initial choice of centroids
K-medoid [28]	Partitioning	Number of clusters, distance metric	Robustness to noise and outliers, applicability to non-spherical clusters	Requires specification of the number of clusters, computationally expensive Sensitivity to the initial choice of medoids.

(Continued)

Table 1 (continued)

Method	Adopted approach	Required parameters	Advantages	Disadvantage
Mean-shift [29]	Partitioning	Bandwidth, Kernel function	Robustness to noise and outliers, applicability to non-spherical clusters	Sensitivity to bandwidth parameter
Agglomerative hierarchical [30]	Hierarchical	Linkage criterion, distance metric	Visualization of the hierarchy, flexible.	Computationally expensive, sensitivity to linkage and distance metric choice,
Affinity propagation [32]	Hierarchical/partitioning	Damping factor	Robustness to noise and outliers, applicability to non-spherical clusters.	Costly memory, difficulty in handling larger datasets, Sensitivity to a damping factor
Gaussian mixtures [35]	Model-based	Number of components, Initialization method, Covariance matrix type	Flexible, applicability to non-spherical clusters, and scale well with large datasets.	Difficulty in determining the number of components, sensitivity to initialization
DBSCAN [36]	Density-based	Epsilon size, minimum points	Ability to handle non-spherical clusters, Robustness to noise, and outliers.	Difficulty in choosing the appropriate parameters, sensitivity to the density of the data.
OPTICS [37]	Density-based	Epsilon size, minimum points, distance metric	Ability to handle non-spherical clusters, robustness to noise, and outliers. Produces a hierarchical clustering result	Difficulty in choosing the appropriate parameters, computationally expensive.
Spectral [38]	Partitioning	Number of clusters, spectral method,	Low sensitivity to initialization, applicability to non-spherical clusters.	Limited applicability to larger dataset ℭs Sensitive to noise

As demonstrated above, the available aggregation methods are diverse in terms of advantages and disadvantages. But they all share one basic drawback, which is that they require preset parameters. Clustering algorithms require many parameters, such as the number of clusters or distance scaling, epsilon size, minimum points, damping factor, etc. [3,5,9,22,27,39–42]. Choosing appropriate values for these parameters is a challenging task, especially when dealing with large datasets or high-dimensional datasets, and greatly affects the clustering result.

In the state of the art, different techniques can be used to select the optimal number of clusters or the initial centroids automatically [1,17,18,23,43]. That includes the rule of thumb [44], Elbow method [45], cross-validation [23], Gap statistic [21], information theory [46], and Silhouette analysis [23].

These techniques mainly select the appropriate parameters for K-means clustering automatically. But there is no one-size-fits-all method for all data and solutions, the choice of parameter selection method depends on the specific problem and dataset. In addition, they still depend on individual analysis and selection made.

The k-means method has received the most attention for addressing this problem because of its advantages. But density-based methods have also greater advantages and limited disadvantages over other approaches. They are simple, highly accurate, rapid, and widely used. However, the number of clusters of previous techniques is not required for the density-based clustering approach. Density-based methods need mainly epsilon size and the minimum number of points parameters.

A set of techniques can be used to select epsilon size and the minimum number of points parameters automatically spatially for DBSCAN such as Local reachability density (LRD) [9,37], k-distance plots [9,37], Silhouette [23], cross-validation [23], and Grid search [36]. In addition, some improvements of DBSCAN developed automated clustering such as DMDBSCAN [47]. But in general, the choice of parameter selection method depends on the specific problem and dataset, and it is necessary to evaluate the results and robustness of parameters. Other methods of density-based approach such as optics are ignored. The used techniques for DBSCAN cannot be used for other methods due to the different operations of finding clusters. In addition, these techniques are computationally expensive, can struggle with datasets of varying density and irregular shapes, and have difficulties in identifying meaningful values with high-dimensional datasets.

3 The Proposed Method

In this work, a fully automated density-based clustering method is proposed. As shown in Fig. 1, this method consists of two main stages: parameter selection and cluster extraction. These stages are presented in detail following:

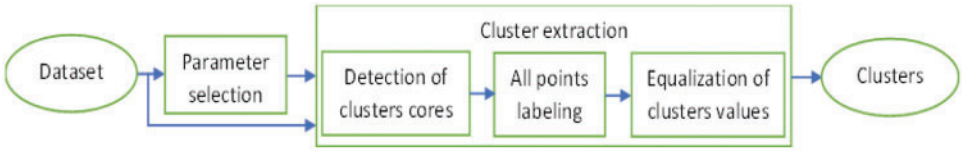


Figure 1: Framework of the proposed clustering method

3.1 Parameter Selection

In general, density-based clustering methods have two typical parameters, epsilon size (*Eps*) and the minimum number of points (Min_{Pts}). *Eps* is the radius of a circular area around a specified point where the covered points are used to check the density. While Min_{Pts} is the threshold value of the number of points in the *Eps* area to consider that the points belong to the same density. The optimal values for these parameters differ between methods based on the proposed operations. But in general, small *Eps* or large Min_{Pts} lead to cluster splitting, and large *Eps* or small Min_{Pts} lead to cluster merging.

3.1.1 Epsilon Size (Eps)

In this work, the optimal *Eps* for the respective dataset are calculated first. The required size of *Eps* is closely related to the characteristics of the data distribution and the resulting distances between

points. To have insight into that, the distance between each point and its nearest neighbor point is extracted based on Eq. (1):

$$D_{min}(P(x,y)) = min\left(\sqrt{(x-x_c)^2+(y-y_c)^2}\right) \tag{1}$$

where $Dmin(P(x,y))$ is the shortest distance between the respective point P and its nearest neighbor, x, and y are the coordinates of the point P, and xc and yc are the coordinates of an adjacent point in the same dataset. This step is applied continuously for each point in the dataset (Eq. (2)), and in end all minimum distance values are sorted in $List_D$ as shown in Eq. (3).

$$List_D = \forall(D_{min}(P)) : P \in Dataset \tag{2}$$

$$List_D = Sorted\,(List_D) \tag{3}$$

Next, a statistical histogram is applied to show the distribution of data of the extracted distances. Whereas notably there are close distance values in the data set space, it is not likely to be completely identical values, which confuses the statistical analysis. To get around this problem, the closest distances are equated into one fixed distance. A normalizing process is applied to the list to balance the close distances into equal-ranged intervals. All distances are rounded to R number of distances. Then the histogram of the normalized list is calculated by Eq. (4).

$$stg = histogram\left(int\left(\frac{List_D}{R}\right)\right) \tag{4}$$

$$R = int\,(Num_{Pnts} \times C) \tag{5}$$

where $Hstg$ is the output histogram, Num_{Pnts} is the number of sample points in the data set, R is the value of the subrange at which distances *will* become similar after normalization, and C is the division factor, after several test trials, the optimal value used in this work is $C = 0.01$. Histogram values are sorted in ascending order.

Figs. 2a–2c depict various distributed self-generated synthetic datasets that have three clusters and are of equal size, along with their corresponding histograms (Figs. 2d–2f). The y-axis represents the frequency of distances, and the x-axis represents the range of distances, illustrating the density properties among the data points within the dataset. As the range of distance values increases, there is a pattern of increased distribution, lower density, and even the possibility of noise between the points in the dataset. A high frequency with a symmetrical distribution around a single point indicates good centrality of the data around the cluster centers. On the other hand, higher frequencies around multiple points or with a skewed distribution indicate varying densities of the clusters.

The literature review suggests that finding the *Eps* value for well-distributed and structured datasets can be done easily through various techniques. However, the above discussion highlights the challenge of determining the optimal *Eps* value for datasets with varying densities. The interplay between the frequency and range of distance values leads to more intricate distributions within the dataset. Consequently, these two factors become crucial considerations in this context.

To address the challenges and compute the appropriate *Eps* value, the number of non-zero distances in the histogram after normalization is divided by the range of distances R_{Avg}. This division yields a value F that indicates the prevalence of noisy and long distances in the dataset, which will be considered when determining the optimal *Eps* later.

$$R_{Avg} = \frac{Num_{Pnts}}{R} \tag{6}$$

$$F = \frac{\#Hstg_{Values}}{R_{Avg}} \times .5 \tag{7}$$

where, $\#Hstg_{Values}$ represents the number of frequency points in the histogram range along the x-axis that are not equal to zero, while R_{Avg} is the average number of data points in each histogram range. The first minimum value (P_{min}) is the first instance where the frequency becomes zero after reaching its maximum value (P_{max}) and represents the larger side of the distance values. In finely distributed datasets, P_{min} encompasses all the symmetric values around the P_{max}. Asymmetric outliers refer to large distance values that are spread over a broad range. To account for these values, the P_{min} selected as the optimal *Eps* value is expanded based on the characteristics of the dataset. The range between the maximum (P_{max}) and minimum (P_{min}) frequency distances is calculated and adjusted to the noisy distribution characteristics of the dataset by multiplying it with F and then adding the result (as per Eq. (8)). This yields the optimal *Eps* size.

$$Eps = P_{min} + ((P_{min} - P_{max}) \times F) \tag{8}$$

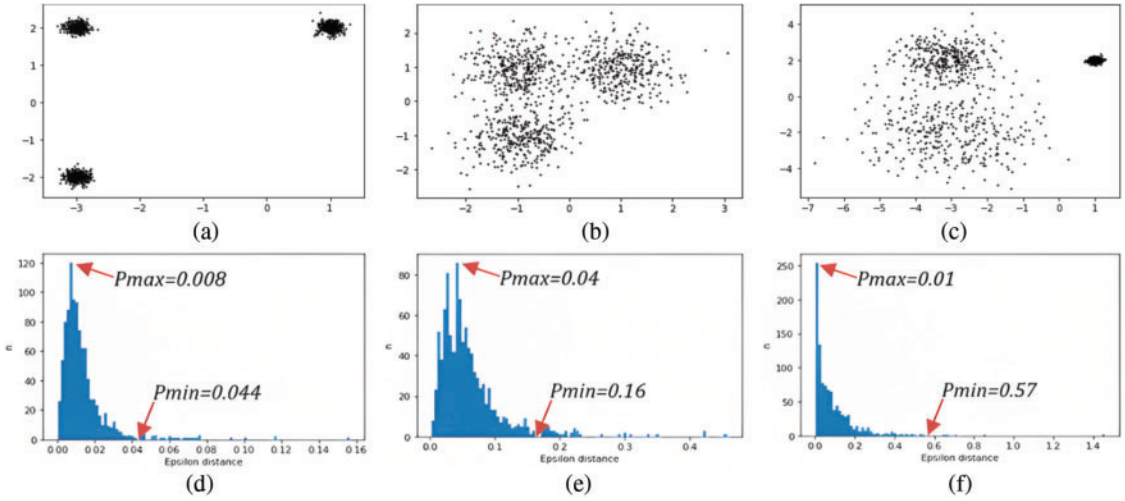


Figure 2: (a–c) are equal size datasets that have three clusters, (d–f) are the corresponding histograms of the datasets (a–c), respectively

3.1.2 Minimum Number of Points

The next step involves computing the second parameter, the minimum number of points (Min_{Pts}), based on the *Eps* value. To do this, all the points in the dataset are scanned using *Eps*, and calculate the number of points within *Eps* (Eq. (9)).

$$D_{Eps}(P(x,y)) = count\left(Eps <= \sqrt{(x-x_c)^2 + (y-y_c)^2}\right) \tag{9}$$

where $D_{Eps}(P(x, y))$ represent the number of distances between a given point P and the interconnected points within *Eps* where the distances are smaller than the *Eps* value. This process is repeated for each point in the dataset, and the results are stored in a list as depicted in Eq. (10).

$$List_{Eps} = \forall(D_{Eps}(P)) : P \in Data \tag{10}$$

After storing the results in $List_{Eps}$, the histogram of $List_{Eps}$ is computed, and the most frequent value is identified. In organized datasets, this value corresponds to the central point and is appropriate to be set as Min_{Pts} (Eq. (12)). However, in some datasets with highly diverse distributions, this value may split the highly distributed clusters. For such datasets, the statistical standard deviation of the number of points in *Eps* provides a more accurate measure to use as the Min_{Pts} value (Eq. (13)). However, in finely distributed or low-diversity datasets, using the standard deviation value may lead to the merging of small clusters. To overcome these issues, the value of Min_{Pts} is set as the maximum value between the two parameters in the dataset (Eq. (11)).

$$Min_{Pts} = MAX(A, B) \tag{11}$$

$$A = index\left(max\left(histogram\left(List_{Eps}\right)\right)\right) \tag{12}$$

$$B = std\left(List_{Eps}\right) \tag{13}$$

here, A represents the value with the highest frequency in the $List_{Eps}$ histogram, and B represents the standard deviation of $List_{Eps}$.

3.2 Clusters Extraction

In the second stage, the sample points in the dataset are assigned cluster labels. Further details are presented below.

3.2.1 Detection of Clusters Cores

This step considers each data point in the dataset, in order from either the smallest to the largest or the largest to the smallest. Each point is chosen as a central point for a circular area with a radius of *Eps*. The step then counts the number of points that are located within this circular area from the central point *P*.

$$NumPnts = count\left(Eps <= \left(\sqrt{(x - x_c)^2 + (y - y_c)^2}\right)\right) \tag{14}$$

If this quantity Num_{Pnts} is greater than or equal to the Min_{Pnts} parameter, then all points are considered to belong to the same cluster. All these neighbor points are assigned an initial cluster value using one of two labeling operations.

If none of the neighboring data points have been labeled in previous steps of the clustering algorithm, then all neighboring points are assigned the next value of the largest label $Label_{Max}$ that has been used previously. This ensures that each cluster is assigned a unique label and avoids any potential conflicts with existing labels.

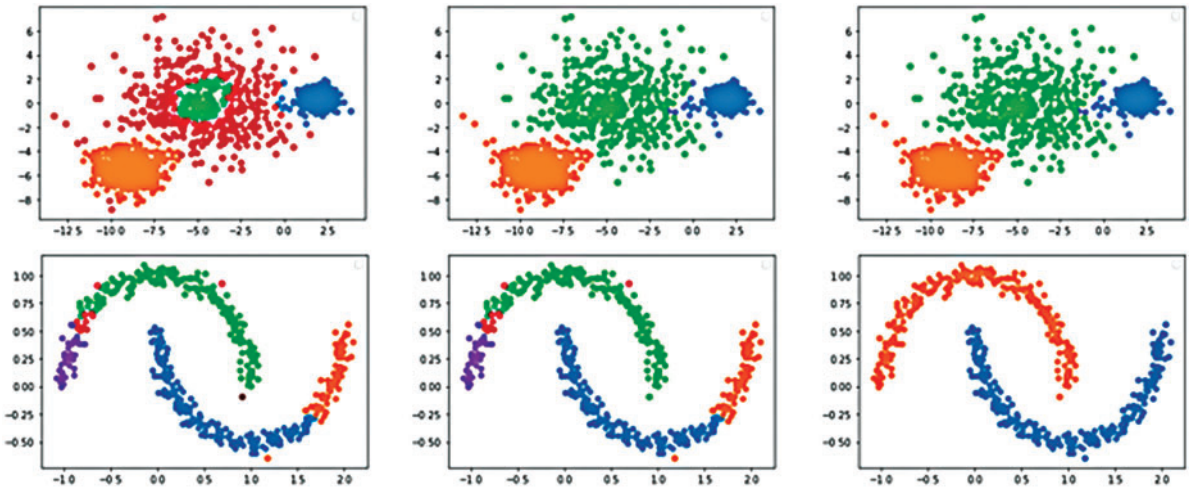
$$Label[P] = Label_{Max} + 1 \tag{15}$$

If at least one of the neighboring data points has already been assigned a label in previous steps of the detection cores of clusters, all neighboring points will be labeled with the smallest label value among the labeled neighbors ($min(Label)$). Additionally, the label values of points in the *Eps* region are recorded and appended as associated labels for all neighboring points (Eq. (17)). This information will be used in the later steps of the algorithm.

$$Label[P] = min(Label) \tag{16}$$

$$Label_{Asso}[P] = Label_{Asso}[P] \cup \{Label[P]\} \tag{17}$$

If the number of points located within the *Eps* area is less than the value of Min_{Pnts}, then the density of this region is considered too low to form a cluster. This process is repeated for all points in the dataset, identifying high-density regions as potential cluster cores. Any points that have not been labeled at this stage will be considered in the next step. The results of this process are shown in Fig. 3a.



(a) (b) (c)

Figure 3: (a) The output of the detection of clusters cores step, (b) the output of all points labeling step, and (c) the output of equalization of clusters values step

3.2.2 All Points Labeling

In this step, all non-clustered points in the dataset are processed. All neighboring points within the circular *Eps* region around each non-labeled point P are considered. If some of these neighboring points have already been assigned a label ($P_{Labeled}$) in the previous step, then the non-clustered points are assigned the label of the closest labeled point to point P based on their distance (Eq. (18)). This process is applied iteratively until no further changes occur, and the algorithm converges to the final set of clusters. The result of this step is presented in Fig. 3b. However, in some highly distributed datasets, some outlier points may still not be assigned to a potential cluster. To address this issue, these outlier points can be connected to their nearest clustering point.

$$Eps <= \left(\sqrt{(x - x_c)^2 + (y - y_c)^2}\right)?Label[P] = Label[P_{Labeled}] \tag{18}$$

3.2.3 Equalization of Clusters Values

Now we have good clustering for many datasets, but clustering values still need to be resolved for datasets with complex shapes and low density. In these cases, some connected clusters need to be merged. To merge connected clusters with complex shapes, the associated label values in $Label_{Asso}$ are replaced by a single value, which then replaces all equivalent values in clusters. Sets of associated labels in $Label_{Asso}$ that share the same elements are considered connected and represented by a single value using the Union-Find method. This method integrates groups with shared values into one group and returns the smallest value in the unionized sets through a search operation. Different Union-Find data structure methods can be used for this purpose. The output of this step is the final clustering result of

the proposed algorithm, which is shown in Fig. 3c. The complete algorithm of the proposed method is presented in Algorithm 1.

Algorithm 1: The pseudocode of the proposed clustering method

Input: Unlabeled dataset ($\boldsymbol{D}$)
Outputs: Clustered dataset ($\boldsymbol{D}_{\_Cluster}$)

```
        # 1-Parameter selections
1:      for each P in D: # 1-Epsilon size (Eps)
2:          List_D ← D_min (P)
3:      Eps = P_min + ((P_min − P_max) × F) # Details in Eqs. (3)–(8)
4:      for each P in D:# 2-Minimum number of points
5:          List_pnts = count points in Eps
6:      Min_Pts = MAX(A, B) # Details in Eqs. (11)–(13)
        #2-Cluster extraction
        #2.1-Detection of clusters cores
7:      for each P in D:
8:          Num_Pnts = count points in Eps
9:        if Num_Pnts > Min_pts
10:             if P in Eps is labeled
11:               Label[P] = min(Label)
12:               Label_Asso [P] = Label_Asso [P] ∪ Label[P]
13:             else Label[P] = New label value
        #2.2-All points labeling
14:     while P in Eps is unlabeled AND at least a point in Eps is labeled:
15:             Label[P] = Label [P_Labeled]
16:     for each unlabeled P:
17:         Label[P] = Closer Label [P]
      # 2.3-Equalization of clusters values
18:   D_ Cluster = Union-Find (Label_Asso)
```

4 Experiments

To evaluate the effectiveness of the proposed method, several forms of experiments are performed as detailed below.

4.1 Datasets

A set of experiments are performed using synthetic databases to illustrate the performance. The used databases are noisy circles, noisy moons, blobs, anisotropic distributed data, no structure, and blobs with varying variances. Using these synthetic databases has been adopted in previously published research [1,19,23,36,48]. To ensure the effectiveness of the proposed method with different structures of data sets, different numbers of samples and densities are used. Some experiments used reconstructed datasets consisting of 1,500 and 500 data samples. In addition, dataset samples contain ground truth labels that are used to evaluate the results of the clustering method. Fig. 4 shows examples of the datasets used.

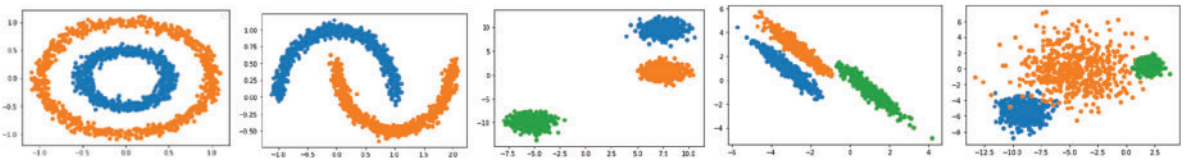


Figure 4: The noisy circles, noisy moons, blobs, anisotropic, and varying variance datasets

4.2 Performance Evaluation

For performance analyses, sets of grouping metrics are used.

(1) *The Rand index (RI), RI* measure is the main measure used to assess the similarity between two labeled datasets. It finds the same or different combinations by ignoring permutations for true precision. *RI* is equivalent to the precision defined as:

$$RI = \frac{a+b}{a+b+c+d} \tag{19}$$

where, a is the number of identical point samples found in the same group in A and in the same group in B, b is the number of identical point samples found in different groups in A and different groups in B, c is the number of identical point samples found in the same set in A and in different sets in B, and d is the number of identical point samples found in different groups in A and in the same group in B. Where A is the labeled ground truth and B is the aggregation results.

(2) *Homogeneity* estimates the quantity of a particular class present in a cluster.

$$Homogeneity = 1 - \frac{H(C|K)}{H(C)} \tag{20}$$

$$H(C|K) = -\sum_{c,k} \frac{n_{c,k}}{N} log(\frac{n_{c,k}}{n_k}) \tag{21}$$

where $H\,(C \mid K)$ is the ratio between the number of points labeled as c and the total number of points in cluster k.

(3) *Completeness* evaluates the number of similar samples that are allocated together to the same cluster.

$$Completeness = 1 - \frac{H(K|C)}{H(K)} \tag{22}$$

(4) The *harmonic* (*V_measure*) is the mean between *Homogeneity* and *Completeness* delivered by Normalized Mutual Information (*NMI*).

$$NMI = 2 * \frac{H * C}{H + C} \tag{23}$$

where H represents *homogeneity*, and C represents *completeness*.

4.3 Analysis of the Automated Parameter Performance

In this Section, the effectiveness of an automated method for calculating clustering parameters is assessed. The proposed approach is compared to two other common methods for automated clustering: A dynamic method for discovering density-varied clusters (DMDBSCAN) [47] and the elbow method used by K-means to estimate the best number of clusters. These methods were selected because they are widely used for automated clustering. Table 2 illustrates that the proposed method

outperformed the other methods in terms of accuracy. DMDBSCAN performed well on datasets with a fine distribution such as noisy circles, noisy moons, and blobs, achieving an accuracy rate of almost 100%. However, it struggled with datasets that had diverse densities and distributions, such as the Aniso and Diverse datasets. In addition, the accuracy of DMDBSCAN was negatively affected by the number of samples in the Aniso dataset, with a drop from 77% to 33% as the number of samples decreased from 1500 to 500.

Table 2: The *RI* of DMDBSCAN, K-means+ elbow, and the proposed clustering methods on noisy circles, noisy moons, blobs, anisotropic, and varying variance, with 1500 and 500 samples datasets

	# Samples	Noisy circles	Noisy moons	Blobs	Aniso	Varied
DMDBSCAN	1500	100%	99%	99%	77%	33%
K-means+ elbow	1500	50%	62%	100%	8%	92%
Proposed FADEC	1500	100%	100%	100%	99%	97%
DMDBSCAN	500	100%	99%	99%	33%	33%
K-means+ elbow	500	49%	62%	78%	3%	90%
Proposed FADEC	500	100%	100%	100%	100%	93%

K-means with elbow achieved good results with datasets that had spherical clusters, such as the bubble and varied datasets, with accuracy rates of 100%, 78%, 92%, and 90%. However, its performance was low with complex structural datasets, achieving only 8% and 3% accuracy rates with the Aniso dataset. Furthermore, the K-means method with elbow was affected by the number of samples in the dataset, with a decrease in accuracy from 100% to 78% when the number of samples decreased from 1500 to 500, as observed in the Blobs dataset. In contrast, the proposed method achieved high and stable accuracy rates across all cases. It was highly adaptable to all datasets, regardless of their structure and features, with high performance in most cases (100% accuracy), and a worst-case scenario of 93% accuracy.

4.4 Analysis of the Effectiveness of the Clustering Process

The clustering process's overall performance, whether the parameter values are set manually or automatically, is evaluated by comparing the output of the proposed method with a set of benchmark clustering methods. The most popular and recent methods are selected from each category, including K-means for Partitioning clustering, Agglomerative Hierarchical for Hierarchical clustering, Gaussian Mixtures for Model-based clustering, and DBSCAN and FCDCSD for Density-based clustering. These methods are applied to Noisy circles, Noisy moons, Blobs, Aniso, and Varied datasets, each containing 1500 samples. Fig. 5 is presented to provide a clear visual representation of the results of each method. The results indicate that the performance of each method varies with the dataset's features. Some methods perform poorly in most cases, such as hierarchical clustering methods, Gaussian mixtures, and K-means methods. In contrast, other algorithms, such as the proposed FADBC, FCDCSD, and DBSCAN, provide more reliable performance regardless of the type of dataset.

To give accurate and analytical results, Tables 3–7 show a statistical analysis of performance based on measures of the *Rand Index* (*RI*), *Homogeneity*, *Completeness*, and *harmonic* (*V_measure*). Each method is applied to every data set approved and evaluated. Overall, the proposed FADBC, FCDCSD, and DBSCAN respectively had the best performance. While the proposed method achieved

high performance with all measurements and datasets used ranging from 100% to 92% accuracy rates (98.6% on average), FCDCSD followed with a performance of 100% to 91% accuracy rates (97.3% on average). The rest of the methods (DBSCAN, Gaussian mix, agglomerative hierarchical, and k-means) achieved average performance of 86.3%, 71.3%, 66.6%, and 58.7%, respectively. Fig. 6 summarizes the average performance of the incorporating methods for each measure with all datasets.

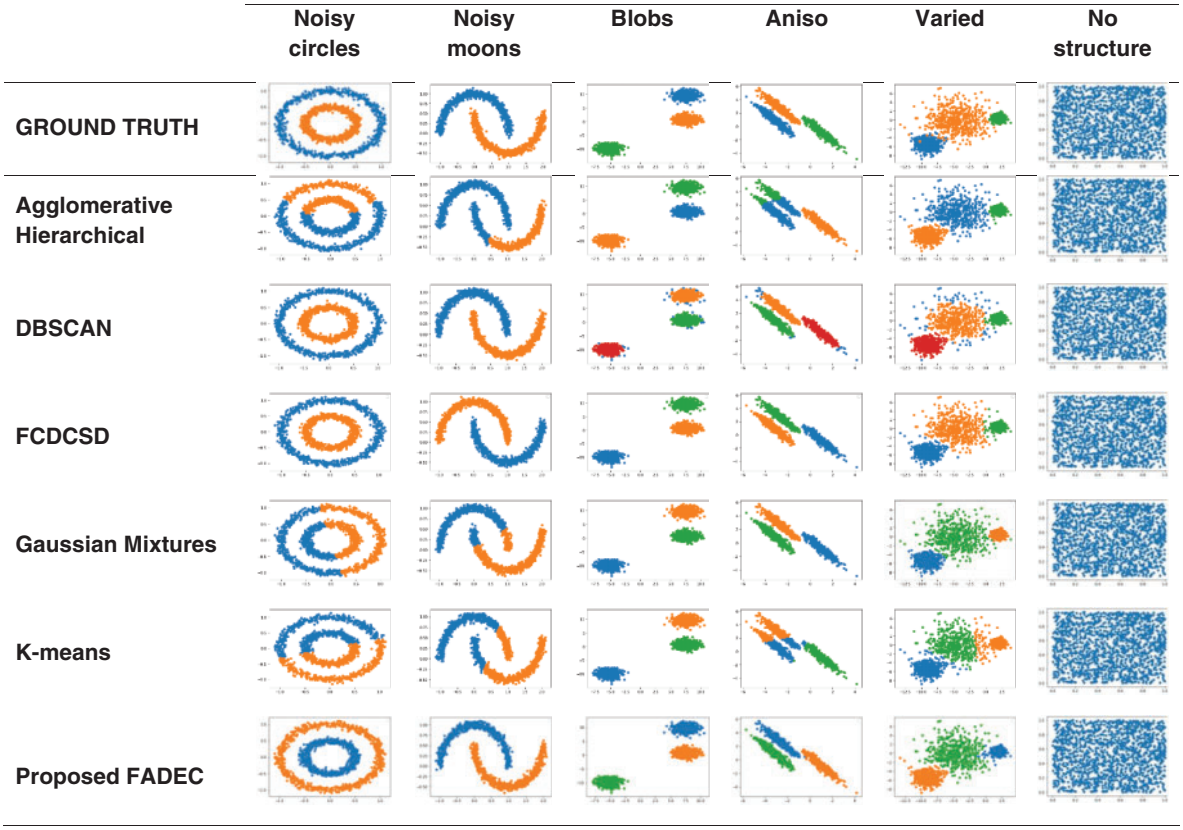


Figure 5: The visual result of agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means, and the Proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, and varying variance datasets

Table 3: The *RI* of agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means, and the proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, varying variance, and no structure datasets

	Noisy circles	Noisy moons	Blobs	Aniso	Varied	No structure
Agglo. hierarchical	50%	73%	100%	80%	99%	100%
DBSCAN	100%	100%	98%	99%	96%	100%
FCDCSD	100%	100%	98%	100%	97%	100%
Gaussian mixtures	50%	75%	100%	100%	99%	100%
K-means	50%	62%	100%	82%	92%	100%
Proposed FADBC	100%	100%	100%	99%	97%	100%

Table 4: The *Homogeneity* of agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means, and the Proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, varying variance, and no structure datasets

	Noisy circles	Noisy moons	Blobs	Aniso	Varied	No structure
Agglo. hierarchical	0%	48%	100%	63%	95%	100%
DBSCAN	100%	100%	98%	99%	94%	100%
FCDCSD	100%	100%	100%	100%	91%	100%
Gaussian mixtures	0%	40%	100%	100%	94%	100%
K-means	0%	18%	100%	63%	81%	100%
Proposed FADBC	100%	100%	100%	100%	92%	100%

Table 5: The *Completeness* of agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means, and the Proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, varying variance, and no structure datasets

	Noisy circles	Noisy moons	Blobs	Aniso	Varied	No structure
Agglo. hierarchical	0%	51%	100%	68%	95%	100%
DBSCAN	0%	18%	100%	63%	82%	100%
FCDCSD	100%	100%	91%	93%	85%	100%
Gaussian mixtures	0%	40%	100%	100%	94%	100%
K-means	0%	19%	100%	63%	82%	100%
Proposed FADBC	100%	100%	100%	100%	92%	100%

Table 6: The *harmonic* (*V_measure*) of agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means, and the Proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, varying variance, and no structure datasets

	Noisy circles	Noisy moons	Blobs	Aniso	Varied	No structure
Agglo. hierarchical	0%	49%	100%	65%	95%	100%
DBSCAN	100%	100%	94%	96%	89%	100%
FCDCSD	100%	100%	100%	100%	91%	100%
Gaussian mixtures	0%	40%	100%	100%	94%	100%
K-means	0%	18%	100%	63%	81%	100%
Proposed FADBC	100%	100%	100%	100%	92%	100%

Table 7: The average of all measurements of agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means, and the Proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, varying variance, and no structure datasets

	Rand index	Homogeneity	Completeness	Harmonic	Average
Agglo. hierarchical	61.8%	62.8%	61.2%	80.4%	66.6%
DBSCAN	95.8%	52.6%	98.2%	98.6%	86.3%

(Continued)

Table 7 (continued)

	Rand index	Homogeneity	Completeness	Harmonic	Average
FCDCSD	98.2%	93.8%	98.2%	99.0%	97.3%
Gaussian mixtures	66.8%	66.8%	66.8%	84.8%	71.3%
K-means	52.4%	52.8%	52.4%	77.2%	58.7%
Proposed FADBC	98.4%	98.4%	98.4%	99.2%	98.6%

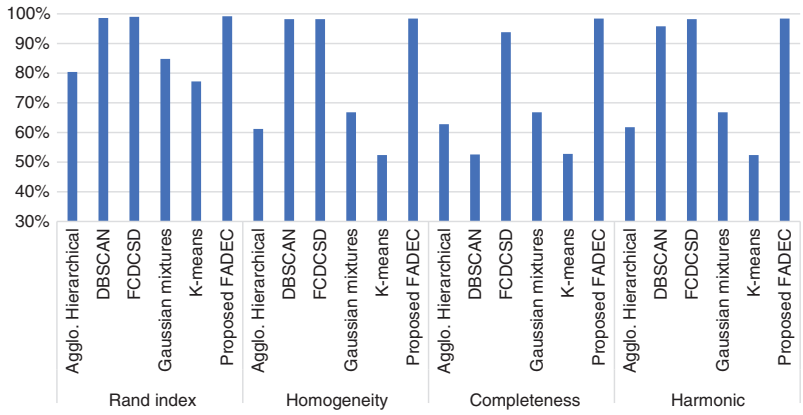


Figure 6: The average values of *Rand Index* (*RI*), *Homogeneity*, *Completeness*, and *harmonic* (*V_measure*) measurements with agglomerative hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, k-means and the proposed FADBC clustering methods on noisy circles, noisy moons, blobs, anisotropic, and varying variance datasets

The summary of previous experiments indicates that the proposed FADBC approach achieved an average accuracy rate of 98.6% across all the datasets tested, surpassing the accuracy rates achieved by Agglomerative Hierarchical, Hierarchical, DBSCAN, FCDCSD, Gaussian mixtures, and K-means, which scored 66.6%, 86.3%, 97.3%, 71.3%, and 58.7%, respectively.

4.5 The Time Complexity

Table 8 presents the time complexity of the involved clustering methods, including the proposed FADBC. The results show that the proposed FADBC has a comparable time complexity to well-known methods. Additionally, it is the only fully free parameter predetermined method among the methods listed in the table, which is a significant advantage.

Table 8: The time complexity (Big O) for each clustering algorithm

Method	Complexity time
Agglomerative hierarchical	$O(N^3)$
DBSCAN	O(N log N)
FCDCSD	O(N)

(Continued)

Table 8 (continued)

Method	Complexity time
Gaussian mixtures	$O(N^3)$
K-means clustering	O(INK)
Mean-shift	$O(IN^2)$
OPTICS	O(N log N)
Spectral clustering	$O(N^3)$
Ward hierarchical clustering	$O(N^3)$
Proposed FADBC	O(IN)

N = number of points, I = number of iterations, K = total number of partitions

5 Discussion

In case the parameters are selected automatically, the results in Table 2 show that the proposed FADBC can automatically select the optimal values for the *Eps* and Min_{Pts} parameters to effectively deal with different types of datasets. Among the automatic parameter clustering methods, K-means + elbow fails with datasets with complex structures such as noisy circles and anisotropic distributed datasets.

In addition, K-means + elbow accuracy varies across several samples in the dataset as presented in the Blobs dataset. On the other hand, DMDBSCAN shows perfect performance with well-structured datasets such as Noisy circles, Noisy moons, and Blobs datasets. However, its mechanism failed to extract an optimal *Eps* value with unstructured datasets as shown with diverse and anisotropic distributed datasets. The proposed method finds optimal values and achieves high performance with any data distribution or sample size. It covers a variety of data types and requires less human intervention.

In the overall performance aspect of the clustering process regardless of whether the parameter values are set manually or automatically, the proposed FADBC, then FCDCSD, and DBSCAN achieved the highest and most stable performance, respectively as presented in Tables 3–7 and summarized in Fig. 4. It shows excellent performance on any dataset with all metrics used. While each of the methods involved had some drawbacks with some cases of datasets. Moreover, the proposed FADBC has a fast execution time. Based on the results in Table 8, its time complexity is O(IN), which is similar to a set of well-known clustering methods.

The FADBC proposal primarily involves extracting the core clusters that are available, associating samples with their most suitable cores, and combining the clusters that fall short of threshold boundaries. This is done by determining the optimal parameters for each dataset automatically, thereby making them adaptable to different data sets. Unlike other methods that rely on manually determined parameters set by the user, which may not be optimal parameter values. The FADBC can deal with complex shapes of clusters and adjust noise points in datasets compared to most clustering methods. In addition, it can handle outliers and variance-dense datasets, while many such as K-means and Agglomerative Hierarchical and Gaussian mixtures failed with datasets such as Noisy circles, Noisy moon, and Aniso datasets. The proposal FADBC can produce clusters in complex shapes, while most current methods can only form clusters in elliptical or circular patterns.

Finally, by adding the high performance of the automatic method to determine the parameters to the previous advantages of the proposed method, it is hypothesized that the proposal FADBC is better

than the included involved benchmark clustering methods. For future work, clustering of overlapping datasets could be considered. It should be noted that both the proposed method and all the methods involved have failed when dealing with overlapping datasets.

6 Conclusions

This study aims to present a novel density-based clustering approach that can overcome the limitations of current clustering methods. The proposed method, called FADBC, does not require any input parameters and exhibits high performance in addressing current challenges in the clustering field. FADBC consists of two stages: parameter selection and cluster extraction. In the parameter selection stage, a statistical approach is used to determine the optimal values for *Eps* and Min_{Pts} based on the size and structure of the dataset. In the cluster extraction stage, a proposed method is applied to label data points based on their ambient density. A series of experiments was conducted to assess the performance of the proposed method. The results show that the proposed FADBC outperforms other benchmark clustering methods and performs well on any dataset based on all metrics used. The method can handle complex cluster shapes, adjust noise points, and deal with outliers and variance-dense datasets. By incorporating the automatic parameter selection method with the advantages, the proposed FADBC method is superior to other benchmark clustering methods.

Acknowledgement: Authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by grant number 23UQU4361009DSR001. I also would like to thank anonymous reviewers for their constructive feedback on the initial manuscript.

Funding Statement: This work is funded by the Deanship of Scientific Research at Umm Al-Qura University, Grant Code: (23UQU4361009DSR001).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Bilal Bataineh; data collection: Bilal Bataineh1 and Ahmad A. Alzahrani; analysis and interpretation of results: Bilal Bataineh1 and Ahmad A. Alzahrani; draft manuscript preparation: Bilal Bataineh1 and Ahmad A. Alzahrani. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data used in this study are freely available and accessible. The sources of the data utilized in this research are thoroughly explained in the main manuscript.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. M. Najim Adeen, A. M. Abdulazeez and D. Q. Zeebaree, "Systematic review of unsupervised genomic clustering algorithms techniques for high dimensional datasets," *Technology Reports of Kansai University*, vol. 62, no. 3, pp. 355–374, 2020.

[2] D. Mustafi, A. Mustafi and G. Sahoo, "A novel approach to text clustering using genetic algorithm based on the nearest neighbour heuristic," *International Journal of Computers and Applications*, vol. 44, no. 3, pp. 291–303, 2022.

[3] M. Kashyap, S. Gogoi, R. K. Prasad and C. Science, "A comparative study on partition-based clustering methods," *International Journal of Computer Applications*, vol. 6, no. 2, pp. 1457–1463, 2018.

[4] J. Shuja, M. Humayun, W. Alasmary, H. Sinky, E. Alanazi *et al.,* "Resource efficient geo-textual hierarchical clustering framework for social IoT applications," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25114–25122, 2021.
[5] C. Zhang, W. Huang, T. Niu, Z. Liu, G. Li *et al.,* "Review of clustering technology and its application in coordinating vehicle subsystems," *Automotive Innovation*, vol. 6, pp. 1–27, 2023.
[6] M. H. Wang, Y. F. Tseng, H. C. Chen and K. H. Chao, "Face clustering method based on nearest neighborhood aggregation," *Computer and Modernization*, vol. 1, no. 12, pp. 83–92, 2022.
[7] B. Bataineh, "Fast component density clustering in spatial databases: A novel algorithm," *Information*, vol. 13, no. 10, pp. 477, 2022.
[8] M. Pandey, O. Avhad, A. Khedekar, A. Lamkhade and M. Vharkate, "Social media community using optimized clustering algorithm," in *ICT Analysis and Applications*, Goa, India, vol. 2022, pp. 669–675, 2022.
[9] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 5–7, 2021.
[10] H. M. Alghamdi and A. Selamat, "Arabic web page clustering: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 1, pp. 1–14, 2019.
[11] Y. Djenouri, A. Belhadi, D. Djenouri and J. C. W. Lin, "Cluster-based information retrieval using pattern mining," *Applied Intelligence*, vol. 51, no. 4, pp. 1888–1903, 2021.
[12] D. Guo, B. Qi and C. Wang, "Fast clustering method of LiDAR point clouds from coarse-to-fine," *Infrared Physics & Technology*, vol. 129, no. 10, pp. 104544, 2023.
[13] H. M. Alghamdi, A. Selamat and N. S. A. Karim, "Improved text clustering using k-mean bayesian vectoriser," *Journal of Information & Knowledge Management*, vol. 13, no. 3, pp. 1–10, 2014.
[14] J. Kumar, M. Sravani, M. Akhil, P. Sureshkumar and V. Yasaswi, "Crime rate prediction based on k-means clustering and decision tree algorithm," in *Computer Networks and Inventive Communication Technologies*, Coimbatore, India, pp. 451–462, 2022.
[15] F. Alalyan, N. Zamzami and N. Bouguila, "Model-based hierarchical clustering for categorical data," in *2019 IEEE 28th Int. Symp. on Industrial Electronics (ISIE)*, Vancouver, Canada, pp. 1424–1429, 2019.
[16] M. Aljibawi, M. Z. A. Nazri and N. O. R. S. Sani, "An enhanced mudi-stream algorithm for clustering data stream," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 9, pp. 3012–3021, 2022.
[17] M. Ahmed, R. Seraj and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, pp. 1–12, 2020.
[18] M. H. Wang, Y. F. Tseng, H. C. Chen and K. H. Chao, "A novel clustering algorithm based on the extension theory and genetic algorithm," *Expert Systems With Applications*, vol. 36, no. 4, pp. 8269–8276, 2009.
[19] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei *et al.,* "One-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 2022–2034, 2019.
[20] A. Zelig and N. Kaplan, "KMD clustering: Robust generic clustering of biological data," *bioRxiv*, pp. 1–25, 2020.
[21] L. Wang, C. Leckie, K. Ramamohanarao and J. Bezdek, "Automatically determining the number of clusters in unlabeled data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 3, pp. 335–350, 2009.
[22] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition—Part II," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 6, pp. 786–801, 1999.
[23] L. H. Dos Santos Fernandes, A. C. Lorena and K. Smith-Miles, "Towards understanding clustering problems and algorithms: An instance space analysis," *Algorithms*, vol. 14, no. 3, pp. 1–29, 2021.
[24] A. U. Rehman and S. B. Belhaouari, "Divide well to merge better: A novel clustering algorithm," *Pattern Recognition*, vol. 122, no. 6, pp. 1–18, 2022.
[25] H. Wang, Y. Yang, B. Liu and H. Fujita, "A study of graph-based system for multi-view clustering," *Knowledge-Based Systems*, vol. 163, no. 2, pp. 1009–1019, 2019.
[26] Y. Djenouri, A. Belhadi and J. C. W. Lin, "Recurrent neural network with density-based clustering for group pattern detection in energy systems," *Sustainable Energy Technologies and Assessments*, vol. 52, no. 16, pp. 1–8, 2022.

[27] C. El Morr, M. Jammal, H. Ali-Hassan and W. El-Hallak, "K-Means," in *Machine Learning for Practical Decision Making: A Multidisciplinary Perspective with Applications from Healthcare, Engineering and Business Analytics*. Switzerland: Springer, pp. 361–384, 2022.
[28] V. Makarenkov, G. S. Barseghyan and N. Tahiri, "Inferring multiple consensus trees and supertrees using clustering: A review," *arXiv Prepr. arXiv*2301.00483, 2023.
[29] H. Zhang, "Battlefield situation aggregation display technology based on meanshift," in *Third Int. Conf. on Computer Vision and Data Mining (ICCVDM 2022)*, vol. 12511, pp. 483–489, 2023. https://doi.org/10.1117/12.2660264
[30] 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.
[31] M. Cheng, T. Ma, L. Ma, J. Yuan and Q. Yan, "Adaptive grid-based forest-like clustering algorithm," *Neurocomputing*, vol. 481, no. 5, pp. 168–181, 2022.
[32] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
[33] F. Nielsen, "Hierarchical clustering," in *Introduction to HPC with MPI for Data Science*, pp. 195–211, 2016.
[34] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
[35] M. S. Yang, C. Y. Lai and C. Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognition*, vol. 45, no. 11, pp. 3950–3961, 2012.
[36] J. Cai, H. Wei, H. Yang and X. Zhao, "A novel clustering algorithm based on DPC and PSO," *IEEE Access*, vol. 8, pp. 88200–88214, 2020.
[37] M. Ankerst, M. M. Breunig, H. P. Kriegel and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM Sigmod Record*, vol. 28, no. 2, pp. 49–60, 1999.
[38] A. Ng, M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *14th Int. Conf. on Neural Information Processing Systems: Natural and Synthetic*, Vancouver British Columbia, Canada, pp. 849–856, 2001.
[39] A. A. Munshi, "Clustering of wind power patterns based on partitional and swarm algorithms," *IEEE Access*, vol. 8, pp. 111913–111930, 2020.
[40] B. Bataineh, "A fast and memory-efficient two-pass connected-component labeling algorithm for binary images," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 2, pp. 1243–1259, 2019.
[41] S. Misbahuddin, A. R. Al-Ahdal and M. A. Malik, "Low-cost MPI cluster based distributed in-ward patients monitoring system," in *2018 IEEE/ACS 15th Int. Conf. on Computer Systems and Applications*, Aqaba, Jordan, pp. 1–6, 2018.
[42] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.
[43] P. Agarwal, M. A. Alam and R. Biswas, "Issues, challenges and tools of clustering algorithms," *International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 523–528, 2011.
[44] F. M. L. Di Lascio, D. Giammusso and G. Puccetti, "A clustering approach and a rule of thumb for risk aggregation," *Journal of Banking and Finance*, vol. 96, no. 4, pp. 236–248, 2018.
[45] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman and B. D. Satoto, "Integration K-means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conference Series: Materials Science and Engineering*, vol. 336, pp. 1–6, 2018.
[46] S. F. Tan and N. A. M. Isa, "Exposure based multi-histogram equalization contrast enhancement for non-uniform illumination images," *IEEE Access*, vol. 7, pp. 70842–70861, 2019.
[47] M. T. H. Elbatta and W. M. Ashour, "A dynamic method for discovering density varied clusters," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 1, pp. 123–134, 2013.
[48] X. Xu, S. Ding, Y. Wang, L. Wang and W. Jia, "A fast density peaks clustering algorithm with sparse search," *Information Science*, vol. 554, no. 3, pp. 61–83, 2021.