Information Clustering Algorithms Review and Analysis

Farah Abbac Sari

Kufa University /Department of Computer Science, Faculty of Computer Science & Mathematics, Najaf, 54001, Iraq

E-mail: faraha.altaee@uokufa.edu.iq

Ali Abdulkarem Habib Alrammahi

Kufa University /Department of Computer Science, Faculty of Computer Science & Mathematics, Najaf, 54001, Iraq

E-mail: alia.alramahi@uokufa.edu.iq

Received: 14th December, 2022; Accepted: 7th February, 2023; 28th February, 2023

Abstract: Clustering algorithms are a powerful tool for analysing and classifying large amounts of data by dividing this information into clusters, so as to group the objects into one cluster when they are similar on certain metrics. To solve this problem, a large no of methods and algorithms have been investigated. Due to the diversity of the way these algorithms work and the variables required for them, remains an urgent problem of selecting specific algorithms that provide accurate results and less consumsion of time for processing large data. The paper presents an attempt to classify the existing methods and algorithms, as well as analyze their applications for processing big data, so that we can choose the appropriate algorithm for the hashing process, based on a comparison of its performance indicators.

Index Terms: Clustering, Big data, Classification, Criterion.

1. INTRODUCTION

Because significant changes in the data storage process are required when compared initial clustering technique, the volume of data is the first and one of the most important characteristics that will affect the data clustering process. Another important feature is speed, as this process is frequently used in online data processing. The third feature is diversity; because the data is created and processed from various sources such as social media, tik tok, and so on, it may belong to a variety of data types (text, image, video).

Despite the large number of reviews of clustering algorithms, [1]It is challenging for users to determine which algorithm is well suited for a specific big data set. The reason for this is that the characteristics of the algorithms are not well understood, and new algorithms have appeared in this area that are not considered in the list presented; and there are no results of empirical analysis that speaks about the advantages of one algorithm over another.

2. CLASSIFICATION OF CLUSTERING METHODS

Given the large number of clustering algorithms, let us present a categorization structure that groups various clustering algorithms. The proposed categorization structure is intended to focus on the clustering method's applied details. Consider the following clustering method classifications [2],[3].

1) Based on the division: All clusters are determined instantly in such algorithms. The initial distribution of objects among clusters is predetermined, and they are redistributed at each iteration. The following requirements must be met by clusters: There must be at least one object in each cluster, and each object can only belong to one cluster. The K-means algorithm method works by averaging coordinates of all objects included in the centre of the cluster. Further, k-medoids clustering algorithm, as compared to the K-means method, at each iteration evaluate the centers of clusters as medoids of points, that is, the mid point of the cluster is necessarily one of its points. Rest of clustering algorithms such as K-modes, CLARANS PAM, FCM, and CLARA.

2) Hierarchical sorting: Data is sorted hierarchically by proximity. A dendrogram is a data set representation in which the individual data are the leaves of a tree. As clustering continues, the original cluster gradually splits into multiple clusters. The two types of hierarchical clustering are agglomerative (bottom-up) and divisional. Agglomerative clustering begins iteratively combines two or more appropriate clusters into one, using one object for each cluster. Divisional clustering begins with a set of data that belongs to a single cluster and after that splits the best-fitting cluster recursively. Up until the halting requirement is met, the operation is repeated, at which point

the no. of clusters (k) is introduced frequently. For the reason that a step (such as a merge or split) cannot be undone once it has been finished, the hierarchical technique has this substantial drawback. The most well-known algorithms in this area include BIRCH, CURE, ROCK, Chameleon, and Echidna..

3) Using density: In this group of methods, data elements are classified according to their connectedness and density. They have a connection to their immediate neighbors. In this case, the cluster expands toward higher object density. This class of clustering methods can create arbitrary-shaped clusters. In addition, natural emission protection is provided. This category includes the DBSCAN, OPTICS, DBCLASD, and DENCLUE methods.

4) Mesh-based: Grids are used to divide the object space. The key advantage of this approach is its quickness. This is due to the fact that the dataset is only parsed once in order to calculate statistical values for the grid: The performance of these methods is determined by the grid size, which is typically much smaller than the amount of data. The WaveCluster, STING, CLINQUE, and the typical examples of this class are the OptiGrid methods.

5) Model-based: This method strengthens the link among the data presented and some (predetermined) mathematical models. It is assumed that A set generates the data of basic probability distributions. As a result, a dependable clustering method and an automatic calculation of no. of clusters based on conventional statistics while taking noise i.e., (outliers) into consideration are produced. The model method is mostly based on the statistics and neural network schemes. The best model-based approach is probably the MCLUST method, although other effective algorithms include EM (in which a mixture density model is applied), conceptual aggregation (for example, COBWEB), and neural network approaches. To identify groups, the statistical approach employs probability measures. To represent each derived concept, probabilistic descriptions are commonly used. The neural network approach employs a network of interconnected I/O blocks, each with its own weight. Neural networks are popular for clustering due to several properties. To begin, inherently parallel and distributed information processing structures are neural networks. Second, in order to process data more effectively, neural networks learn by adjusting the weights of their relationships.

This gives them the ability to standardize or prototype. Patterns are used to extract characteristics (or features) for different groups. Third, neural networks use scalar vectors to represent object patterns and require quantitative features. In the clustering method using neural networks, each cluster is represented as a sample. The pattern serves as a block prototype and does not need to be associated with a specific object. Based on some distance metrics, new features can be assigned to the group whose example is the most similar. CLASSIT methods and SOMs are also included in this method class.

3. EVALUATING BIG DATA CLUSTERING METHODS

- Detailed evaluation criteria must be employed to compare the relative merits and shortcomings of each algorithm in terms of volume, speed, and variety when comparing big data clustering methods. Consider the key features of big data clustering methods.
- Volume refers to the clustering algorithm's ability to handle bulk quntity of data. The following criteria are considered when selecting an appropriate clustering algorithm based on volume property: (a) data set size, (b) high dimensionality processing, and (c) outliers/noisy data processing.
- Diversity refers to a clustering method's ability to manage various data types (numeric, categorical, and hierarchical). The following criteria are considered when selecting an appropriate clustering algorithm for the diversity property: (a) the dataset type, and (b) the cluster shape.
- The speed at which the big data clustering process occurs. The following criteria are considered when selecting an appropriate clustering algorithm based on speed: Algorithm complexity (a) and runtime performance (b).
- We will give a detailed explanation of the selection criteria for each property of big data. [4], [5], [6].

1) Dataset type: The majority of Traditional clustering algorithms are intended for use with either numerical or categorical data. Because merging data in the real world frequently contains both types of data at the same time, traditional clustering algorithms are difficult to apply directly to this type of data. Clustering algorithms work well with both numerical and categorical data. They do not work well with data that is both categorical and numerical.

2) Dataset size: The quality of clustering is significantly impacted by the amount of the dataset. Some clustering techniques are more effective than rest of techniques when the amount of data is small, and vice versa.

3) Input variable: For "practical" clustering, because a large no. of parameters can impair cluster quality because they depend on parameter values, the desired function has fewer parameters. Handling Outliers/Noisy Data: Because most real-world applications use dirty data, a felicitous algorithm is often capable of handling outliers/noisy data. Furthermore, the noise prevents the algorithm from grouping the object into an appropriate cluster. As a result, it has an impact on the algorithm's output.

4) Time constraint. Most clustering techniques require repeated use to increase clustering quality. Due to this, if a process takes an excessive amount of time, it may become unsuitable for big data applications.

5) Stability. Any clustering algorithm must have the capacity to produce the same data division regardless of the sequence in which the patterns are provided to the algorithm.

6) Due to the fact that several applications call for the study of objects with numerous properties, dealing with huge areas is particularly crucial in cluster analysis. As attributes, text documents, for example, may have tens of thousands of keywords or concepts. This is made difficult by the high dimensionality. Many dimensions measure become meaningless. When there are more measures, the average density of points at any location in the data is most often low, and the data becomes sparse, making measuring the distance between two points meaningless.

7) The any shape of cluster can be produced by a decent clustering algorithm, which can handle both genuine and fake data.

4. CLASSIFICATION ALGORITHMS

The following classification algorithms FCM [7], [8], [9], BIRCH [10], [11], [12] DENCLUE [13], [14] OptiGrid [15], [16] and EM [17] are considered. Next, we will consider each of them and evaluate their strengths and weaknesses.

4.1 C-Means FUZZY CLUSTERING METHOD (FUZZY C-MEANS, FCM)

Is a representative fuzzy clustering algorithm that uses k-means concepts to spilit a data set into distinct clusters. The FCM algorithm is a "fuzzy" clustering method that assigns objects to clusters with a high degree of certainty. As a result, an object can be classified as belonging to multiple clusters with varying degrees of certainty. It seeks the cluster center, which is the most distinct point in each cluster. For each feature, the degree of membership in each cluster is then calculated. Intracluster variance is also reduced by the fuzzy C-means algorithm. However, it inherits k-means problems because the minimum is only local and the resulting clusters are determined by the initial weights.

The FCM algorithm works in the similar fashion as the k-means algorithm, iteratively searching for cluster centers and updating feature membership. The primary distinction is that instead of conclusively deciding which cluster a pixel should belong to, it gives the item a number between 0 and 1 to indicate how likely it is to be in that cluster. According to the fuzzy rule, the sum of a pixel's membership values in all clusters must equal 1. The likelihood of a pixel belonging to the cluster increases as the membership value increases. When clustering, the objective function must be minimized.

$$J = \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik}^{m} |X_i - v_k|^2 \tag{1}$$

where "J" is the objective function and n denotes the no. of objects, and c is the no. of defined clusters μ_{ik} - m is the fuzziness parameter, and the possible values of assigning object I to cluster k are, $|p_i - v_k|^2$ - The i-th object, X_i, and the k-th cluster center, vk, were measured by using equation (2) to determine their Euclidean distance:

$$|X_i - v_k| = \sqrt{\sum_{i=1}^{n} (X_i - v_k)} \tag{2}$$

The k^th cluster's centroid is defined as

$$v_k = \frac{\sum_{i=1}^{n} \mu_{ik}^{m} X_i}{\sum_{i=1}^{n} \mu_{ik}^{m}} \tag{3}$$

where

$$\mu_{ik} = \frac{1}{\sum_{l=1}^{c} \left(\frac{|X_i - v_k|}{|X_i - v_l|} \right)^{\frac{2}{m-1}}} \tag{4}$$

This algorithm is extended to cluster an image's RGB color, i.e. distance calculation (2) is converted to:

$$|X_i - v_k| = \sqrt{\sum_{i=1}^{n}(X_{iR} - v_{kR})^2 + (X_{iG} - v_{kG})^2 + (X_{iB} - v_{kB})^2} \quad (5)$$

4.2. **The BIRCH algorithm**

Creates a clustering feature tree dendrogram (CFT). The DPC can be constructed by incrementally and dynamically scanning the dataset. As a result, the entire data set is not required up front. It is divided into two steps: first, the algorithm is implemented to cluster the leaf nodes after scanning the database to create an in-memory tree. The KDP is a height-balanced tree with two variables determining the branching factor B and the threshold T. The first phase entails creating the DPC for the data points, which is a highly balanced tree structure. The data point's membership in the cluster is determined if the closest cluster to the present data point can eventually be located. If not, a new cluster is formed that has a diameter higher than the specified T. A single scan of the data set by the BIRCH algorithm often yields a good result, and a few more scans usually result in results of higher quality. It can also effectively deal with noise. BIRCH, on the other hand, may not perform well when clusters are not spherical because it controls the cluster boundary using the concept of radius or diameter.

It is also order sensitive, producing different clusters depending on the order of the same input. [10].

4.3. **DENCLUE algorithm**

Analyze the cluster distribution using the combined reflective functions of all the data points. A data point influences its surroundings, according to the influence function. Clusters are then identifiable as density attractors. The overall local maxima of the density function are known as density attractors. With this method, any cluster shape may be precisely described by a simple equation with a density function kernel.[17].

Despite the fact that it requires careful selection of input parameters, for example, and, DENCLUE has several advantages over other clustering algorithms. [19]: a) It's mathematically sound and has generalized other clustering techniques like hierarchical partitioning. b) It has good clustering properties for noisy datasets; c) It enables the mathematical description of clusters of any shape to be concise. d) Grid cells are used, and only data for cells with points is stored there. It is much faster than DBSCAN and other methods since it maintains these cells using an access tree structure. It manages these cells using an access tree structure, which allows it to be much faster than DBSCAN and other algorithms.

4.4. **Algorithm OptiGrid**

Its goal is to find the best mesh partition. An assortment of preset projections is used to create the optimum cutting hyperplanes. The best cutting planes are then determined using these projections. Every cutting plane is selected to have the lowest possible point density while also dividing the dense region into two halves. OptiGrid uses a density function to find clusters after every move of building a multidimensional grid based on the best cutting planes. The algorithm was applied to the clusters recursively. Every recursive loop, OptiGrid only supports data items in dense grids from the preceding recursion cycle. [15].

Large multidimensional datasets can be clustered using this technique. However, because its recursive technique only decreases dimensions by one at a time, it might find it difficult to find clusters buried in a very big database's low-dimensional subspace. Furthermore, it is sensitive to parameter choice and is incapable of handling grid sizes greater than the amount of RAM that may be used. Furthermore, OptiGrid requires users to exercise extreme caution when selecting projections, evaluating density, and determining what constitutes the best or optimal cutting plane. Depending on the data, each situation's level of complexity is assessed.

4.5. **EM algorithm**

Its goal is to calculate the statistical model's maximum probability parameter in a range of circumstances, such as if an indirect method for solving the equations is required. The EM technique iteratively approximations the unknown model parameters through two steps, namely: E and M. Using the current model parameter values, step E involves estimating the posterior distribution of the latent variables. Based on this posterior distribution, objects are then fractionally assigned to each cluster. The assignment is determined at stage M via re-estimating the model parameters based on the maximum. When estimating model parameters, there will always be a local maximum found by the EM method. The requirement for a nonsingular covariance matrix, sensitivity to initial parameter selection, the potential for output to a local optimum, and a slow pace of convergence are all drawbacks of the EM algorithm. [18].

5. COMPARISON OF CLUSTERING ALGORITHMS

The finest large data clustering algorithm selection solely on theoretical grounds is insufficient in some cases. As a result, the empirical investigation of algorithm behavior is the primary focus.

To compare clustering algorithms, the following UCI machine learning repository public datasets were used: the dataset of network device attacks (DS1) [19], the dataset of recognition of heterogeneous human activity obtained from smartphones and smart watches (DS2) [20], the dataset of cybersecurity containing nine different network attacks through available commercial IP tracing system and IoT network (DS3) [21], and the dataset of real online retail transaction (DS4).

Each dataset's instances are randomly selected and split into training and test sets. As a result, each clustering algorithm's performance is evaluated constructing a model with the training set and comparing it to the test set. We take into consideration a comprehensive collection of indicators spanning all features connected with the experimental study of clustering in response to the growing demand for an impartial technique to contrast clustering techniques.

Unsupervised learning methods are evaluated differently than supervised learning methods. In this part, we provide a summary of the standards used to assess performance against internal and external validation indicators. The first evaluation criterion is to evaluate the quality of the data section using dataset-inherited quantities and features like the Compactness Index (CI) and the Dunn Index. The cross-validation procedure used in the evaluation of supervised learning method is analogous to the final evaluation criterion. Classification accuracy (CA), the corrected Rand index (CRIand), and normalized mutual information are evaluation criteria.

It is feasible to evaluate how successfully the clustering technology splits the data based on the right class labels when given a dataset with known class labels. Because internal indices lack centroids, they are inapplicable to some clustering algorithms. We use the measure from [22] and the Euclidean distance metric to find the cluster centroid.

The compactness index This is one of the most used measurements for assessing clusters based purely on data from the dataset. Therefore, effective clustering will result in clusters with related or close-by instances. As seen below, the computer calculates the typical separation between each pair of data points:

$$\overline{CI_i}=\frac{1}{|\Omega|}\sum_{\Omega}|x_i-w_i| \tag{8}$$

Where Ω the collection of cases (xi) that have been clustered, and W is the collection of values w_i of the cluster centroids Ω. The average of all clusters is determined as a general indicator of compactness using the formula below:

$$\overline{CI}=\frac{1}{K}\sum_{k=1}^{K}\overline{CI_k} \tag{9}$$

The "K" is the no. of clusters formed as a result of clustering. Members of each cluster should ideally be as near to each other as possible. As a result, a lower CI value indicates that the clusters are better and more compact.

The Separation Index (SI) calculates the distance between clusters. The following is the average Euclidean distance between cluster centroids:

$$\bar{SI}=\frac{2}{k^2-k}\sum_{i=1}^{k}\sum_{j=i+1}^{k}\|w_i-w_j\|_2 \tag{10}$$

SI values close to zero indicate closer clusters.

By comparing the sum of scatterings within a cluster to splits between clusters, the Davis-Bouldin Index (DBI) can detect cluster overlap. This is how it is defined:

$$DBI=\frac{1}{k}\sum_{i=1}^{k}\max_{j\neq 1}\left(\frac{\bar{C}_i+C_j}{\|w_i-w_j\|_2}\right) \tag{11}$$

Where a DBI approximately near to "0" shows that the clusters are compact as well as far apart.

The Dunn index (DI) determines not only the degree of compactness of clusters, but also the degree of separation between individual clusters.

Comparing intra-cluster distances to inter-cluster distances, DI measures separation (compactness), The definition of such an index is provided by the following equation for a certain number of clusters K:

$$DI=\frac{\min\limits_{0<m\neq n<K}\left\{\min\limits_{\forall x_i\in\Omega_m}\|x_i-x_j\|\right\}}{\min\limits_{0<m\leq K\forall}\ \max\limits_{x_i,x_i\in\Omega_m}\left\{\|x_i-x_j\|\right\}} \tag{12}$$

If the data set represents compact and well-separated clusters, then clusters are most often spaced far apart, and modest diameters are anticipated. Thus, a larger DI value indicates compact and well-separated clusters.

The percentage of data points in a clustering solution that are correctly categorised in comparison to established class labels is measured by the CA score, which is defined as:

$$CA=\sum_{i=1}^{K}\frac{\max(C_i|L_i)}{|\Omega|} \tag{13}$$

Where: C_i – instances within the i-th cluster; for all instances in the i-th cluster, L_i these are the class labels., max (C_i | L_i) - the number of instances in C_i that are labeled with the majority label in the i-th cluster, for instance, if the label l is more prevalent in the i-th cluster than any other label. The no. of instances in both the same cluster and other clusters are taken into account by the CRI and index.

$$CИRand= \frac{n_{11}+n_{00}}{n_{00}+n_{01}+n_{10}+n_{11}}=\frac{n_{11}+n_{00}}{\binom{n}{2}} \tag{14}$$

Where n_{11} represents the number of instance pairs that belong to the same cluster; The no. of pairs of instances spread across various clusters is given by n_{00}; The number of instances in a pair that are located in the same cluster in A but in distinct clusters in B is n_{10}; The number of instances in a pair that are in distinct clusters in A but the same cluster in B is called n_{01}. The CRIand value goes from 0 to 1, with a higher value denoting accurate clustering of all data instances and the cluster's exclusive inclusion of pure instances.

A well-liked external clustering validation metric for evaluating clustering quality in regard to a certain data labeling class is called normalized mutual information. The statistical data that random variables representing cluster assignments and predefined instance label assignments share is quantified by the NMI measure. Thus, the NMI is computed as shown below:

$$NMI= \frac{\sum d_{h,l}\left(\frac{|\Omega| d_{h,l}}{d_h c_l}\right)}{\sqrt{\left(\sum_h d_h \log\left(\frac{d_h}{d}\right)\right)\left(\sum_l c_l \log\left(\frac{c_l}{d}\right)\right)}} \tag{15}$$

Where dh is the no.of threads in class h, cl is the number of threads in cluster l, and dh, l is the total number of threads in class h and cluster l. And NMI value is “1” when the clustering solution completely matches the predefined label assignments; otherwise, it is close to 0.

6. RESULTS AND ANLYSIS

The outcomes of comparing clustering algorithms in terms of external and internal validity indicators will be considered first, followed by an examination of stability, runtime performance, and scalability.

The goal of validity testing is to see how well a clustering algorithm groups objects from two different populations. Using a single metric to assess the validity of clustering algorithms can lead to incorrect conclusions; thus, computational experiments were performed to determine the following indicators (CA, CRIand, and NMI) for the same data set: With such measurements, we can make use of previously understood data labels for clusters and known data sections.

Table 1 displays the results for external validity. When compared to alternative clustering methods, Table 1 demonstrates that the “EM”algorithm delivers the best clustering results for all external indicators. The FCM algorithm is the next best clustering technique in terms of external validity. The results show that, OptiGrid, DENCLUE and BIRCH have the lowest clustering quality when compared to EM and FCM.

Table 1 displays the outcomes for various clustering techniques' external validity.

Index	Algorithms	DS1	DS2	DS3	DS4
CA	DENCLUE	68.9	51.71	65.02	70.51
	OptiGrid	74.81	42.74	52.05	62.22
	FCM	75.39	59.97	67.44	73.11
	EM	81.57	62.86	74.79	79.89
	BIRCH	71.31	24.51	59.34	77.34
CИRand	DENCLUE	57.35	34.86	46.67	60.75
	OptiGrid	34.95	56.02	33.01	52.24
	FCM	59.14	39.77	50.03	59.11
	EM	69.95	45.61	56.14	66.03
	BIRCH	51.02	20.16	51.36	61.08

The results for internal validity measures are shown in Table 2. This is critical, especially when no prior knowledge of the correct dataset class label is available.

Table 2. Results of internal validity of clustering algorithms.

Index	Algorithms	DS1	DS2	DS3	DS4
CI	DENCLUE	1.99	1.01	1.49	0.83
	OptiGrid	1.63	2.45	1.2	1.97
	FCM	3.24	3.95	2.56	2.73
	EM	3.85	1.54	2.87	1.37
	BIRCH	3.19	5.53	1.83	4.13
SI	DENCLUE	2.98	1.13	2.01	0.81
	OptiGrid	1.91	3.17	1.25	2.44
	FCM	4.02	4.89	3.14	3.31
	EM	4.54	1.59	3.27	1.38
	BIRCH	3.61	6.39	1.89	4.69
DBI	DENCLUE	1.81	6.870	7.71	6.07
	OptiGrid	3.79	4.73	5.11	8.5
	FCM	3.97	10.85	10.24	9.31
	EM	8.2	11.02	10.12	2.29
	BIRCH	5.01	11.9	9.41	6.71
DI	DENCLUE	0.35	0.59	0.42	0.84
	OptiGrid	0.53	0.45	0.63	0.47
	FCM	0.55	0.51	0.52	0.51
	EM	0.52	0.64	0.56	0.71
	BIRCH	0.57	0.54	0.64	0.55

Various aspects of the clustering output are evaluated solely on the basis of raw data, with no explicit subject-area information used to evaluate the resulting cluster.

Consider the clustering algorithm's stability index calculation results. Stability is defined as the change in the output of a specific clustering algorithm. As a result, higher values indicate smaller changes in output and are always preferred. Table 3 compares the stability of the results of each clustering algorithm on all data sets. It should be noted that increasing the persistence values roughly orders the candidate clustering algorithms. Let's examine some of the most remarkable phenomena in light of the sustainability findings that were provided. First off, the majority of the time, the total stability level is below 0.599, indicating that clustering algorithms frequently have instability problems and fail to generate stable results. Second, the EM algorithm typically yields the highest compitability value across all datasets.

Table 3. Stability of possible clustering algorithms.

Algorithms	DS1	DS2	DS3	DS4
DENCLUE	0.42	0.29	0.32	0.36
OptiGrid	0.53	0.22	0.36	0.48
FCM	0.57	0.27	0.29	0.28
EM	0.5	0.48	0.44	0.46
BIRCH	0.57	0.38	0.32	0.41

To examine the efficacy of potential clustering algorithms, we realized each clustering technique to ten datasets. The execution time of each algorithm was then computed. Table 4 displays the execution times of five different clustering algorithms. As a result, DENCLUE clearly outperforms all other clustering algorithms.

To show the effectiveness of potential clustering techniques, we used ten datasets for each clustering algorithm. The execution time of each algorithm was then computed. Table 4 displays the execution times of five different clustering algorithms. As a result, DENCLUE clearly outperforms all other clustering algorithms.

Table 4. Execution time of possible clustering algorithms.

Algorithms	DS1	DS2	DS3	DS4
DENCLUE	0.37	7.13	0.23	17.35
OptiGrid	0.08	23.7	0.4	56.73
FCM	0.11	262.4	1.77	401.8
EM	3.68	1982.8	6.43	20429.28
BIRCH	1.1	241.07	1.25	364.6

A clustering algorithm classification is proposed, and the most promising algorithms in each category were evaluated using a large number of criteria and a large set of initial data.

7. CONCLUSION

We can draw the following conclusions based on the results obtained. No single clustering algorithm excels in all evaluation criteria. Except for algorithms that segment multidimensional data, namely: “EM” and “FCM” clustering algorithms produce excellent clustering output. However, these algorithms have high computational time requirements. Thus, one potential remedy is to rely on sophisticated programming language as well as on hardware technologies that enable these algorithms to be implemented more effectively. The stability parameter affects all assembly techniques. The best clustering methods for huge datasets include DENCLUE, OptiGrid, and BIRCH, mainly DENCLUE and OptiGrid, they can be able to handle high-complexity data. This paper presents the most important advantages and disadvantages of each algorithm, and thus it is possible to benefit from the analysis and classification of segmentation algorithms in IoT systems and decision-making systems.

ACKNOWLEDGEMENTS

The author sincere wishes to thank the researcher who contributed to collecting information related to the study Farah Abbac Sari.

REFERENCES

[1] Sadaaki Miyamoto, Hidetomo Ichihashi, and Katsuhiro Honda, Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications: Springer - Incorporated, 2008; URL: https://dl.acm.org/doi/book/10.5555/1817129#cited-by-sec

[2] Keller, James M., “Fuzzy Set Methods For Object Recognition In Space Applications,” NASA Johnson Space Center , United States, 1992; URL: https://ntrs.nasa.gov/api/citations/19920023849/downloads/19920023849.pdf.

[3] Gan, Guojun and Ma, Chaoqun and Wu, Jianhong, Data Clustering: Theory, Algorithms, and Applications, Society for Industrial and Applied Mathematics, 2007, https://doi.org/10.1137/1.9780898718348.

[4] Viktor Mayer-Schönberger, Kenneth Cukier, Big Data: A Revolution That Will Transform How We Live, Work, and Think, Boston New York: Eamon Dolan/Houghton Mifflin Harcourt, 2013, https://doi.org/10.1093/aje/kwu085.

[5] Nada Elgendy, Ahmed Elragal, “Big Data Analytics: A Literature Review Paper,” *Springer International Publishing Switzerland,* pp. 214-227, 2014, doi. 10.1007/978-3-319-08976-8_16.

[6] D. P. Acharjya, Kauser Ahmed P, “A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools,” *International Journal of Advanced Computer Science and Applications,* vol. 7, nº 2, pp. 511-518, 2016, doi：10.14569/IJACSA.2016.070267.

[7] Bensaid, Amine Mhamed, Improved Fuzzy Clustering for Pattern Recognition with Applications to Image Segmentation, University of South Florida, USA, 1995, URL: https://dl.acm.org/doi/book/10.5555/221058

[8] Ming-Chuan Hung, Don-Lin Yang, “An efficient Fuzzy C-Means clustering algorithm,” em IEEE International Conference on Data Mining, San Jose, CA, USA, 2001, doi: 10.1109/ICDM.2001.989523.

[9] Janmenjoy Nayak, Bighnaraj Naik, H.S. Behera, “Fuzzy C-Means (FCM) ClusteringAlgorithm: A Decade Reviewfrom 2000 to 2014,” em *Computational Intelligence in Data Mining*, Odisha, India, Springer India, 2015, pp. 133-149, https://doi.org/10.1007/978-81-322-2208-8_14.

[10] Zhang T., Ramakrishnan R., Livny M., “BIRCH: an efficient data clustering method for very large databases,” SIGMOD international conference on Management of data - SIGMOD '96., vol. 25, nº 2, pp. 103-114, 1996, https://doi.org/10.1145/235968.233324

[11] Zhang, T., Ramakrishnan, R. & Livny, M., “BIRCH: A New Data Clustering Algorithm and Its Applications,” *Data Mining and Knowledge Discovery,* vol. 1, p. 141–182, 1997, https://doi.org/10.1023/A:1009783824328.

[12] Boris Lorbeer, Ana Kosareva, Bersant Deva, Dženan Softić, Peter Ruppel, Axel Küpper, “Variations on the Clustering Algorithm BIRCH,” Big Data Research, vol. 11, pp. 44-53, 2018, https://doi.org/10.1016/j.bdr.2017.09.002.

[13] Hinneburg A., Keim D. A., “efficient approach to clustering in large multimedia databases with noise,” Proc. ACM SIGKDD Conf. Knowl. Discovery Ad Data Mining (KDD), p. 58–65, 1998, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.aaai.org/Papers/KDD/1998/KDD98-009.pdf.

[14] Hajar REHIOUI, Abdellah IDRISSI, Manar ABOUREZQ, Faouzia ZEGRAR, “DENCLUE-IM: A New Approach for Big Data Clustering,” *Procedia Computer Science,* vol. 83, p. 560 – 567, 2016, DOI: 10.1016/j.procs.2016.04.265.

[15] Hinneburg A., Keim D. A., “Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering,” *Hinneburg A., Keim D. A. Optimal grid-clustering: Towards breaking the curse of diProc. 25th Int. Conf. Very Large Data Bases (VLDB),* p. 506–517, 1999, chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://kops.uni-konstanz.de/bitstream/handle/123456789/5790/vldb99.pdf?sequence=1.

[16] Babu, B.Hari, N.Subash Chandra, and T. Venu Gopal., “Clustering Algorithms For High Dimensional Data – A Survey Of Issues And Existing Approaches,” *Babu, B.Hari, N.Subash Chandra, and T. Venu Gopal. “Clustering International Journal of Computer Science and Informatics,* pp. 293-299, 2013, doi:10.47893/IJCSI.2013.1108.

[17] Neal, R.M., Hinton, G.E., “A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants,” em *NATO Science Series D*, Jordan, M.I. (eds) Learning in Graphical Models, Springer, Dordrecht, 1998, p. 355–368, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://glizen.com/radfordneal/ftp/emk.pdf

[18] Han J., Kamber M., Data Mining: Concepts and Techniques, San Mateo: CA, USA: Morgan Kaufmann, 2006, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf.

[19] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Dominik Breitenbacher, Asaf Shabtai, and Yuval Elovici, “N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders,” *IEEE PERVASIVE COMPUTING,* vol. 13, nº 9, pp. 1-8, 201, doi: 10.1109/MPRV.2018.03367731.

[20] Stisen A., Blunck H., Bhattacharya S., and et al., “Stisen A., BlunckSmart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition,” em *Stisen A., Blunck H., Bhattacharya S., and et al. Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogen13th ACM Conference on Embedded Networked Sensor Systems*, Stisen A., Blunck H., Bhattacharya S., and et al. Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity RSeoul, Kor, Stisen A., Blunck H., Bhattacharya S., and et al. Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity 2015, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://pure.au.dk/ws/files/93103132/sen099_stisenAT3.pdf.

[21] A. N. Mahmood, C. Leckie and P. Udaya, “An efficient clustering scheme to exploit hierarchical data in network traffic analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, nº 6, p. 752–767, June 2008, doi: 10.1109/TKDE.2007.190725.

[22] Yin, L.; Li, M.; Chen, H.; Deng,W. An Improved Hierarchical Clustering Algorithm Based on the Idea of Population Reproduction and Fusion. Electronics 2022, 11, 2735. https://doi.org/10.3390/electronics11172735.

BIOGRAPHIES OF AUTHORS

	Farah Abbas Obaid Sari obtained a master’s degree in Information Technology from dr babasaheb ambedkar marathwada university in Aurangabad, India, in 2012. Currently, work at University of Kufa in the city of Najaf in Iraq. Her research interests include image processing and artificial intelligence. She can be contacted by e-mail: faraha.altaee@uokufa.edu.iq
	Ali Abdulkarem Habib Alrammahi obtained a master’s degree in Information Technology from dr babasaheb ambedkar marathwada university in Aurangabad, India, in 2012. Currently, work at University of Kufa in the city of Najaf in Iraq. His research interests include data analysis, and image processing. He can be contacted by e-mail: alia.alramahi@uokufa.edu.iq