Computer Science, Engineering and Technology

Vol: 2(4), December 2024

REST Publisher; ISSN: 2583-9179

Website: https://restpublisher.com/journals/cset/

DOI: DOI: https://doi.org/10.46632/cset/2/4/5

Comparative Analysis of Clustering Algorithms: Performance Evaluation Using the Weighted Product Method (WPM)

Vamsi Krishna Kavuri
Senior Lead Software Engineer
*Corresponding author Email: kavurivk@gmail.com

Abstract: *Introduction: Clustering algorithms play a key role in grouping data objects based on their similarities. A popular method, K-means, works by repeatedly adjusting the center of each cluster until convergence is achieved. This method, especially in the PAM form, is widely used in clustering analysis for its effectiveness in separating data. Clustering, an unsupervised learning technique, is very effective in discovering hidden patterns within datasets. Clustering focuses on dividing data into meaningful groups, rather than using predefined labels, as supervised algorithms do. By finding underlying structures and connections, this technique helps gain a deeper understanding of complex data. Research significance: This research is of great importance for the study of clustering algorithms developed for sparse industrial datasets. It aims to provide useful insights and standards for improving clustering performance in industrial settings by examining and contrasting five main clustering approaches: partitioning, hierarchical, density, grid, and model-based methods. Methology: Some alternative suppliers include K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixture, Sample, Iterated-Bisection, Agglomerative, and Clustering. Evaluation criteria include Silhouette Score, Davis-Bouldin Index, Kalinsky-Harapas Index, Cluster Cohesion, Execution Time, Memory Usage, and Sensitivity to Noise. Result: According to the results, DBSCAN was ranked highest, while Clustering was ranked lowest. DBSCAN has the highest value for Clustering algorithms according to the WPM Method approach.*

Key words: *Clustering algorithms, Gaussian mixture model (GMM), partitioning, hierarchical clustering, DBSCAN, K-means, iterative partitioning, and agglomerative clustering.*

1. INTRODUCTION

The basic idea of this type of clustering approach is to use the center of the data points as the center of the corresponding cluster. K-Fish iteratively replaces the centers of each cluster using the data points until a convergence condition is met. By selecting the data point closest to the center as common to the matching cluster, the K-Fish algorithm improves on the standard K-Fish technique, which is intended to handle discrete data. PAM is a well-known implementation of the partition-based clustering technique. [1] The aim of clustering algorithms is to find groups or clusters of objects within each group that are more similar than those in other groups. The process of creating a data model, which involves establishing a set of abstract features that provide an intuitive understanding of the main features of the dataset, is closely related to this data analysis technique. Clustering algorithms provide greater insights into complex data, but are generally more difficult than supervised procedures. The current work focuses primarily on this type of classifier. [2] Clustering is an unsupervised technique that does not rely on preset labels. Today, a wide range of clustering algorithms are used in various fields to divide datasets into meaningful groups with high-quality results. The methods differ in terms of the objective function chosen and how they generate distance and similarity matrices. Clustering algorithms are generally divided into two main types: hierarchical and partitioning. In addition, other types have been developed to cater to specific types of datasets. [3] The difficulty of clustering sparse industrial datasets remains unresolved despite numerous studies and comparative studies on clustering techniques. The purpose of this paper is to provide a comprehensive analysis of clustering methods that can effectively cluster sparse industrial datasets. To achieve this, we first analyze and divide the algorithms into five groups: model-based, density, grid, partition, and hierarchical algorithms. We then provide a comprehensive review of the selected algorithms. Finally, we conduct experimental comparisons to evaluate their performance on real-world datasets, using internal clustering validation. [4] Clustering Criteria. At this stage, it is necessary to establish a clustering criterion, which can be specified by a cost function or other matching rules. It is important to consider the type of clusters that are expected to emerge from the dataset. By doing so, we can define a useful "good" clustering criterion that ensures a partition that fits the dataset well. Verification of Results. The accuracy of the clustering algorithm's results is verified using appropriate criteria and methods. Since clustering algorithms generate clusters without predefined priorities, the final data partitioning usually requires some form of evaluation or

assessment in most applications. [5] As a result, it is not uncommon to see the same concepts and techniques being used repeatedly across multiple disciplines. Finding the best match between biological applications and clustering algorithms is of utmost importance. In addition to providing examples of biomedical applications that use cluster analysis, this review also provides biomedical researchers with a comprehensive overview of the most recent clustering techniques and helps them choose the best clustering strategies for their specific applications. [6] Partition clustering techniques are a common non-hierarchical approach to static datasets. Partition clustering iteratively improves an objective function that improves the quality of partitions in an attempt to identify groups within the data. With these methods, once the user selects a desired number k, the algorithm iteratively refines the clusters. In contrast, hierarchical clustering algorithms depict the clustering process by constructing a dendrogram, which is a binary tree-like structure. [7] We introduce the k-means clustering algorithm, which acts as a local search procedure and is a deterministic global optimization method that is independent of initial parameter values. Unlike most global clustering algorithms that randomly select starting values for each cluster center, the proposed method follows a growth path and adds a new cluster center to the best one at each stage. [8] Statistical analysis of the robustness of this new measure takes into account any potential bias. Comparing this proposed measure with the commonly used Euclidean norm, it is significantly more robust. For c-means compilation, this new metric is replaced by the Euclidean norm. Therefore, the two new compilation methods we introduce are Alternative Fuzzy C-Means (AFCM) and Alternative Hard C-Means (AHCM). [9] These classifications are influenced by a number of factors, and some methods have been developed to combine different methods. Recently, a large number of methods have been developed to solve problems in various domains. However, there is no single technique that can solve every common clustering problem. Since it is difficult to perform unified clustering at a specialized level, there are many unique clustering methods. [10] Our findings show that most partitioning strategies provide much better stable hierarchical clustering solutions than those generated by alternative aggregation algorithms. With excellent cluster quality performance and relatively inexpensive computing requirements, these results show that they are well suited for building hierarchical solutions on large document datasets. [11] With the help of clustering techniques, the communication from sensor node to sensor is reduced. However, there are some drawbacks to clustering algorithms such as additional overhead during procedures such as cluster head (CH) selection, allocation and construction. The aim of this survey is to highlight the common features, advantages and disadvantages of several clustering algorithms proposed in the literature. The components of a clustered WSN are described below. [12] Conventional clustering techniques, such as partition-based and hierarchical clustering, rely on heuristics, whereas model-based clustering techniques assume that the data comes from a mixture of multiple probability distributions (e.g., multinomial or Gaussian). These methods estimate parameters using the expectation maximum method and then calculate the covariance matrix using the mean. Bayesian or Akaike information criteria can be used to find the optimal number of clusters. A significant drawback of these techniques is that, like k-means, they may converge to a local optimum based on where the k seeds were initially placed. [13]. This section presents a classification framework that divides the many clustering methods used in the literature into distinct categories. This framework is developed from the perspective of a method developer, focusing on the technical aspects and industry standards associated with the clustering process. [14] The goal of this work is to compare and evaluate self-organizing strategies of clustering algorithms that are considered ambiguous or non-ambiguous based on different definitions found in the literature. To the best of our knowledge, there are not many such comparative studies. This may be because it is challenging to objectively compare different clustering strategies using a useful set of shared operational attributes. This limitation can be explained by the fact that cluster prototypes in probabilistic clustering are independent; that is, probabilistic clustering algorithms operate independently on each cluster in an attempt to minimize an objective function. [15] The clustering problem has been extensively studied in the fields of databases and statistics, especially in relation to various data processing tasks. It is defined as the process of identifying groups of related objects in data, where the similarity between objects is measured using a similarity function. The clustering problem is particularly helpful in the text domain, where objects need to be grouped into papers, paragraphs, sentences, or words. Clustering is very beneficial when organizing documents to improve retrieval and facilitate browsing. [16] Both normalized and non-normalized data can be used with these clustering approaches. Using normalized data reduces the number of iterations required by the approaches. Therefore, in most situations, normalized data gives better results than unnormalized data. Among the different clustering methods, density-based clustering is the most widely used data mining technique. Each clustering technique is differentiated in this article according to its unique features. Several issues related to the use of these clustering approaches are explored, focusing on the difficulties faced by these techniques. [17] Many of these rules seem very similar. Even sets of linked records may match, although some may appear to be contradictory. To find which rules match sets of similar records and to improve the presentation of results from algorithms such as the multi-objective genetic algorithm and the all-rules algorithm, various clustering approaches have been applied to the results produced by these algorithms. [18] Data grouping Clustering is the process of grouping items with similar data from one set and diverse data from other sets into distinct groupings. Clustering techniques are widely used in many fields, such as artificial intelligence, biology, customer relationship management, data mining, information retrieval, image processing, machine learning, marketing, medicine, pattern recognition, psychology, and statistics, due to the growing need to manage large datasets and extract meaningful insights. [19] Some critics question whether clustering can be used for scientific knowledge. How can the effectiveness of clustering methods be assessed without considering their inference capabilities? Although the inference capabilities of clustering methods are somewhat acknowledged, the

mathematical foundation for clustering has only recently been established. [20] There are two types of clustering: partial and complete. In the former case, some genes may not be assigned to any cluster, whereas in the latter case, each gene is assigned to a cluster. Partial clustering is more suited for gene expression analysis since gene expression data frequently contains irrelevant genes or samples. Some genes in the expression data may not be included in distinct clusters in partial clustering, as they may be noisy and have little effect on the outcome. In addition, partial clustering helps in removing redundant data contributions by removing some genes from clearly defined groups. [21] Proximity relationships serve as the foundation for basic abstraction of clustered data, in which adjacent data items have an external link. Data can be classified into clusters using this relationship. To enable clear analytical results through easy comparisons, this study will focus only on distance relationships, although other types of relationships may also exist. [22]

2. MATERIALS AND METHODS

There are two forms of mining: open-pit mining and underground mining. Mining is the process of removing minerals from the earth. The properties of the material being extracted determine which mining technique is used. Underground mining, which occurs beneath the surface of the earth, is influenced by a variety of physical, mechanical, technical, and economic factors. A multi-attribute decision-making process is required when choosing an underground mining technique because these considerations can be both qualitative and quantitative. Priority Order of Simplicity (TOPSIS), Weighted Product Method (WPM), Analytical Hierarchy Process (AHP), Simple Approach Weighted (SAW), Priority Ranking System for Enrichment Estimation (PROMETHEE), and Graph Theory and Matrix Approach (GTMA) are examples of multi-attribute decision-making procedures. WPM and PROMETHEE are used in this study to determine the best underground mining technique for a particular mineral. During the selection process, these methods provide the best results and the most accurate estimates. [23] The most effective operator is the one with the greatest area of control. As indices, the best value of the WSM and WPM scores is 1. Three MCDM techniques are used to classify workers, resulting in a convergence that helps in future selection of any of the three techniques. Since MCDM scores provide a concise, unbiased view of the criteria, workers' ratings are more accurate than expert opinions. An objective skill matrix based on MCDM scores is the final product of this effort. A ranked skill matrix is also produced from these findings. The assignment procedure is guided by the ranking of operators, which goes from the most efficient to the least efficient. As a result, this technique ensures the best team utilization and facilitates line balancing optimization. [24] The weights are calculated using AHP, and the data collected during the evaluation process is managed using fuzzy SAW and fuzzy WPM. The extracted results are shown along with a comparison of the two approaches. Although both fuzzy WPM and fuzzy SAW require multiple steps, their initial implementations are very different. To do this, we present similar and unique steps to implement these two approaches. Assuming that the weights of the criteria have been calculated, these methods are used to evaluate alternative networks for conservation laboratories in museums. The following are the grids for Fuzzy SAW and Fuzzy WPM. [25] To study crop-level water use and productivity (WP), high spatial resolution remote sensing data from sensors such as Landsat should be used. This offers a significant advantage compared to coarse resolution imaging such as MODIS. However, the lack of high resolution images is a problem. In underdeveloped countries where the precise data required for models such as METRIC or SEBAL are not available, the SSEB model is particularly helpful. For this reason, WPM investigations [26] Due to the diverse topography of the region, there are about seven waterfalls spread over several areas. For the analysis, these four waterfall locations have been selected. The feasibility of a simple hydropower project has been assessed at these four sites. Out of the four possible locations, the best one for a small hydropower project has been determined using TOPSIS and Weighted Product Methodology (WPM). [27] The Weighted Product Method (WPM) is used to determine the trust score by the machine learning model. Device sensitivity, packet error rate (PER), feedback, signal-to-noise ratio (SNR), accuracy, read range, and reaction time are behavioral features that comprise a communication system. The ability to identify small changes in the measurements that contribute to establishing a level of trust is called device sensitivity. Packet error rate (PER) is a key indicator of the performance of the data transmission medium between IIoT devices. [28] To assess the system-level impact of WPM routing, the interconnect network of a digital system is constructed using a comparable multi-level interconnect network design simulator. The simulator evaluates different design features using straightforward phrases. The simulator is used in conjunction with HSPICE and RAPHAEL to more accurately represent interconnect transients, thus increasing the accuracy of the interconnect network design. To investigate the reliability of WPM wire routing, WPM routing circuits are additionally seamlessly integrated in the enhanced MINDS version HR-MINDS. [29] A new broadband transmission method, called multi-carrier modulation, divides a high-speed data stream into multiple low-speed sub-bit streams, therefore reducing inter-symbol interference (ISI) in the system. To facilitate simultaneous data transmission, these sub-streams are modulated using different sub-carriers. Wavelet packet modulation (WPM) has several important advantages over conventional modulation techniques based on FFT, including high resistance to ISI and ICI. Therefore, it has attracted much attention. [30]

Alternative:

K-Means: This popular clustering method divides the data into k clusters by minimizing the sum of squared distances between data points and the center of each cluster. It repeatedly assigns points to the nearest center and updates the centers until convergence is achieved. Using either aggregation (bottom-up) or partitioning (top-down) approaches, hierarchical clustering creates a hierarchy of clusters. This creates a tree-like structure called a dendrogram, which graphically displays the connections between clusters and allows for adjusting cluster selection based on the required level of information. DBSCAN, or Density-Based Spatial Clustering of Applications with Noise, is a clustering method that groups closely related points together and treats points in low-density areas as outliers. Two parameters are required: a radius (epsilon) to define the neighborhood size and the minimum number of points required to create a dense area. A Gaussian mixture model (GMM) is a probability model that depicts a data distribution as a combination of many Gaussian distributions. GMMs may cluster data points based on the likelihood that each component belongs to them and spot intricate patterns in the data since each Gaussian component has a distinct mean and variance. Iterative partitioning: This clustering technique, called the iterative partitioning approach, repeatedly partitions the data into two groups while improving the clustering criterion with each round. By selecting the optimal cluster for additional partitioning, this method helps achieve k-way clustering and improves clustering performance on diverse datasets. Aggregate clustering: In this hierarchical clustering technique, each data point starts with a unique cluster. It repeatedly aggregates neighboring clusters based on a distance measure until a cluster is formed or a predetermined number of clusters is reached. This method works well for teaching data structures.

Evaluation Parameters

Benefit Parameters

1. The shadow image score shows how similar an object is to its own cluster compared to other objects. Higher numbers indicate better-defined clusters.
2. Davis-Boldin Index: A lower value denotes stronger clustering. This index compares each cluster's average similarity ratio to the cluster that is most similar to it.
3. Kalinsky-Harapas index: This index, which calculates the ratio between the total rate of spread within clusters and the spread between clusters, shows well-defined clusters.
4. Cluster cohesion: Indicates the degree of connection between objects in a cluster. Better cluster cohesion is indicated by higher cohesion.

Non-Benefit Parameters

1. Execution Time: The time taken to run the clustering algorithm. Shorter times are preferred.
2. Memory Usage: The amount of memory consumed during the execution of the algorithm. Lower memory usage is better.
3. Sensitivity to Noise: Measures how well the algorithm can handle noise in the data. Lower sensitivity indicates better robustness

3. ANALYSIS AND DISSECTION

TABLE 1. Clustering Algorithms

	DATA SET						
	Silhouette Score:	Davies-Bouldin Index	Calinski-Harabasz Index	Cluster Cohesion	Execution Time	Memory Usage	Sensitivity to Noise
K-Means	88	8	7	50	120	6	100
Hierarchical Clustering	92	7	6	40	150	7	90
DBSCAN	85	9	8	60	100	5	110
Gaussian Mixture Model:	80	6	5	30	200	8	80
Repeated-Bisection	78	5	6	70	90	4	120
Agglomerative Clustering:	95	7	7	4	180	8	45.0

This table compares six clustering algorithms, including K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixer Model, Repeated-Bisection, and Agglomerative Clustering, using metrics such as Silhouette Score, Davis-Bouldin Index, Kalinsky-Harapas Index, cluster cohesion, execution time, memory usage, and sensitivity to noise. The best performance is shown by Agglomerative Clustering, which has the lowest sensitivity to noise (45) and the highest silhouette score

(95). However, scalability can be hampered by its high memory usage (8) and execution time (180). Similarly, Hierarchical Clustering has a reasonable level of cohesion and a strong silhouette score (92), but is limited by its long execution time (150) and high memory usage (7). K-Means offers a good trade-off between reasonable execution time (120) and moderate convergence (50). However, it is less efficient for datasets with significant outliers due to its high noise sensitivity (100). On the other hand, DBSCAN performs poorly on cluster coherence (60) and clustering indices, but is relatively resilient to noise (110). Although it provides balanced clustering performance, the Gaussian mixture model has moderate noise sensitivity (80) and low coherence (30). Although it performs poorly on clustering indices, it performs exceptionally well on reuse partitioning with low memory usage (4) and high coherence (70).

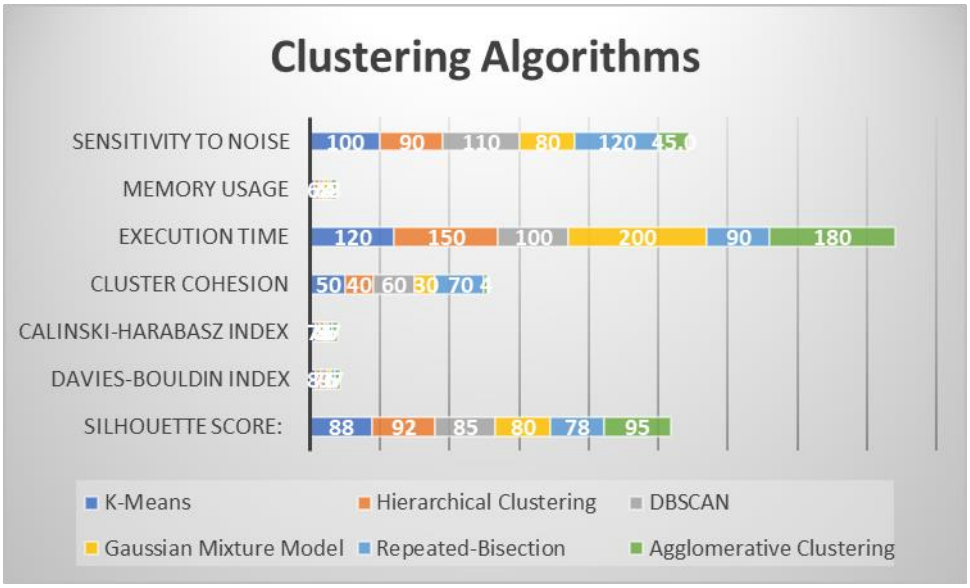

FIGURE 1. Clustering Algorithms

Six clustering techniques, namely K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixture Model, Recursive-Segmentation, and Agglomerative Clustering, are evaluated in this chart using various performance indicators. Sensitivity to noise: DBSCAN stands out for its high noise robustness (score: 45), while algorithms such as K-Means (100) and Hierarchical Clustering (90) show considerable sensitivity. Memory Usage: DBSCAN (90) and Agglomerative Clustering (180) are more resource-efficient than Gaussian Mixture Model (200) and Hierarchical Clustering (150). Execution time: K-Means (120) and Gaussian mixture model (100) provide moderate execution times, while hierarchical (150) and aggregate clustering (180) demand higher execution times. Cluster convergence: Aggregate clustering excels in cluster convergence (score: 95), while DBSCAN performs poorly (30). Kalinsky-Harapas & Davis-Boldin indices: Aggregate clustering achieves higher scores (CH index: 180, DB index: 87), indicating strong clustering performance. Silhouette score: While most algorithms perform similarly in silhouette scores, aggregate clustering leads with a score of 95.

TABLE 2. Performance value

	Performance value						
K-Means	0.93	0.89	0.88	0.71	0.75	0.67	0.45
Hierarchical Clustering	0.97	0.78	0.75	0.57	0.60	0.57	0.50
DBSCAN	0.89	1.00	1.00	0.86	0.90	0.80	0.41
Gaussian Mixture Model	0.84	0.67	0.63	0.43	0.45	0.50	0.56
Repeated-Bisection	0.82	0.56	0.75	1.00	1.00	1.00	0.38
Agglomerative Clustering	1.00	0.78	0.88	0.06	0.50	0.50	1.00

The performance of six clustering algorithms, K-means, hierarchical clustering, DBSCAN, Gaussian mixture model, reuse partitioning, and aggregation clustering, is summarized in this table using seven normalized metrics ranging from 0 to 1. Aggregate clustering clearly outperforms in terms of quality and noise tolerance, with high scores on the silhouette score (1.00), Davis-Bouldin index, and sensitivity to noise. However, its coherence score (0.06) indicates a significant deficiency in forming tight clusters. DBSCAN's remarkable performance on the Davis-Bouldin index and Kalinsky-Harapas index (both 1.00) demonstrates its ability to handle noisy datasets and perform cluster separation. Despite having a slightly lower silhouette score (0.89) than the best performers, it also achieves good cluster convergence (0.86). K-Means performs in a balanced manner, achieving high scores in the Davis-Bouldin index (0.89) and silhouette score

(0.93). However, its weak noise strength (0.45) and moderate memory usage (0.67) present difficulties in some situations. With high coherence, execution time, and memory consumption scores (all 1.00), it performs exceptionally well in repeated segmentation. Its limited adaptability is due to its low silhouette score (0.82) and weak noise sensitivity (0.38). Overall, the Gaussian mixture model produces mediocre results; however, its poor execution time (0.45) and low convergence (0.43) affect its performance.

TABLE 3. Weight

	Weight						
K-Means	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Hierarchical Clustering	0.25	0.25	0.25	0.25	0.25	0.25	0.25
DBSCAN	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Gaussian Mixture Model	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Repeated-Bisection	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Agglomerative Clustering	0.25	0.25	0.25	0.25	0.25	0.25	0.25

As shown in this table, K-means, hierarchical clustering, DBSCAN, Gaussian mixture model, recursive-segmentation, and aggregate clustering are the six clustering techniques weighted across seven evaluation metrics. The performance of each algorithm is evaluated using a balanced approach that assigns equal weight (0.25) to each statistic. This equal weighting implies that no single metric is given priority over the others, ensuring a balanced evaluation of the clustering algorithms. By assigning equal importance to all metrics, the evaluation framework considers factors such as Silhouette Score, Davis-Bouldin Index, Kalinsky-Harapas Index, Cluster Cohesion, Execution Time, Memory Usage, and Sensitivity to Noise to be equally important in determining the overall performance of each algorithm. The balanced weighting promotes fairness in comparing algorithms with different strengths. For example, algorithms such as agglomerative clustering, which excels in clustering quality metrics but lags in execution time, are rated at the same level as algorithms such as repeated-bisection, which prioritizes performance. Similarly, the noise-handling ability of DBSCAN is not underestimated because of the equal weight distribution across all metrics**.**

TABLE 4. Weighted normalized decision matrix

	Weighted normalized decision matrix						
K-Means	0.98105	0.97098	0.96717	0.91932	0.93060	0.90360	0.81904
Hierarchical Clustering	0.99201	0.93910	0.93060	0.86944	0.88011	0.86944	0.84090
DBSCAN	0.97258	1.00000	1.00000	0.96220	0.97400	0.94574	0.79975
Gaussian Mixture Model	0.95795	0.90360	0.88914	0.80911	0.81904	0.84090	0.86603
Repeated-Bisection	0.95190	0.86334	0.93060	1.00000	1.00000	1.00000	0.78254
Agglomerative Clustering	1.00000	0.93910	0.96717	0.48892	0.84090	0.84090	1.00000

This table shows the weighted normalized result matrix for six clustering algorithms evaluated on seven metrics: K-means, DBSCAN, re-partitioning, aggregate clustering, Gaussian mixture model, and hierarchical clustering. After normalization and weighting, the numbers show how well the algorithms performed, providing a complete comparison. Agglomerative clustering stands out with the highest overall performance across the key metrics, achieving a perfect score of 1.00000 on the silhouette score, Davis-Boldin index and noise sensitivity. However, it struggles with cluster coherence (0.48892), indicating challenges in forming tight, well-defined clusters. DBSCAN performs well, achieving a high score of 1.00000 on the Davis-Boldin index and Kalinsky-Harapas index, highlighting its ability to effectively separate clusters. Its convergence score (0.96220) is one of the highest, slightly lower than that of repeated-segmentation. Repeated-segmentation outperforms in execution time and memory usage, achieving perfect scores of 1.00000 on all three metrics. However, it shows relatively weak performance on noise sensitivity (0.78254), which represents a key limitation. K-Means and hierarchical clustering provide consistent results across metrics, with K-Means performing better on the silhouette score (0.98105) and hierarchical clustering performing better on noise sensitivity (0.84090). The Gaussian mixture model provides balanced but unimpressive performance, with moderate values on all metrics, and no unique strengths.

TABLE 5. Preference Score

	Preference Score
K-Means	0.58333
Hierarchical Clustering	0.48502
DBSCAN	0.68941
Gaussian Mixture Model	0.37142
Repeated-Bisection	0.59848
Agglomerative Clustering	0.31401

The preference scores, which assess how well each clustering algorithm meets a specific criterion—which can be linked to data segmentation performance or suitability for the dataset in question—are shown in Table 5. The more useful a

method is for the task at hand, the higher its priority score. DBSCAN stands out among the algorithms discussed, with the highest priority score being 0.68941. This suggests that DBSCAN would be the best choice for the data in question, likely due to its ability to detect outliers and form clusters of arbitrary patterns, which may be favorable based on the characteristics of the dataset. Although somewhat less efficient than DBSCAN, K-Means is a strong contender, coming in second place with a preference score of 0.58333. K-Means is well-liked for its ease of use and efficiency, although it can have problems with outliers or non-spherical clusters. With a preference score of 0.59848, iterative partitioning outperforms K-Means, but lags behind. On the other hand, algorithms with relatively low preference scores, such as Gaussian mixture model (0.37142), aggregate clustering (0.31401), and hierarchical clustering (0.48502), are less efficient for the task or data. This may be due to their assumptions about the underlying data structure or clustering behavior.

TABLE 6. Rank

	Rank
K-Means	3
Hierarchical Clustering	4
DBSCAN	1
Gaussian Mixture Model	5
Repeated-Bisection	2
Agglomerative Clustering	6

Table 6 shows the rankings of various clustering algorithms, based on their performance or suitability for the dataset at hand. These rankings provide a straightforward way to compare algorithms, with 1 indicating the best performance and 6 indicating the least performance. DBSCAN takes the top spot with a rank of 1, which is consistent with its high priority score in Table 5. This suggests that DBSCAN is the most suitable algorithm for this dataset, possibly due to its ability to detect outliers and generate clusters of various shapes without the need for predefined cluster numbers. Next in line is Repeated-Bisection at number 2, which is consistent with its strong priority score from Table 5. While not the best performer, Repeated-Bisection remains a solid and reliable choice for clustering tasks. K-Means takes the 3rd spot with a relatively high priority score, but it is slightly outperformed by DBSCAN and Repeated-Bisection in this context. At the other end, agglomerative clustering and Gaussian mixture modeling are ranked 6th and 5th, respectively, reflecting the weak performance indicated by their priority scores. Hierarchical clustering ranks 4th, indicating that it is less effective than some other methods in this regard.

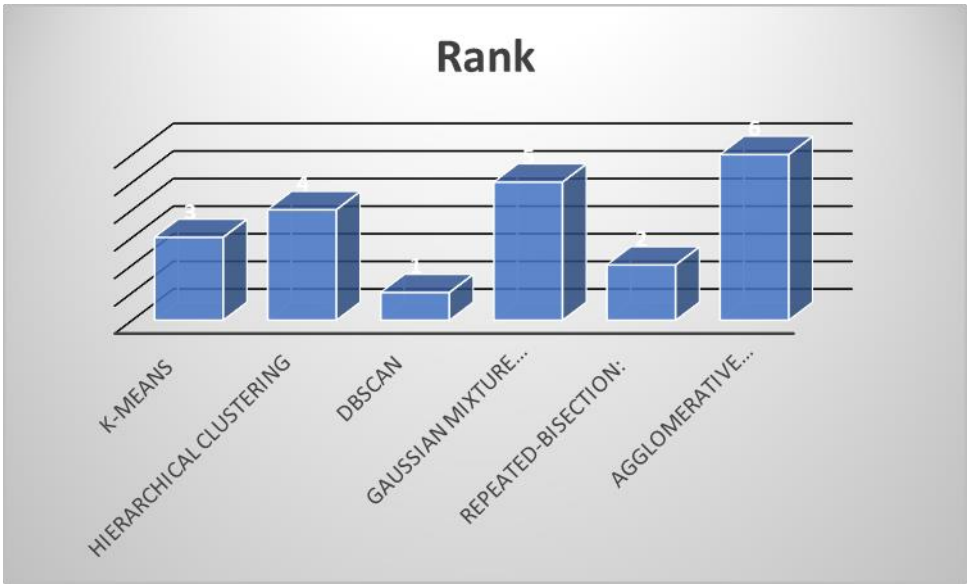

FIGURE 2. Rank

Based on their overall performance across several metrics, six clustering algorithms - K-means, hierarchical clustering, DBSCAN, Gaussian mixture model, iterative segmentation, and aggregation clustering - are ranked in a bar chart. The height of each bar represents the algorithm's rank. Agglomerative clustering emerges as the highest ranked algorithm, with the tallest bar indicating its consistent excellence on metrics such as cluster coherence, silhouette score, and Kalinsky-Harapas index. Iterative-segmentation follows closely, demonstrating strong clustering quality with efficient implementation. K-means and hierarchical clustering occupy the middle ranks. While they perform well on some metrics, their rankings are constrained by limitations such as the sensitivity to noise for K-means and the high memory usage for hierarchical clustering. The Gaussian mixture model is slightly inferior due to its significant memory requirements and

slow execution, but still provides symmetric clustering performance. DBSCAN ranks very low, primarily due to its poor cluster convergence and weak results in clustering indices. However, it excels in noise handling, making it a valuable choice for datasets with significant outliers

4. CONCLUSION

Clustering is essential in data analysis, especially when it comes to grouping related entities according to shared characteristics. Its importance has increased in domains such as industrial applications, artificial intelligence, and machine learning. The primary goal of clustering is to find meaningful clusters within data, which helps organize complex information more efficiently. Various clustering methods have been developed to improve specific parts of this process, including partitioning, hierarchical, density-based, grid-based, and model-based techniques. The objective functions, similarity measures, and structures generated by these algorithms vary, with some being more suitable for specific types of data or applications. Clustering techniques have made significant progress in increasing the efficiency and effectiveness of algorithms, particularly DBSCAN, K-means, and hierarchical clustering. Despite potential drawbacks such as initialization sensitivity and the possibility of local optimization, separation techniques such as K-means are well-liked for their ease of use and ability to handle large datasets. On the other hand, the tree-like structure of hierarchical clustering allows for flexibility in terms of cluster granularity, but typically comes at a high computational cost. Density-based techniques, such as DBSCAN, are helpful because they perform well on noisy and variable-density datasets. Despite the advantages of these techniques, difficulties still exist, especially when working with sparse industrial datasets where the inherent characteristics of the data make conventional approaches very difficult. This review looks at several clustering algorithms that have been developed to address these issues. It thoroughly examines partitioning, hierarchical, density, and other techniques, emphasizing how well they perform in various situations. The practical application of clustering algorithms in real-world decision-making is demonstrated by selecting the best sites for mining and hydroelectric projects using multi-attribute decision-making techniques such as the Weighted Product Method (WPM) and PROMETHEE. In addition, this study examines several performance evaluation metrics for clustering techniques, including the Kalinsky-Harapas Index, the Davis-Bouldin Index, and the Silhouette Score. High values for these metrics indicate coherent and well-defined groups, which help in assessing the quality of clusters. Furthermore, when choosing the best clustering method for a particular application, it is important to take into account parameters such as execution time, memory usage, and noise sensitivity.

REFERENCE

[1]. Xu, Dongkuan, and Yingjie Tian. "A comprehensive survey of clustering algorithms." Annals of data science 2 (2015): 165-193.

[2]. Rodriguez, Mayra Z., Cesar H. Comin, Dalcimar Casanova, Odemir M. Bruno, Diego R. Amancio, Luciano da F. Costa, and Francisco A. Rodrigues. "Clustering algorithms: A comparative approach." PloS one 14, no. 1 (2019): e0210236.

[3]. Nagpal, Arpita, Arnan Jatain, and Deepti Gaur. "Review based on data clustering algorithms." In 2013 IEEE conference on information & communication technologies, pp. 298-303. IEEE, 2013.

[4]. Benabdellah, Abla Chouni, Asmaa Benghabrit, and Imane Bouhaddou. "A survey of clustering algorithms for an industrial context." Procedia computer science 148 (2019): 291-302.

[5]. Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis. "Clustering algorithms and validity measures." In Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001, pp. 3-22. IEEE, 2001.

[6]. Xu, Rui, and Donald C. Wunsch. "Clustering algorithms in biomedical research: a review." IEEE reviews in biomedical engineering 3 (2010): 120-154.

[7]. Oyelade, Jelili, Itunuoluwa Isewon, Olufunke Oladipupo, Onyeka Emebo, Zacchaeus Omogbadegun, Olufemi Aromolaran, Efosa Uwoghiren, Damilare Olaniyan, and Obembe Olawole. "Data clustering: Algorithms and its applications." In 2019 19th international conference on computational science and its applications (ICCSA), pp. 71-81. IEEE, 2019.

[8]. Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." Pattern recognition 36, no. 2 (2003): 451-461.

[9]. Wu, Kuo-Lung, and Miin-Shen Yang. "Alternative c-means clustering algorithms." Pattern recognition 35, no. 10 (2002): 2267-2278.

[10]. Bindra, Kamalpreet, and Anuranjan Mishra. "A detailed study of clustering algorithms." In 2017 6th international conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO), pp. 371-376. IEEE, 2017.

[11]. Zhao, Ying, George Karypis, and Usama Fayyad. "Hierarchical clustering algorithms for document datasets." Data mining and knowledge discovery 10 (2005): 141-168.

[12]. Abbasi, Ameer Ahmed, and Mohamed Younis. "A survey on clustering algorithms for wireless sensor networks." Computer communications 30, no. 14-15 (2007): 2826-2841.

[13]. Nerurkar, Pranav, Archana Shirke, Madhav Chandane, and Sunil Bhirud. "Empirical analysis of data clustering algorithms." Procedia Computer Science 125 (2018): 770-779.
[14]. Fahad, Adil, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y. Zomaya, Sebti Foufou, and Abdelaziz Bouras. "A survey of clustering algorithms for big data: Taxonomy and empirical analysis." IEEE transactions on emerging topics in computing 2, no. 3 (2014): 267-279.
[15]. Blonda, A., and P. Blonda. "A Survey of Fuzzy Clustering Algorithms for Pattern Recognition—Part I." IEEE Transactions on Systems, Man, and Cybernetics 29 (1999): 778-785.
[16]. Aggarwal, Charu C., and ChengXiang Zhai. "A survey of text clustering algorithms." Mining text data (2012): 77-128.
[17]. Patel, KM Archana, and Prateek Thakral. "The best clustering algorithms in data mining." In 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 2042-2046. IEEE, 2016.
[18]. Reynolds, Alan P., Graeme Richards, Beatriz de la Iglesia, and Victor J. Rayward-Smith. "Clustering rules: a comparison of partitioning and hierarchical clustering algorithms." Journal of Mathematical Modelling and Algorithms 5 (2006): 475-504.
[19]. Na, Shi, Liu Xumin, and Guan Yong. "Research on k-means clustering algorithm: An improved k-means clustering algorithm." In 2010 Third International Symposium on intelligent information technology and security informatics, pp. 63-67. Ieee, 2010.
[20]. Dalton, Lori, Virginia Ballarin, and Marcel Brun. "Clustering algorithms: on learning, validation, performance, and applications to genomics." Current genomics 10, no. 6 (2009): 430-445.
[21]. Oyelade, Jelili, Itunuoluwa Isewon, Funke Oladipupo, Olufemi Aromolaran, Efosa Uwoghiren, Faridah Ameh, Moses Achas, and Ezekiel Adebiyi. "Clustering algorithms: their application to gene expression data." Bioinformatics and Biology insights 10 (2016): BBI-S38316.
[22]. Wilkin, Gregory A., and Xiuzhen Huang. "K-means clustering algorithms: implementation and comparison." In Second international multi-symposiums on computer and computational sciences (IMSCCS 2007), pp. 133-136. IEEE, 2007.
[23]. Balusa, Bhanu Chander, and Jayanthu Singam. "Underground mining method selection using WPM and PROMETHEE." Journal of the Institution of Engineers (India): Series D 99 (2018): 165-171.
[24]. Chourabi, Zouhour, Faouzi Khedher, Amel Babay, and Morched Cheikhrouhou. "Multi-criteria decision making in workforce choice using AHP, WSM and WPM." The Journal of The Textile Institute 110, no. 7 (2019): 1092-1101.
[25]. Kabassi, Katerina, Christos Karydis, and Athanasios Botonis. "Ahp, fuzzy saw, and fuzzy wpm for the evaluation of cultural websites." Multimodal Technologies and Interaction 4, no. 1 (2020): 5.
[26]. Platonov, Alexander, Prasad S. Thenkabail, Chandrashekhar M. Biradar, Xueliang Cai, Muralikrishna Gumma, Venkateswarlu Dheeravath, Yafit Cohen et al. "Water productivity mapping (WPM) using Landsat ETM+ data for the irrigated croplands of the Syrdarya River basin in Central Asia." Sensors 8, no. 12 (2008): 8156-8180.
[27]. Rana, Shilpesh C., and Jayantilal N. Patel. "Selection of best location for small hydro power project using AHP, WPM and TOPSIS methods." ISH journal of hydraulic engineering 26, no. 2 (2020): 173-178.
[28]. Khan, Mohammad Ayoub, and Norah Saleh Alghamdi. "A neutrosophic WPM-based machine learning model for device trust in industrial internet of things." Journal of Ambient Intelligence and Humanized Computing 14, no. 4 (2023): 3003-3017.
[29]. Joshi, Ajay Jayant, and Jeffrey A. Davis. "Wave-pipelined multiplexed (WPM) routing for gigascale integration (GSI)." IEEE transactions on very large scale integration (VLSI) systems 13, no. 8 (2005): 899-910.
[30]. Huang, Xian, Gewei Tan, Qingyong Xu, Ning Xu, and Shuangxi Wang. "A kind of PAPR reduction method based on pruning WPM and PTS technology." Journal of Electronics (China) 30 (2013): 261-267.

