



## Research article

# Assessing the Efficiency and Accuracy of K-Means Clustering Compared to Other Clustering Techniques

Iliyas Karim khan, Hanita Binti Daud, Nooraini binti Zainuddin, Abdus Samad Azad,  
Ahmad Abubakar Suleiman

Fundamental and Applied Science Department, Universiti Teknologi PETRONAS, Perak 32610, Malaysia.  
[iliyas\\_22008363@utp.edu.my](mailto:iliyas_22008363@utp.edu.my), [hanita\\_daud@utp.edu.my](mailto:hanita_daud@utp.edu.my), [aini\\_zainuddin@utp.edu.my](mailto:aini_zainuddin@utp.edu.my), [raja.sokkalingam@utp.edu.my](mailto:raja.sokkalingam@utp.edu.my),  
[abdus\\_22009918@utp.edu.my](mailto:abdus_22009918@utp.edu.my), [abdussamad\\_22009779@utp.edu.my](mailto:abdussamad_22009779@utp.edu.my) and [ahmad\\_22000579@utp.edu.my](mailto:ahmad_22000579@utp.edu.my)

## ARTICLE INFO

### Article history:

Received March 24, 2024  
Revised January 23, 2025  
Accepted July 05, 2025  
Available online August 07, 2025

### Keywords:

Accuracy, Execution Times,  
Comparative Analysis and  
Clustering Algorithms

### Please cite this article in IEEE style as:

I. Khan, "Assessing the Efficiency and Accuracy of K-Means Clustering Compared to Other Clustering Techniques", Data Science Insights, vol. 3, no. 2, pp. 49–65, Aug. 2025.

## ABSTRACT

Clustering is an important method in data analysis, faces challenges due to the different nature of datasets, resulting in certain algorithms being less effective and taking a long time. Choosing the most effective clustering method involves evaluating its accuracy and computational speed for a dataset poses a significant challenge for today's researchers. To address these issues, current study compares different clustering methods, by using datasets, including iris, seed, and well log to evaluate their accuracy and execution speed. Results show that K-means performs better with large datasets. As sample size increases, the accuracy of the K-means algorithm tends to improve. The execution time of k-means is influenced by the number of features in the dataset, with datasets having a larger number of features typically requiring more time to process. Mean shift algorithm and spectral clustering algorithm are performed well in small data sets, but it takes a long time.

Correspondence:  
[iliyas\\_22008363@utp.edu.my](mailto:iliyas_22008363@utp.edu.my), Mobile  
Number: +60103682060

Data Science Insights is an open access under the with [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## 1. Introduction

The partitioning of the input space into separate clusters based on a common theme is known as clustering or unsupervised pattern classification. The purpose of a clustering algorithm is to carry out a partition such that all the objects within a cluster are like each other and the objects among different clusters are unlike [1, 2]. Hence, the aim of clustering is to detect the natural forms in the given set and clusters are popularly used in various applications including psychology[3], biology, pattern recognition [4, 5], image processing[6], and computer security[7]. Once a clustering algorithm has processed a dataset and obtained partition of the input data, a relevant question arises: Does the proposed partition solve the problem of input data? The reason behind this question is of great importance in many ways. Secondly, there exists no optimal clustering algorithm. Therefore, various algorithms — or even similar configuration of the single algorithm lead to alternative divisions which were never deemed to be optimum under all circumstances[8]. Hence, in developing efficient clusters, there ought to be a calculation of various segments of partitioning among which we should take the one that matches better the data. Secondly, many clustering algorithms cannot ascertain the number of natural clusters within the data, thus the algorithm is normally provided with this value – typically called the k parameter. The common procedure applied is to use the algorithm multiple times for each run of the k value. Subsequently, each partition is evaluated and determined to choose the best that suits the supplied data. Clustering accuracy measures how close the generated clusters are to the real structure or ground truth in the given dataset[9]. This evaluates how well data points are put together as a cluster. Typically, metrics such as the adjusted rand index and silhouette score are used. Validating unsupervised results is difficult because there is no labelled

ground truth for determining cluster accuracy. Varying cluster shapes, sizes, and densities generate ambiguity that affects determination of proper quality of clusters. Moreover, an accurate evaluation is complicated through sensitivity to algorithm parameters as well as the selection of evaluation metrics[10]. This article aims to comprehensively evaluate and compare various clustering methods, K-means, Hierarchical Agglomerative Clustering, DBSCAN, Mean Shift, and Spectral Clustering. The assessment involves applying these methods to different datasets—iris, seed, and well log data—with varying observation counts. The goal is to analyze their performance based on accuracy and execution time. The results highlight K-means clustering as a robust performer, particularly excelling in accuracy and efficient processing with larger datasets. However, when considering the iris dataset, Spectral Clustering emerges as the most accurate among the mentioned algorithms. In the case of seed data, Mean Shift demonstrates favorable results, albeit with longer execution times compared to the other methods. Notably, K-means struggles in scenarios involving noisy data and isn't suitable for smaller datasets. Future research endeavors should focus on enhancing K-means clustering's resilience to noise and adaptability for smaller datasets.

The main goals of this study are outlined as follows:

1. Assessing the accuracy of various clustering algorithms.
2. Evaluating the execution time of these clustering algorithms.
3. Analyzing the performance of these algorithms across various datasets with differing features.

The literature review is presented in section 2, while section 3 includes methodology highlighting different clustering algorithms like K-means, Hierarchical Agglomerative Clustering, DBSCAN, Mean Shift, and Spectral Clustering. Section 4 deals with experimental results and analyses adopting iris, seed, and well log data sets. Finally, Section 5 involves discussions on research findings including contributions and conclusions.

## 2. Literature Review

Clustering is the essential procedure in unsupervised machine learning where each data point is grouped with others like it to discover hidden relations or patterns without label. This process involves employing various algorithms such as K-Means [11], DBSCAN [12], hierarchical clustering[13], mean shift clustering algorithm[14] and spectral clustering algorithm[15] with the intention of organizing data into clusters that have similar characteristics. Optimal number of clusters and choice of distance measure are important issues, and clustering is applied for instance to market segmentation, image analysis or anomaly detection. Notwithstanding, the selection of an algorithm and sensitivity to noise should be considered while undertaking clustering analysis[16]. Studied a new GPU based K-means; ASB-K-means is introduced which is faster than current GPU based k-means algorithms thus allowing K-means Clustering of huge sets of data that surpasses the available GPU memory[17]. A Centroids-Guided Deep Multi-View K-means method linking DL with MVC is proposed by the paper. This centers on cluster centroids to help it with deep representation learning giving the K-means friendly representations. Its effectiveness in multi-view task is evaluated by showing that this approach leads to improved clustering and closer-to-cluster-semantic matching of representations across datasets [18]. This paper tackles major data concepts like why big data is important and the role of data in clustering cases. It presents a new version of Mahalanobis Distance based K-Means clustering scheme combining partitioning and correlation method. The evaluation also demonstrates why it is better than other methods by clustering similar data with more precision than that of the current techniques [19]. The ratio-cut polytope, a crucial component of K-means and spectral clustering, was introduced to analyze the structure of these algorithms and derive inequalities. A new linear programming relaxation of K-means was developed, along with specific recovery conditions for two clusters based on a stochastic ball model. This relaxation outperforms any previous guarantee, demonstrating that the LP consistently recovers clusters when the distance between two centers exceeds one plus a root of three. Cluster recovery experiments showed that this approach surpassed semidefinite programming relaxation [20]. A modified hierarchical grouping method, Hierarchical++, was proposed in the paper. This method uses initial seeds placed in golden boxes to maintain disconnected groups. The aim was to improve clustering results without any intermediate rearrangements. In experiments, Hierarchical++ showed better results compared to traditional hierarchical techniques (single-link, complete-link, etc.), K-means, and K-means++[21]. The studied document provides an overview of hierarchical clustering in astronomy. It discusses how it traces its origin, its use in various astronomical scales, and revealing celestial hierarchy while at the same time classifying objects. By elaborating on them, it explains how these algorithms work under these conditions, what they can or cannot do and allow reliable astronomical discovery [22, 23]. The paper introduced a novel hierarchical clustering technique named HMC (Hierarchical Means Clustering). Instead of relying on traditional non-hierarchical methods, HMC employed nested partitions and estimated centroids using least squares to minimize within-cluster deviance across  $n$  partitions in the hierarchy. This resulted in a cascade of  $(n-1)$  divisions, each with minimal overall cluster variance, ranging from two clusters to  $n$  clusters. The paper presented six case studies comparing HMC to established model-based hierarchical clustering algorithms like k-means, Ward's method, and Bisecting k-means. [24]. This article presented a novel approach based on hierarchical clustering to tackle the small files problem in Hadoop Distributed File Systems (HDFS). By leveraging Dendrogram analysis, the technique compared file structures to identify and recommend optimal file consolidation strategies for efficient storage. In a simulation involving 100 CSV files with diverse structures, the algorithm successfully identified and recommended merging seven specific files, leading to a reduction in memory consumption. The results demonstrated the proposed algorithm's effectiveness in enhancing

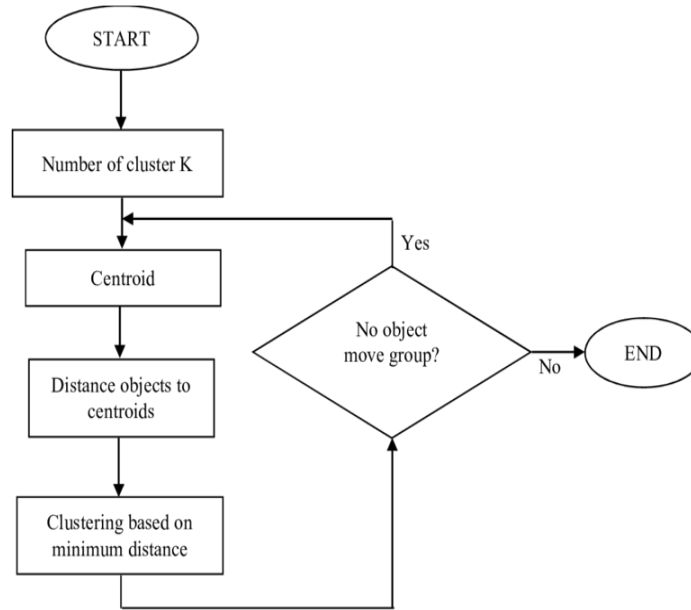
file management efficiency [25]. HY-DBSCAN a parallel DBSCAN algorithm paper. Modified KD-tree for decomposition, spatial indexing via grids, and the distributed merging scheme. When it comes to performance of DBSCAN implementation in distributed system on scientific datasets, they are superior to existing solutions up to 2048 cores leveraging the process and thread parallelization [26]. The paper enhances DBSCAN as an efficient clustering technique used in various polygonal-shaped databases. This drastically reduces computation costs by taking samples from an operational subset within that region where the samples can be calculated against. Its speed is tested and has been found to be superior to various recent approaches with only slightly reduced accuracy as compared to conventional DBSCAN [27]. It presents STRP-DBSCAN, a parallel DBSCAN method for clustering of spatial temporal trajectory data. The approach helps in increasing the computation distribution and reducing communication overhead, which can reduce the clustering time up to 96.2% in total. It introduces PER-SAC as a deep reinforcement learning-based technique for automatically tuning DBSCAN's parameters which results in better accuracy of 8.8% and higher than other parameter tweaking strategies [28]. This paper suggests a revised DBSCAN algorithm for detecting anomalies of seasonally correlated time-series data. The new method is also indicated in relation to conventional DBSCAN for seasonal sets of data allowing detection of both inter-annual and within year abnormalities. Although DBSCAN finds 2.21% more anomalies (i.e., 3.92), the refined method identifies 4.79%, which represents an enhancement of 2.16% by the proposed method. Thus, the modified DBSCAN algorithm has higher efficiency in detecting local abnormalities in seasonal data [29]. This paper introduced RMS, a novel clustering algorithm that merged update equations from MS and BMS. Notably, RMS lacked the convergence threshold present in the similar NN-BMS algorithm, making it an appealing alternative. Combining both kernel and NN configurations, RMS leveraged the BMS convergence theorem to guarantee its convergence. Empirical tests on artificial and real-world data demonstrated that RMS outperformed both MM and MBMS clustering algorithms. Furthermore, RMS and MS exhibited wider attraction basins, leading to superior performance with smaller kernel apertures or fewer nearest neighbors. While NN-BMS required a convergence threshold, the NN version of RMS eliminated it entirely [30]. Data gathering often involves conflicting information, necessitating the selection of a single, most reliable source. Truth discovery methods, popular for integrating diverse data types, usually aim to summarize everything into a single consensus. While excluding outliers before consolidation might seem logical, it's often challenging. Our approach employed the mean shift clustering algorithm to identify and remove abnormal data, ultimately uncovering the truth value. This strategy has proven successful in various settings [31]. The paper focused on problems related with SC, introducing AFDSC, a combination of attribute fluctuation and density peak clusters, as well as AFHC, a variation of attribute fluctuation-based algorithms. Using evaluation, it was discovered that these algorithms performed better than other clustering algorithms on fifteen UCI datasets [15]. This paper was focused on reconsidering the effectiveness of spectral clustering algorithms in water distribution systems that are based on graph theory. The study evaluated these algorithms in several water networks with different sizes using diverse clustering techniques and measures of performance. The study showed that the efficiency of clustering techniques, such as K-means, for partitioning is not consistent across network platforms. On the other hand, PAM is a slightly better alternative regarding modularity and k-means as well as hierarchic clustering's were faster in terms of internal indices. The various stability indices supported the use of PAM and CLARA as effective algorithms [32]. It presented a differential privacy-based privacy protecting spectral clustering algorithm. The measure was meant to add some noise in the inputs prior to clustering with an aim of protecting against sensitive information leakage. The stability, usefulness and efficiency of the algorithm for privacy risk reduction with high cluster precision was demonstrated by experimental results [33].

## 2. Research Methods

Within this section, we delineate the clustering algorithms namely, K-means, hierarchical clustering, DBSCAN, Mean Shift, and spectral clustering employed in this study. This includes detailed algorithm flowcharts and the fundamental equations crucial for their functionality.

### 3.1 k-mean clustering.

K-means clustering stands out as a top algorithm in the realm of unsupervised machine learning. The clustering process is visually represented in the flowchart below.



**Fig 1.** Flow chart of k-mean clustering algorithm.

Above Fig 1 shows a partitioning algorithm called k-means which splits a dataset into k homogenous sets without any overlaps. It is through the iterative assignment of data points to their closest clustering centroid with subsequent updates to these centroid points to mean of the data assigned which minimize within cluster sum of square. A clustering technique aims at minimizing intra-cluster variance and maximizing inter-cluster separation as measured by some typical distance like the usual Euclidean one[34]. K-means clustering is an algorithm that is iterative, so it performs the operations of allocating data point into clusters and of updating centroid. It is, however, not as direct as several other algorithms. The main steps of the algorithm can be broken down into:

Assign data points to cluster.

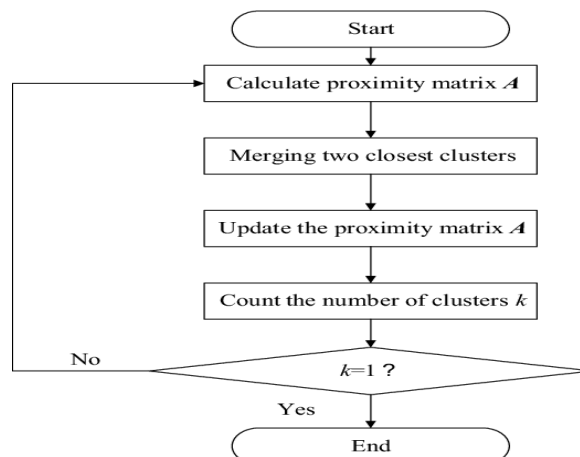
$$C_i = \arg.\min_j \|x_i - \mu_j\|^2 \quad (1)$$

Where  $C_i$ : cluster to which data points.  $x_i$   $\mu_j$ : centroid of clusters.  $\|x_i - \mu_j\|^2$ : Euclidean distance Updating cluster centroid  $\mu_j = \frac{1}{\|C_j\|} \sum_{x_i \in C_j} X_i$ ,  $\sum_{x_i \in C_j} X_i$ : summation of all data points in clusters j

The process will be iterative, and convergence will be achieved when the assignments and centroid stop changing or if a stopping criterion is reached. K-means does not present a general equation which is also the case with linear regression and others.

### 3.2 Hierarchical Clustering Algorithm

Unsupervised machine learning uses Versatile Technique for Hierarchical Clustering where it structures data in HCL (Hierarchical Cluster Learning). Unlike others, however, it does not involve enumerating the number of clusters prior. Figure 2 below illustrates all the steps involved in the performance process.



**Fig 2.** Hierarchical clustering algorithm

The above Fig 2 shows as follow

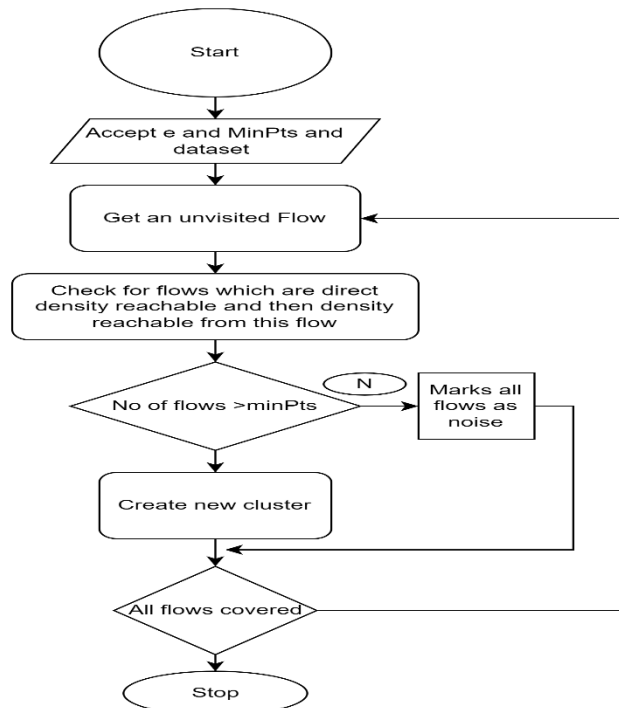
**Start:** Begin with each data point as a separate cluster. **Calculate Proximity Matrix (A):** Compute pairwise distances/similarities between data points. **Merge Closest Clusters:** Iteratively merge the two closest clusters based on a chosen linkage criterion. **Update Proximity Matrix (A):** Recalculate distances between the merged cluster and all remaining clusters. **Count Clusters:** Continue merging until reaching a desired number of clusters or a stopping criterion. Hierarchical clustering progressively merges clusters based on proximity, updating the cluster structure until a hierarchy is formed or a stopping condition is met [24, 35]. In hierarchical clustering, various linkage methods determine how the distance between clusters or data points is computed. Here are the equations for three common linkage methods[36]. Single linkage method equation  $d_{\text{single}}(A, B) = \min \text{Error! Bookmark not defined.}$  A and B is defined as the minimum distance between any pair of points, one from each cluster. Complete linkage defines as  $d_{\text{complete}}(A, B) = \max \{\text{dist}(a, b) | a \in A, b \in B\}$  The distance between clusters A and B is defined as the maximum distance between any pair of points, one from each cluster.

$$d_{\text{average}}(A, B) = \frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} \text{dist}(a, b) \quad (2)$$

The distance between clusters A and B is defined as the average distance between any pair of points, one from each cluster. There, “dist(a,b)” designates the separation or difference of single data points a and b. The linkage procedure in hierarchical clustering is based on these equations that calculate the distances between clusters.

### 3.3 Density-based spatial clustering of applications with noising (DBSCAN)

An instance of the algorithms commonly utilized in data mining and machine learning is called DBSCAN, which is a density-based clustering algorithm. It associates points in proximity and defines them as clusters which are regions of high density surrounded by low density regions. It is reliable in different kinds of analysis because it can recognize outliers as noise.

**Fig 3.** Flow chart for DBSCAN.

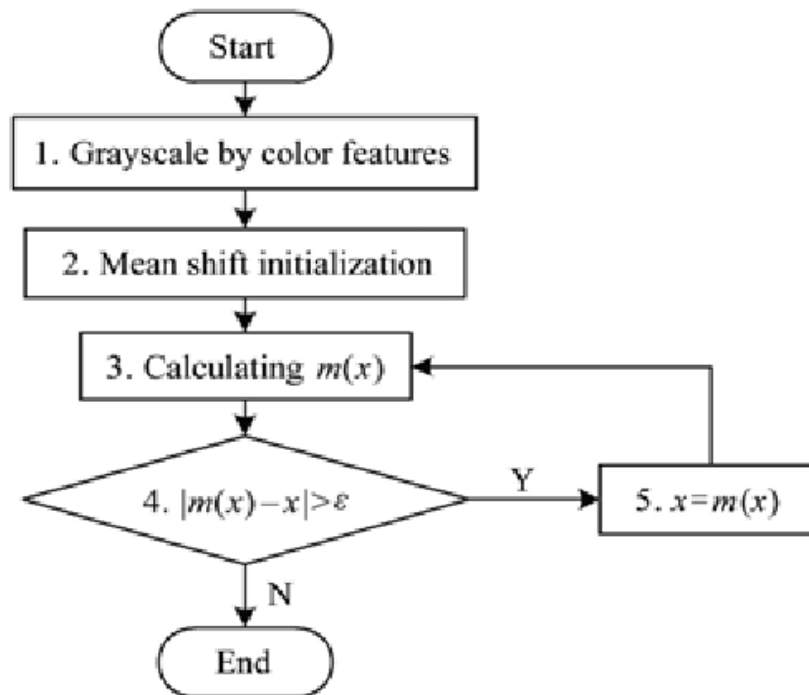
The detailed flowchart outlining the steps for density-based spatial clustering of applications with noise (DBSCAN) of Fig3 are as follows.

**Start:** Initialize the algorithm by providing the dataset, parameter  $\varepsilon$  (epsilon), and parameter MinPtsMinPts (minimum number of points required to form a dense region). **Accept Parameters:** Accept input parameters  $\varepsilon$  and MinPtsMinPts. **Initialize:** Mark all points in the dataset as unvisited. **Get an Unvisited Point:** Select an unvisited point from the dataset. **Check Density Reachability:** Check for points within  $\varepsilon$  distance from the selected point. If the number of points within  $\varepsilon$  is greater than or equal to MinPtsMinPts, proceed to the next step. **Create New Cluster:** Start a new cluster with the selected point as the core point. Add all directly density-reachable points (points within  $\varepsilon$  distance) to the cluster. If a point is a core point, repeat steps 5 and 6 recursively to expand the cluster. **Mark as Visited:** Mark all points in the cluster as visited. **Repeat or Stop:** If there are unvisited points remaining in the dataset, repeat steps 4 to 7. Otherwise, terminate the algorithm. **Check for Noise:** Identify points that are not assigned to any cluster as noise points. **Output Clusters and Noise:** Output the formed clusters and noise points. **End:** End of the algorithm.

This flowchart outlines the step-by-step process of DBSCAN clustering, where clusters are formed based on density connectivity, and points that do not belong to any cluster are identified as noise points [12, 27].

### 3.4 Mean Shift Clustering Algorithm

Adaptive clustering Mean Shift which works in an iterative mode moving data points towards the point of high densities or the centroid. It's mode-search procedure is self-defined such that it does not need prior clusters number but can adjust itself for any shape or size. It is computationally expensive as it defines an area for density estimation using a bandwidth parameter that is widely used in the image segmentation and the unsupervised clustering tasks.



**Fig 4.** Mean shift clustering algorithm

Fig 4 explains the detailed flowchart outlining the steps for Mean Shift clustering:

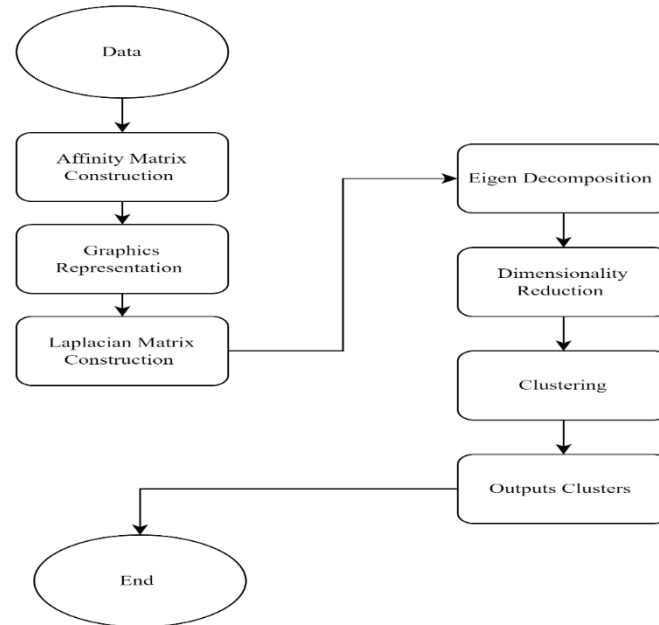
**Start:** Initialize the Mean Shift clustering algorithm. **Grayscale by Color Features:** If the data contains color features, convert them to grayscale. This step ensures that the clustering algorithm operates on a consistent feature space. **Mean Shift Initialization:** Initialize the centroids for each data point. This can be done randomly or using a specified initialization method. **Calculate  $m(x)$ :** For each data point  $x$ , compute the mean shift vector  $m(x)$ . The mean shift vector is calculated as the weighted average of the data points within a certain radius  $hh$  from  $x$ . The weight of each point is determined by a kernel function, such as the Gaussian kernel.  **$m(x) - x > \varepsilon$  Check:** Check if the mean shift vector  $m(x)$  minus the original data point  $x$  is greater than a specified threshold  $\varepsilon$ . If  $m(x) - x > \varepsilon$  for any data point  $x$ , proceed to the next step. Otherwise, terminate the algorithm. **Shift Data Points:** Update each data point  $x$  by shifting it towards  $m(x)$ . This step moves each data point towards the mode of the local data distribution. **Repeat or End:** If any data point has moved significantly (i.e.,  $m(x) - x > \varepsilon$ ), repeat steps 4 to 6. Otherwise, terminate the algorithm. **Output Clusters:** Assign each data point to its nearest mode. The modes represent the cluster centers. **End:** End of the algorithm. This flowchart outlines the step-by-step process of Mean Shift clustering, where data points are iteratively shifted towards the modes of their local data distributions until convergence. Finally, the algorithm assigns each data point to its nearest mode, resulting in the formation of clusters.

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K\left(\frac{\|x_j - x_i\|}{h}\right) * x_j}{\sum_{x_j \in N(x_i)} K\left(\frac{\|x_j - x_i\|}{h}\right)} - x_i \quad (3)$$

Where,  $x_i$  and  $x_j$  are data points.  $N(x_i)$  represents neighbor of data points and  $K$  is the kernel function, typically Gaussian, determining the weight of each point in the neighborhood based on its distance from  $x_i$ . While this equation describes the mean shift vector computation, the clustering process in Mean Shift involves iteratively applying these shifts to converge data points towards local density peaks, leading to the identification of clusters [37, 38].

### 3.5 Spectral clustering Algorithm

Spectral clustering is a graph-based method that partitions data by leveraging eigenvalues and eigenvectors to identify similarities between data points. It transforms data into a lower-dimensional space for clustering, emphasizing relationships between points regardless of their geometric arrangement, making it effective for non-linearly separable data. Spectral clustering excels in segmenting complex structures and is commonly used in image segmentation and community detection tasks.



**Fig 5.** Spectral clustering algorithm

The above Fig 5 shows the detailed flowchart outlining the steps for Spectral Clustering:

**Start:** Initialize the Spectral Clustering algorithm. **Data:** Input the dataset. **Affinity Matrix Construction:** Construct the affinity matrix  $W$  based on pairwise similarities between data points. Common methods for affinity computation include Gaussian kernel,  $k$ -nearest neighbors, or epsilon-neighbors. **Graphical Representation:** Represent the dataset as a weighted graph, where nodes represent data points and edges represent pairwise affinities. The affinity matrix  $W$  serves as the adjacency matrix of the graph. **Laplacian Matrix Construction:** Compute the Laplacian matrix  $L$  from the affinity matrix  $W$ . There are different types of Laplacian matrices: unnormalized, normalized, and symmetric normalized. **Eigen Decomposition:** Perform eigen decomposition on the Laplacian matrix  $L$  to obtain its eigenvectors and eigenvalues. The number of eigenvectors selected depends on the desired number of clusters. **Dimensionality Reduction:** Select the eigenvectors corresponding to the  $k$  smallest eigenvalues, where  $k$  is the number of clusters. These eigenvectors form a lower-dimensional representation of the data. **Clustering:** Apply a standard clustering algorithm (e.g., KMeans) to the reduced-dimensional space formed by the selected eigenvectors. The number of clusters is determined by the number of eigenvectors selected in the previous step. **Output Clusters:** Assign each data point to its corresponding cluster based on the clustering results. **End:** End of the algorithm.

This flowchart outlines the step-by-step process of Spectral Clustering, which involves constructing an affinity matrix from the data, representing the data as a graph, computing the Laplacian matrix, performing eigen decomposition, reducing dimensionality, and finally, clustering the data in the reduced space [15].

#### 4. Result

In this section, we provide an overview of the datasets, evaluation criteria, clustering algorithms targeted for optimization, initial methodologies used as a benchmark, and the configurations of parameters applied in the study.

##### 4.1 Datasets

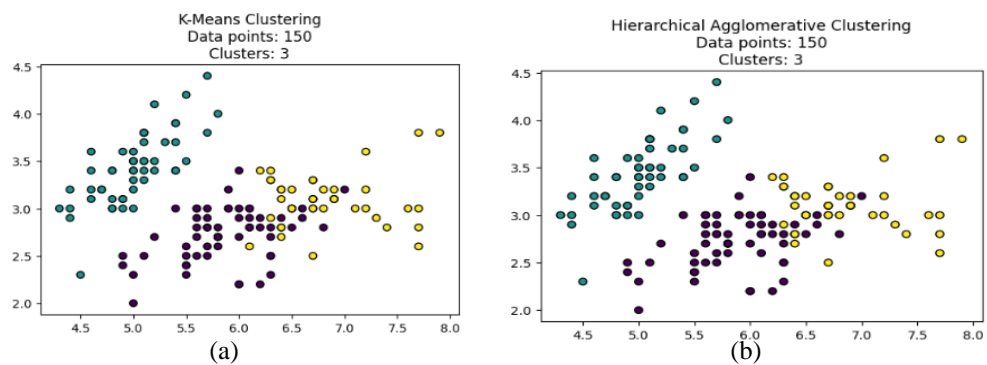
In Table 1 The provided data reveals insights into various measurements across different datasets. For iris data, Setosa exhibits a slightly negatively skewed sepal length distribution with a mean of 5.76 cm and a standard deviation of 0.80 cm, while Versicolor displays a slightly negatively skewed sepal width distribution with a mean of 3.07 cm and a standard deviation of 0.53 cm. Virginica's petal width distribution is negatively skewed with a mean of 3.46 cm and a standard deviation of 1.75 cm. Moving to seed data, Kama and Rosa seeds both have slightly negatively skewed length distributions, with means of 14.80 mm and 14.55 mm, and standard deviations of 2.90 mm and 1.30 mm, respectively. Canadian seeds exhibit a moderately flat distribution of gamma ray values with a mean of 0.87 and a standard deviation of 0.02. In well log data, the gamma ray values display a positively skewed distribution with a mean of 44.72 and a standard deviation of 26.60, while the sonic log values also show a positively skewed distribution with a mean of 0.06 and a standard deviation of 0.064. These insights offer a comprehensive understanding of the central tendency, variability, and distribution shape across the measured features within each dataset.

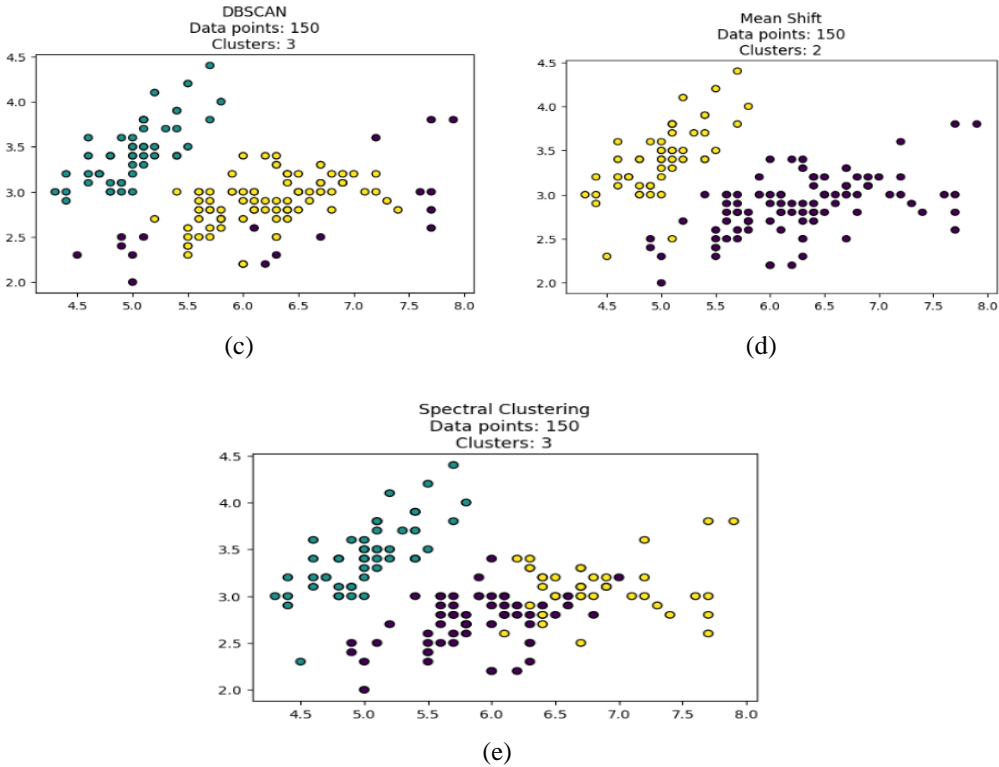
**Table 1:** The Description of Dataset

Measure	Iris Data			Seed Data			Well Log Data	
	Setosa	versicolor	virginica	Kama	Rosa	Canadian	Gamma Ray	Sonic log
Mean	5.76	3.07	3.46	14.80	14.55	0.87	44.72	0.06
S. D	0.80	0.53	1.75	2.90	1.30	0.02	26.60	0.064
Kurtosis	-0.19	-0.16	-1.43	-1.08	-1.10	-0.14	0.72	1.131
Skewness	0.46	0.31	0.39	0.38	-0.53	0.13	1.08	1.21
T Observation	150			210			2435	

##### 4.1.1 Iris data and applying different clustering Methods.

In unsupervised clustering different clusters methods are used i.e. K-mean, Hierarchical Agglomerative clustering, Density-based spatial clustering of applications with noising (DBSCAN), Mean shift data points, Gaussian mix data points and spectral clustering.



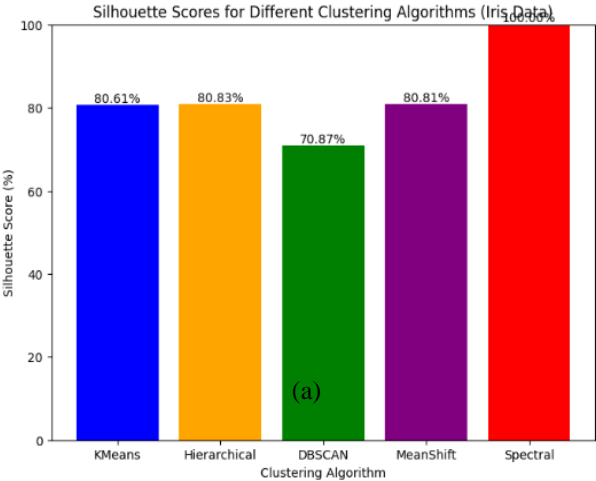


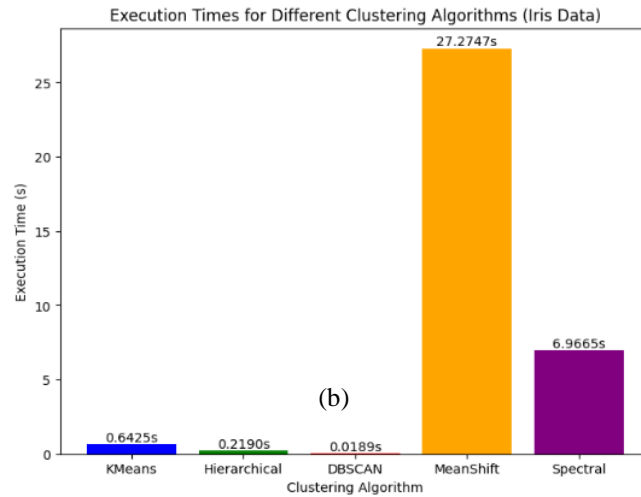
**Fig 6.** Clustering algorithm performance by using iris dataset.

Fig 6(a) k-mean clustering (b) Hierarchical Agglomerative clustering (c) DBSCAN clustering (d) Mean shift clustering (e) spectral clustering algorithm. Iris data set are not well defining data have outliers and non-spherical data. Above all method give three number of clusters accept mean shift clustering algorithm performed two number of clusters. K mean clustering also not performed well because of outliers and non-spherical data.

**4.1.2 Accuracy level and execution time of each clustering algorithm**

Evaluation of each method’s effectiveness is done based on its precision and performance speed. However, the preferred approach is faster and more correct. Figure 1: To evaluate the efficiency and accuracy in various clustering techniques, we compare their correctness and processing time.



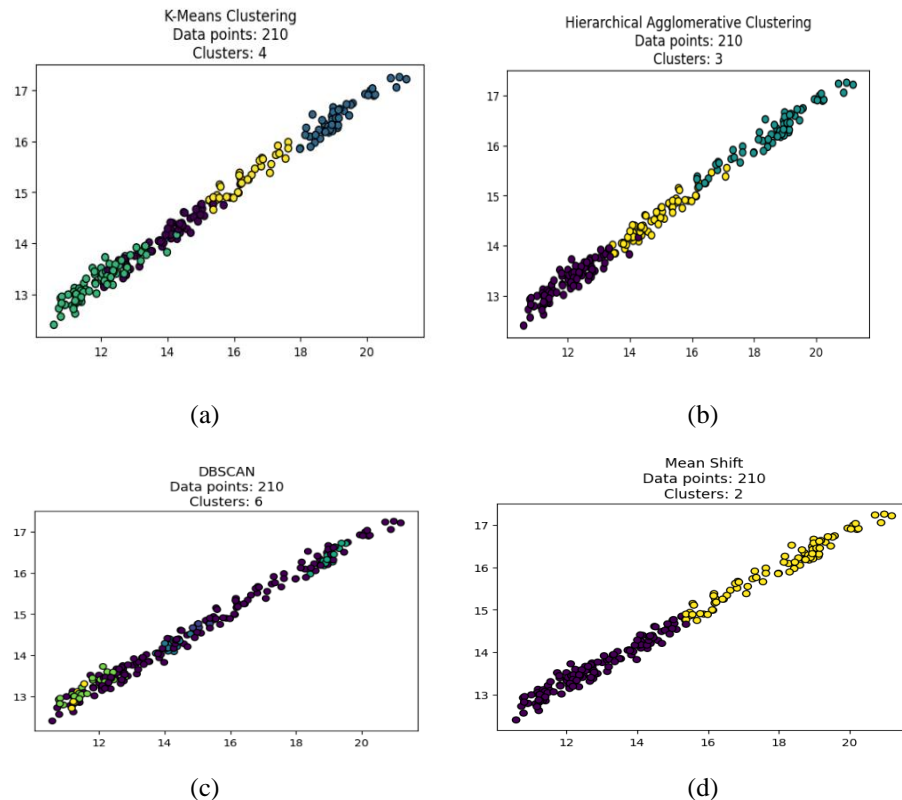


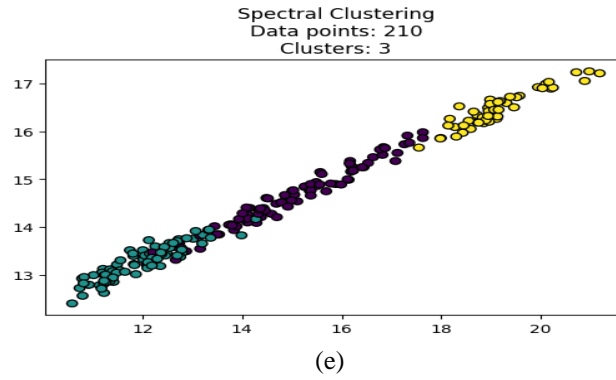
**Fig 7.** Accuracy scores and Execution times of clustering algorithms.

Fig 7(a) accuracy level of different clustering algorithm (b) time execution of the algorithms. In this figure accuracy level of spectral clusters become 100% but executive time of spectral clustering become more 6.969 second and mean shift accuracy level is good 80.81but executive time become 27.27 second. DBSCAN clustering take less execution times as compared to the other methods.

#### 4.2 Seed data and different clustering algorithms

A total of 210 observations from the seed data set were used in evaluating the performance of different clustering techniques. This dataset was run through all these clustering algorithms like K-means, Hierarchical Agglomerative Clustering, DBSCAN, Spectral Clustering, and Mean Shift. This aim was to assess and determine which algorithm offered an ideal way of dividing the seed set into useful groups. This thorough assessment has been undertaken with an intention of identifying the strength of the cluster methods using the 210 data points, and the ability of such techniques to group these observation points into distinct clusters according to underlying trends, correlations, or similarities.



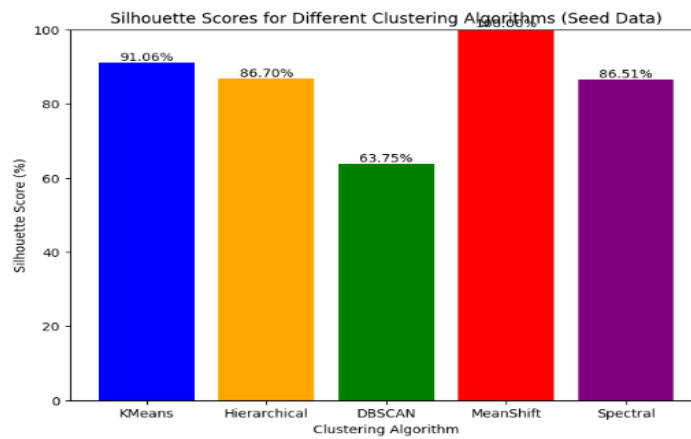


**Fig 8.** Clustering outcomes observed employing the seed dataset.

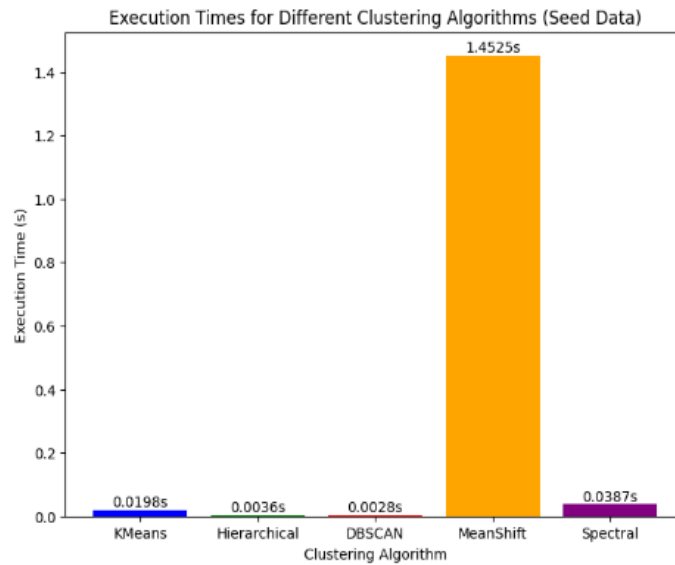
Fig 8 (a) applied k mean clustering and find four number of clusters, (b) hierarchical clustering algorithm give three number of clusters, (c) DBSCAN six number of clusters (d) mean shift algorithm give two number of clusters and (e) spectral clustering distributed the data in three similar group.

#### 4.2.1 Efficiency and accuracy of clustering algorithm by using Seed dataset.

Unlike other previous experiments, in this evaluation we raised the observation number in the seed dataset. We used a larger data set to investigate if the algorithmically performance depends on the data set's scale. The figure below describes the precision as well as runtime of every clustering algorithm used on the enlarged set of seeds. By comparing results, we can judge those algorithms' performance with the increased data volume and reveal their accuracy and efficiency in this sense.



(a)



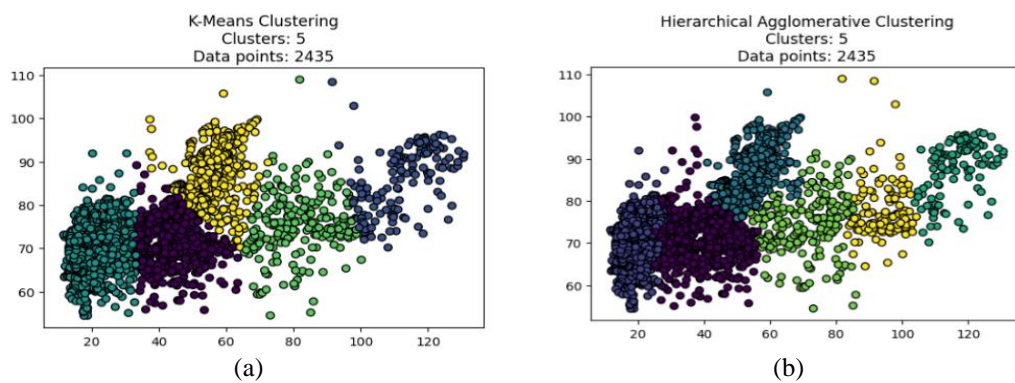
(b)

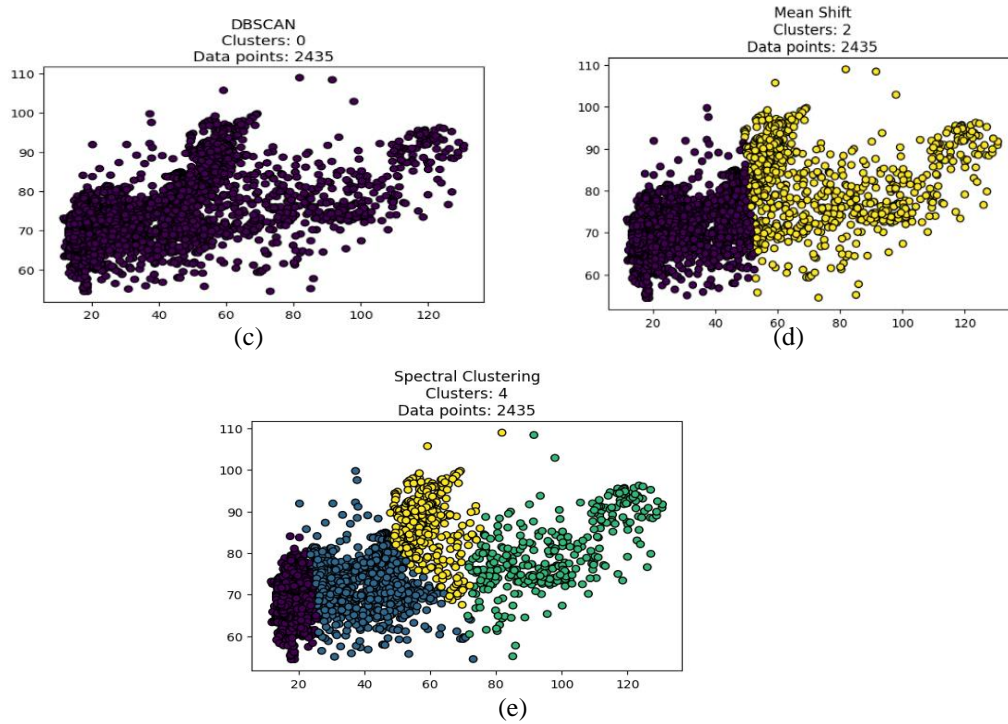
**Fig 9.** Accuracy and Execution time using the seed dataset.

Figure 9 (a) accuracy level of different algorithm (b) shows the execution times. The plot shows mean shift gives accurate results as compared to other methods, but its execution time is very high taking 1.45. k means clustering algorithm performed better results its accuracy is 91.06% also execution time is very less 0.01. as compared to the other methods k mean performed better results in terms of accuracy and execution times.

#### 4.3 Well log data clustering

In the realm of machine learning, various unsupervised algorithms serve the purpose of clustering, facilitating the grouping of data points based on similarities or patterns without labeled information. To explore the efficacy of these methods, we employed them across diverse datasets. Specifically, we utilized log data well to evaluate the performance of different clustering algorithms. This evaluation sought to assess how well these algorithms could delineate and organize the well log data, revealing inherent structures or relationships within the dataset. By applying a range of clustering methods to this specific dataset, we aimed to gauge their effectiveness in capturing distinct patterns or clusters present in the well log data.



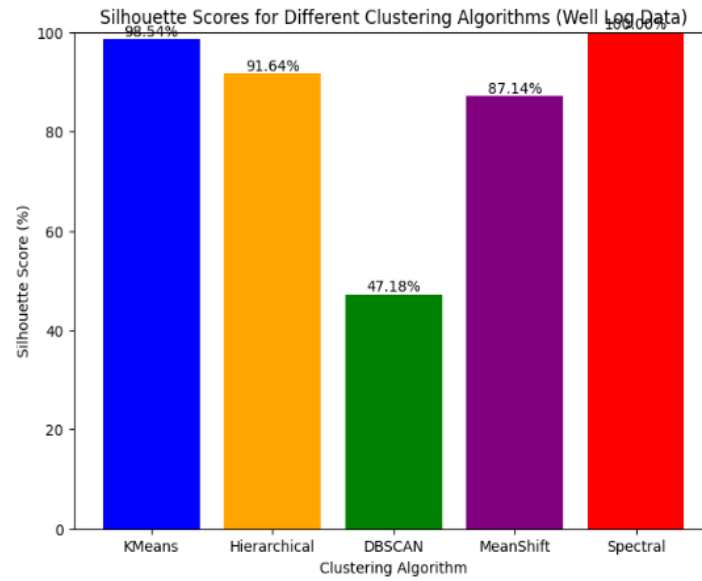


**Fig 10.** Clustering by using well log dataset.

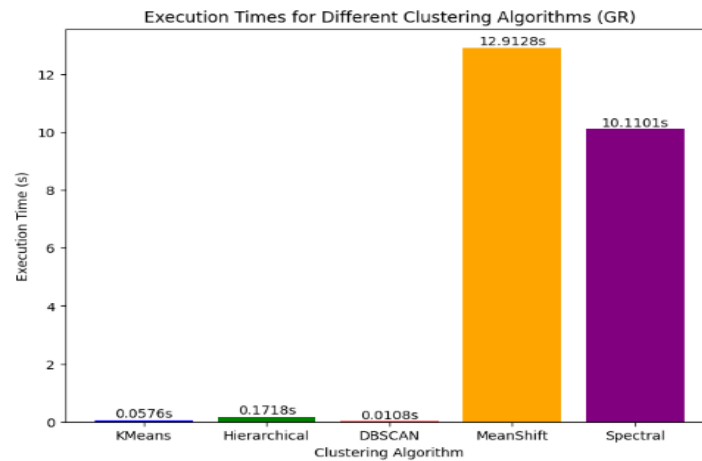
Fig10, in clusters (a - e) were generated utilizing various clustering algorithms. Specifically, (a, b, and e) employed four clusters, while DBSCAN utilized eleven clusters, and the Mean Shift method utilized two clusters. This analysis involved the

#### 4.3.1 Accuracy and time execution.

This stage involved conducting an analysis of well log data including 2435 observations. The purpose of this study was to measure how well the algorithms could deliver accurate results while also being efficient computationally. However, this evaluation was centered on scoring using silhouettes which is a measure of cluster compactness and separation. We wanted to find out how effective they were in grouping the Gamma Ray data into clusters with respect to their execution time. Through this analysis, the researchers wanted to grasp how well the clustering techniques were able to deal with a sample of 2435 observations, considering both accuracy and computation requirements.



(a)



(b)

**Fig 11.** Accuracy and Execution time.

Fig 11 (a) shows the level of accuracy of different clustering algorithms, (b) execution times of clustering algorithm. In (a) k mean clustering performed accurate results as compared to the other method and (b) k mean time execution is less as compared to the other methods. Spectral cluster show less accurate results and its execution time also more 10.100s

Clustering is a crucial part of unsupervised machine learning, especially when dealing with diverse data types. Its goal is to group similar data together, setting the stage for deeper analysis. With a variety of clustering algorithms available, the challenge lies in picking the best method for different datasets. Surprisingly, there hasn't been much comparative study to check accuracy and execution time across these algorithms. This paper dives into a comparative analysis of various clustering methods, including K-means, Hierarchical Agglomerative Clustering, DBSCAN, Mean Shift, Gaussian Mix, and Spectral Clustering. The aim is to assess their performance by examining both accuracy and execution time using diverse datasets. The findings reveal interesting insights. K-means, for instance, delivers accurate results and is efficient when handling large datasets. Hierarchical clustering automatically determines the number of clusters but falls short in accuracy. DBSCAN isn't ideal for larger datasets. Meanwhile, Mean Shift and Spectral Clustering perform well with small datasets but struggle with larger ones, taking more time compared to other algorithms.

#### 4.4 comparative study of all the methods using clustering.

Table 2 presents clustering results for three datasets (Iris, Seed, and Well Log) using different algorithms, including K-means, Hierarchical K-means, DBSCAN, Mean Shift, and Spectral clustering. The accuracy percentages and execution times are provided for each algorithm.

### 1. Iris Dataset:

K-means and Hierarchical K-means achieved similar accuracy at around 80%, with K-means slightly higher. DBSCAN had a lower accuracy of 70.87%. Mean Shift and Spectral clustering performed well, with Mean Shift achieving 100% accuracy. Notably, K-means maintained stable and consistent accuracy as the sample size increased. Time execution for K-means: 0.61s

### 2. Seed Dataset:

K-means demonstrated the highest accuracy at 91.06%, outperforming other algorithms. Hierarchical K-means also performed well with an accuracy of 86.70%. DBSCAN had a lower accuracy of 63.75%, while Mean Shift achieved perfect accuracy (100%). Remarkably, K-means consistently maintained high accuracy even with an increasing sample size. Time execution for K-means: 0.01s

### 3. Well Log Dataset:

K-means exhibited the highest accuracy at 98.54%, followed by Hierarchical K-means at 91.64%. DBSCAN faced challenges with a lower accuracy of 47.18%. Mean Shift and Spectral clustering showed reasonable accuracy. Notably, K-means accuracy remained consistently high with an increasing sample size. Time execution for K-means: 0.057s

Including the time execution information provides insights into the computational efficiency of each clustering algorithm, with K-means showing relatively low execution times in all three datasets.

Including the information about K-means maintaining or improving accuracy with an increasing sample size adds an important dimension to the interpretation of the results, suggesting the robustness and scalability of K-means clustering in these datasets.

**Table: 2** Summary of all the results

Dataset		K-mean	Hierarchical mean	k	DBSCAN	Mean Shift	Spectral	Total observation
Iris	Accuracy%	80.61	80.83		70.87	80.81	100	150
	Time execution/s	0.61	0.21		0.01	27.27	0.96	
Seed	Accuracy %	91.06	86.70		63.75	100	86.51	210
	Time Execution/s	0.01	0.003		0.0028	1.45	0.038	
Well Log	Accuracy %	98.54	91.64		47.18	87.14	100	2435
	Time Execution/s	0.057	0.171		0.010	12.91	10.11	

## 3. Conclusion

Clustering is very important for unsupervised ML when working with large data sets as well as intricate/detailed feature spaces. The development of clustering ensemble shows promising avenues towards robust clustering and dealing with data noise. Unsupervised machine learning has five popular clustering algorithms having individual limitations. Other algorithms might not work well with dirty data sets, whereas others will not do well with big data sets. Specific problems in each clustering algorithm are discussed by various researchers. Evaluations of these algorithms typically focus on two main aspects: accuracy and execution time. Despite the critical role clustering algorithms play, there's a notable gap in research: comparative study involving all cluster-based measures considering both precision and speed. Consequently, there is a need to fill the gap in the research by examining how different sized datasets affect the algorithms' performance systems. The findings suggest that the K-means clustering algorithm is well-suited for handling larger datasets. Notably, as the volume of observations grows, K-means demonstrates an intriguing trend of improving both in accuracy and processing speed. However, when faced with substantial noise within datasets, mean shift and spectral clustering algorithms outperform K-means in terms of accuracy. It's worth noting, though, that these alternatives exhibit longer execution times compared to K-means.

**Funding:** A YUTP-FRG Grant provided the funding for this project, operating under the cost center: 015LC0-432, at Universiti Teknologi PETRONAS, Malaysia.

**Acknowledgments:** The authors extend their appreciation to the unnamed reviewers and the editor for their thorough assessment of this paper, as well as for their invaluable recommendations and insights provided in their comments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Li, W., et al., An ensemble clustering framework based on hierarchical clustering ensemble selection and clusters clustering. *Cybernetics and Systems*, 2023. 54(5): p. 741-766.
2. Li, H., et al., LSEC: Large-scale spectral ensemble clustering. *Intelligent Data Analysis*, 2023. 27(1): p. 59-77.
3. Shanmugam, G., et al., Student Psychology based optimized routing algorithm for big data clustering in IoT with MapReduce framework. *Journal of Intelligent & Fuzzy Systems*, 2023(Preprint): p. 1-13.
4. Li, Y., et al., ZINBMM: a general mixture model for simultaneous clustering and gene selection using single-cell transcriptomic data. *Genome Biology*, 2023. 24(1): p. 208.
5. Singh, S. and K. Singh, Novel fuzzy similarity measures and their applications in pattern recognition and clustering analysis. *Granular Computing*, 2023: p. 1-23.
6. Flores, M.A., et al., Thermographic image processing analysis in a solar concentrator with hard C-means clustering. *Energy Reports*, 2023. 9: p. 312-321.
7. Kiran, A., et al., Enhancing Data Security in IoT Networks with Blockchain-Based Management and Adaptive Clustering Techniques. *Mathematics*, 2023. 11(9): p. 2073.
8. Wiroonsri, N., Clustering performance analysis using a new correlation-based cluster validity index. *Pattern Recognition*, 2024. 145: p. 109910.
9. Ahmadijad, N., Y. Chung, and L. Liu, J-Score: a robust measure of clustering accuracy. *PeerJ Computer Science*, 2023. 9: p. e1545.
10. Li, Q., et al., How to improve the accuracy of clustering algorithms. *Information Sciences*, 2023. 627: p. 52-70.
11. Kodinariya, T.M. and P.R. Makwana, Review on determining number of Cluster in K-Means Clustering. *International Journal*, 2013. 1(6): p. 90-95.
12. Gholizadeh, N., H. Saadatfar, and N. Hanafi, K-DBSCAN: An improved DBSCAN algorithm for big data. *The Journal of Supercomputing*, 2021. 77: p. 6214-6235.
13. Monath, N., et al. Scalable hierarchical agglomerative clustering. in *Proceedings of the 27th ACM SIGKDD Conference on knowledge discovery & data mining*. 2021.
14. Demirović, D., An implementation of the mean shift algorithm. *Image Processing On Line*, 2019. 9: p. 251-268.
15. Song, X., et al., A spectral clustering algorithm based on attribute fluctuation and density peaks clustering algorithm. *Applied Intelligence*, 2023. 53(9): p. 10520-10534.
16. Löster, T., Determining the optimal number of clusters in cluster analysis. *Proceedings of the 10th international days of statistics and economics*, 2016: p. 8-10.
17. Li, M., E. Frank, and B. Pfahringer, Large scale K-means clustering using GPUs. *Data Mining and Knowledge Discovery*, 2023. 37(1): p. 67-109.
18. Liu, J., F. Cao, and J. Liang, Centroids-guided deep multi-view k-means clustering. *Information Sciences*, 2022. 609: p. 876-896.
19. Brown, P.O., et al. Mahalanobis distance based k-means clustering. in *International Conference on Big Data Analytics and Knowledge Discovery*. 2022. Springer.
20. De Rosa, A. and A. Khajavirad, The ratio-cut polytope and K-means clustering. *SIAM Journal on Optimization*, 2022. 32(1): p. 173-203.
21. Pinheiro, W.A. and A.B.S. Pinheiro, Hierarchical++: improving the hierarchical clustering algorithm. *International Journal of Data Mining, Modelling and Management*, 2023. 15(3): p. 223-239.
22. <lee 2022.pdf>.
23. Yu, H. and X. Hou, Hierarchical clustering in astronomy. *Astronomy and Computing*, 2022: p. 100662.
24. Vichi, M., C. Cavicchia, and P.J. Groenen, Hierarchical means clustering. *Journal of Classification*, 2022. 39(3): p. 553-577.
25. Koren, O., A. Shamalov, and N. Perel, Small Files Problem Resolution via Hierarchical Clustering Algorithm. *Big Data*, 2023.
26. Wu, G., et al., HY-DBSCAN: A hybrid parallel DBSCAN clustering algorithm scalable on distributed-memory computers. *Journal of Parallel and Distributed Computing*, 2022. 168: p. 57-69.
27. Hanafi, N. and H. Saadatfar, A fast DBSCAN algorithm for big data based on efficient density calculation. *Expert Systems with Applications*, 2022. 203: p. 117501.
28. An, X., et al., STRP-DBSCAN: A Parallel DBSCAN Algorithm Based on Spatial-Temporal Random Partitioning for Clustering Trajectory Data. *Applied Sciences*, 2023. 13(20): p. 11122.
29. Jain, P.K., M.S. Bajpai, and R. Pamula, A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality. *Int. Arab J. Inf. Technol.*, 2022. 19(1): p. 23-28.
30. Cariou, C., S. Le Moan, and K. Chehdi, A novel mean-shift algorithm for data clustering. *IEEE Access*, 2022. 10: p. 14575-14585.
31. Chen, J., et al., Robust Truth Discovery Scheme Based on Mean Shift Clustering Algorithm. *Journal of Internet Technology*, 2021. 22(4): p. 835-842.

- 
32. Belloum, F., L. Houichi, and M. Kherouf, The Performance of Spectral Clustering Algorithms on Water Distribution Networks: Further Evidence. *Engineering, Technology & Applied Science Research*, 2022. 12(4): p. 9056-9062.
  33. Cui, Y., et al. A Spectral Clustering Algorithm Based on Differential Privacy Preservation. in *International Conference on Algorithms and Architectures for Parallel Processing*. 2021. Springer.
  34. Ikotun, A.M., et al., K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 2023. 622: p. 178-210.
  35. Murtagh, F. and P. Contreras, Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2012. 2(1): p. 86-97.
  36. Dogan, A. and D. Birant, K-centroid link: a novel hierarchical clustering linkage method. *Applied Intelligence*, 2022: p. 1-24.
  37. Ozertem, U., D. Erdogmus, and R. Jenssen, Mean shift spectral clustering. *Pattern Recognition*, 2008. 41(6): p. 1924-1938.
  38. Gou, S., X. Zhuang, and L. Jiao, Quantum immune fast spectral clustering for SAR image segmentation. *IEEE Geoscience and Remote Sensing Letters*, 2011. 9(1): p. 8-12.
-