AIMS Mathematics, 9(9): 25070–25097.
DOI: 10.3934/math.20241222
Received: 14 June 2024
Revised: 17 July 2024
Accepted: 23 July 2024
Published: 27 August 2024

https://www.aimspress.com/journal/Math

Research article

Addressing limitations of the K-means clustering algorithm: outliers, non-spherical data, and optimal cluster selection

Iliyas Karim khan[1,*], Hanita Binti Daud[1], Nooraini binti Zainuddin[1], Rajalingam Sokkalingam[1], Abdussamad[1], Abdul Museeb[1] and Agha Inayat[2]

[1] Fundamental and Applied Science Department, Universiti Teknologi PETRONAS, Perak 32610, Malaysia
[2] Department of Statistic University of Malakand Chakdara Lower Dir, Khyber Pakhtunkhwa Pakistan

* **Correspondence:** Email: iliyas_22008363@utp.edu.my.

Abstract: Clustering is essential in data analysis, with K-means clustering being widely used for its simplicity and efficiency. However, several challenges can affect its performance, including the handling of outliers, the transformation of non-spherical data into a spherical form, and the selection of the optimal number of clusters. This paper addressed these challenges by developing and enhancing specific models. The primary objective was to improve the robustness and accuracy of K-means clustering in the presence of these issues. To handle outliers, this research employed the winsorization method, which uses threshold values to minimize the influence of extreme data points. For the transformation of non-spherical data into a spherical form, the KROMD method was introduced, which combines Manhattan distance with a Gaussian kernel. This approach ensured a more accurate representation of the data, facilitating better clustering performance. The third objective focused on enhancing the gap statistic for selecting the optimal number of clusters. This was achieved by standardizing the expected value of reference data using an exponential distribution, providing a more reliable criterion for determining the appropriate number of clusters. Experimental results demonstrated that the winsorization method effectively handles outliers, leading to improved clustering stability. The KROMD method significantly enhanced the accuracy of converting non-spherical data into spherical form, achieving an accuracy level of 0.83 percent and an execution time of 0.14 per second. Furthermore, the enhanced gap statistic method outperformed other techniques in selecting the optimal number of clusters, achieving an accuracy of 93.35 percent and an execution time of 0.1433 per second. These advancements collectively enhance the performance of K-means

clustering, making it more robust and effective for complex data analysis tasks.

Keywords: K-mean; non-spherical; spherical; KROMD; gap statistic
Mathematics Subject Classification: 68T10, 91C20

1. Introduction

Clustering is a cornerstone of data analysis, essential across various domains including data mining [1], pattern recognition [2], information retrieval [3] and engineering optimization [4,5]. Among the plethora of clustering algorithms, K-means is renowned for its simplicity [6], computational efficiency, and scalability [7]. It divides a dataset into a predetermined number of clusters [8,9], each anchored by its centroid, which encapsulates the average of the data points allocated to it. This method is widely embraced due to its straightforward implementation and ability to handle large datasets efficiently. In this research paper, we delve into the realm of clustering, focusing particularly on addressing key challenges faced by the K-means algorithm. Our aim is to enhance the robustness and applicability of K-means in practical scenarios. We identify three primary challenges that affect its performance and propose comprehensive solutions to overcome them [10,11]. The first challenge revolves around the sensitivity of K-means to outliers within the dataset. Outliers, or data points significantly deviating from the majority, can distort centroid computation and undermine clustering accuracy. To mitigate this, we employ outlier detection techniques and advocate for the application of the winsorization technique. This method replaces data values below the lower threshold with the value at the lower threshold and values above the upper threshold with the value at the upper threshold, thus mitigating the impact of outliers without data loss [12,13]. The second challenge emerges when K-means encounters datasets with clusters exhibiting non-spherical shapes. Traditional K-means operate under the assumption of spherical and isotropic clusters, which may not hold true for datasets containing elongated, irregular, or overlapping clusters. To address this, we employ the KROMD method, which combines the rank order distance (ROD) technique with Gaussian kernels to transform non-spherical data into a more suitable representation for K-means clustering. This transformation enhances the algorithm's ability to accurately capture the underlying structure of the data [14,15]. The third challenge pertains to determining the optimal number of clusters (k) for K-means clustering. Selecting an appropriate value for k is pivotal in obtaining meaningful and interpretable clustering results. However, this task is often fraught with challenges, relying heavily on domain knowledge or heuristic methods prone to uncertainty and computational inefficiency. We propose an enhanced approach based on the gap statistic, incorporating an exponential distribution to automatically determine the optimal number of clusters. This approach provides a more accurate and effective means of selecting the optimal number of clusters compared to traditional methods [16–19].

The following sections are organized as follows: Section 2 reviews the relevant literature. Section 3 describes our methodology, including outlier detection and handling outliers by using the winsorization method (Section 3.1), the transformation of non-spherical data (Section 3.2), and the enhanced gap statistic for optimal clustering (Section 3.3). Section 4 presents the experimental setup and descriptive statistics of the dataset. Section 5 discusses the results, covering outlier mitigation (Section 5.1), data transformation accuracy (Section 5.2), optimal cluster selection (Section 5.3), and research contributions (Section 5.4). Finally, Section 6 provides the conclusion of the research.

2. Related work

This section of the paper discusses various literature reviews and related methods for outlier detection, transforming non-spherical data into spherical form, and selecting the optimal number of clusters in K-means.

2.1. Impact of outliers in K-means

The literature review emphasized the significance of outlier detection, particularly in contexts like fraud and fault detection. Resolving issues with the density peak clustering algorithm involves overcoming manual parameter setting and high time complexity. The proposed solution involves substituting density peaks with k-nearest neighbors clustering and automating the selection of clustering centers based on density and distance [20]. Outlier detection in batch and streaming data played a crucial role in data mining, but existing algorithms had shortcomings. For batch data, accuracy suffered due to limited data point labeling from histogram-based feature vectors. Streaming data algorithms were hindered by sensitivity to data distance and lengthy parameter tuning. To overcome these challenges, the authors introduced PDC (Probability Density-based Clustering), which leveraged probability density for lightweight outlier detection, ensuring accuracy and insensitivity to data distance [21,22]. Seismic clustering serves as a vital technique in seismology for identifying patterns in seismic events and offering insights into geological processes. However, its application to ongoing landslide-induced signals and the impact of outliers have not received much research attention. The paper presented a novel consensus clustering strategy with outlier removal for landslide-induced seismic signals. The proposed approach incorporated a parameter setting method to improve clustering accuracy and robustness. Experimental results demonstrated that the proposed approach outperformed state-of-the-art clustering methods [23].

2.2. Impact of non-spherical data in K-means: K-nearest neighbors (KNN)

The data is transformed to a more spherical shape using a whitening transformation or principal component analysis (PCA) followed by normalization. This standardizes the data to have a zero mean and unit variance, removes correlations with PCA, and scales for uniform variance the dataset with

$$Z_i = {}^{x_i - \mu}/_{\delta},$$

where x_I is the original data point, μ is the mean, and σ is the standard deviation. Applying PCA and normalization as $w_i = \frac{PCA\,(Z_i)}{\sqrt{\lambda}}$, where PCA(Zi) is the principal component transformation and λ is its eigenvalue. Transforming non-spherical data into a spherical form enhances algorithms like K-nearest neighbors (KNN) by improving distance measurements and overall accuracy across application [24].

Spectral clustering: The process begins with constructing an n×n affinity matrix A, where each element A_{ij} quantifies the similarity between data points *si* and *sj* using a Gaussian kernel:

$$A_{ij} = \exp\left(\frac{-\| s_i - s_j \|^2}{2\delta^2}\right)$$

ensuring self-similarity, A_{ij} is zero. Next, a diagonal matrix D is defined, with D representing the sum of similarities in the iii-*th* row of A. The normalized Laplacian matrix $L = D^{1/2}AD^{-1/2}$ is then computed to emphasize relationships between data points [14]. Eigenvectors $x_1, x_2, x_3, \ldots x_n$ of L are found orthogonally and arranged into matrix X. Each row Xi of X is normalized so that $Y_{ij} = \frac{X_{ij}}{\sqrt{\sum_j - X_{ij}^-}}$.

In reduced dimensionality R^k, each row of matrix Y is treated as a point and clustered using algorithms like K-means to minimize intra-cluster and maximize inter-cluster distances.

Principal component analysis (PCA): PCA was a method used to reduce the dimensionality of data while maintaining its essential structure. This was achieved by transforming the data into a new coordinate system defined by its principal components [25].

Mean calculation: Calculate the mean vector μ of the data points: $\mu = \frac{1}{2}\sum_{i=1}^{n} x_i$.

Covariance matrix: Compute the covariance matrix Σ to understand the relationships between the dimensions of the data: $\sum = \frac{1}{n}\sum_{i=1}^{n}(x_I - \mu)(x_I - \mu)^T$.

Eigen decomposition: Perform eigen decomposition on Σ to obtain its eigenvalues and eigenvectors: $\sum = QAQ^T$, where Q is a matrix containing orthogonal eigenvectors and Λ is a diagonal matrix of eigenvalues.

Select principal components: Choose the top k eigenvectors associated with the largest eigenvalues to form the projection matrix W.

Data projection: Project the original data $x_1, x_2, x_3, \ldots x_n$ onto the principal components W. $Z_i = W^T(x_i - \mu)$, where Z_I represents the data point in the reduced-dimensional space. PCA is extensively used in various fields for dimensionality reduction and data preprocessing. It converts non-spherical data into a spherical form, making it suitable for clustering algorithms that assume spherical clusters, such as K-means.

Rank order distance: Euclidean distance is a commonly used similarity measure in clustering, with a long history of application. It is calculated between two samples, a and b, as the square root of the sum of the squared differences between their respective features. The formula for the distance is: $d(a,b) = \sqrt{\sum_{i=1}(a_i - b_i)^2}$ where a_i is i-*th* feature of a and b_i is ith feature of b. Traditional clustering methods, like K-means and single-link, rely on this metric. However, when dealing with non-spherical data characterized by high-level noise, the effectiveness of mining data clusters is limited using the Euclidean distance. Consequently, researchers have explored alternative similarity measures, such as the Gaussian kernel and rank order distance (ROD), to address these challenges [26]. Unlike the Euclidean distance, they involve a structural alignment process on samples, enhancing the ability to uncover true data structures for clustering purposes. In this context, a_i and b_i are the iii-*th* features of samples a and b. Traditional clustering methods like K-means use Euclidean distance but struggle with non-spherical, noisy data. Researchers have explored alternatives like the Gaussian kernel and ROD, which better reveal true data structures. Rank order distance from a to b is calculated by: $R(a,b) = \sum_{k=0}^{O_a(b)} O_b\big(fa(k)\big)$.

In this context, fa(k) denotes the element ranked k-th in a distance list, while Oa(b) represents b rank in a list. For a rank order distance (ROD) between a and b, $R^- = \frac{R_{(a,b)}+R_{(b,a)}}{\min(O_a(b),O_b(a))}$. In scenarios involving non-spherical data and noise challenges, ROD excels over Euclidean distance in assessing sample similarities by incorporating neighborhood structures.

Clustering, especially K-means, has been widely used in signal processing for its efficiency and simplicity. However, K-means clustering struggles with non-spherical clusters. To address this, the self-weighted Euler K-means (SWEKM) model was proposed, integrating clustering and feature selection while using a Euler kernel to manage noisy points and outliers. Experiments on UCI datasets demonstrated that SWEKM outperformed state-of-the-art kernel K-means in handling non-spherical clusters in signal-processing tasks [27]. The resting dynamics of non-spherical particles were studied using a sharp interface–immersed boundary method and a kinematic-based collision model. Simulations showed that hydrodynamic moments, influenced by Reynolds number (Re), affected angular velocities but not trajectories. Using the shape factor K-n, the best scaling was achieved with the projected area of non-spherical particles. A linear relationship between mean K-n and Re was found, highlighting the effectiveness of particle-resolved simulations for modeling non-spherical particles [28].

2.3. Optimal number of clusters in K-means: Davies-Bouldin index (DBI)

This index helps to determine the optimal number of clusters in a dataset by evaluating the similarity of each point to every cluster. It considers both the dispersion and dissimilarity within clusters. The index aims to find clusters that are compact and well-separated. The optimal number of clusters, identified by the minimum value of the index, represents the configuration where clusters are maximally distinct and internally cohesive [29].

$$\mathrm{DB_C} = \frac{1}{\mathrm{c}}\sum_{\mathrm{i}=1}^{\mathrm{c}} \max_{\mathrm{j}=1\ldots..\mathrm{c}\ \ \mathrm{i}\neq\mathrm{j}} \left\{\frac{\mathrm{dia}\ (\mathrm{c_i}) + \mathrm{dia_{cj}}}{\left\|\mathrm{c_i} - \mathrm{c_j}\right\|}\right\}, \tag{1}$$

where c represents the total number of clusters, n_i represents the number of points, and c_i is the centroid of cluster c_i. This index measures the minimum distance within each cluster.

Calinski-Harabasz index: This index calculates the average sum of squared distances between clusters (inter-cluster) and within clusters (intra-cluster). It provides a faster approach for determining the optimal number of clusters compared with other indices. The index aims to maximize the dispersion between clusters while minimizing it within clusters. The optimal number of clusters (ONC) is represented by the maximum value of this index, indicating that the clusters are both compact and well-separated [30].

$$\mathrm{CH(c)} = \frac{\mathrm{traceB_m}}{\mathrm{traceW_m}} * \frac{\mathrm{N} - \mathrm{C}}{\mathrm{C} - 1}. \tag{2}$$

In the above Eq (2) $\mathrm{B_m}$ denotes the between-cluster scatter matrix, $\mathrm{W_m}$ denotes the internal scatter matrix, N is the total number of clustered samples, and c indicates the number of clusters. Where $\mathrm{W_m} = \sum_{\mathrm{i}=1}^{\mathrm{c}} \sum_{\mathrm{x}\varepsilon\mathrm{c_i}} (\mathrm{x} - \mathrm{c_i})(\mathrm{x} - \mathrm{c_i})^{\mathrm{T}}$ and $\mathrm{B_m} = \sum_{\mathrm{i}} \mathrm{n_i} (\mathrm{c_i} - \mathrm{k})\,(\mathrm{c_i} - \mathrm{k})^{\mathrm{T}}$. $\mathrm{C_I}$ are the points that comprise cluster $\mathrm{C_I}$, $\mathrm{n_I}$ represents the number of points in cluster $\mathrm{C_I}$, and k is the center of the entire dataset.

Silhouette score: The silhouette method evaluates clustering performance by considering two key factors: cohesion and separation. Cohesion measures how similar an object is to its own cluster, while separation assesses how distinct a cluster is from the others. This evaluation is quantified using the silhouette score, which ranges from -1 to 1. A score close to 1 indicates a strong association between an object and its cluster, suggesting effective clustering, while a low score indicates poor clustering quality [17]. A high average silhouette score across a dataset suggests that the clustering model is both appropriate and reliable. The silhouette score is calculated as follows:

$$S(i) = \frac{b(i) - a(i)}{max\big(a(i),\ b(i)\big)}. \tag{3}$$

In Eq (3), $S(i)$ is the silhouette score for the i-*th* data point, indicating its clustering quality. $A(i)$ is the average distance from the i-*th* data point to other points within the same cluster. $B(i)$ is the smallest average distance from the i-*th* data point to points in any other cluster.

Elbow method: The Elbow method determines the optimal number of clusters (K) by calculating the squared Euclidean distances between data points and their cluster centroids, producing a series of KKK values. The sum of squared errors (SSE) measures clustering performance, with lower SSE indicating tighter clustering. As K increases, SSE decreases sharply until reaching an "elbow" point, which suggests the optimal cluster number [31]. However, this point can be subjective, and adding more

$$SSE(x)=\sum\sum\|yi-di\|2 \tag{4}$$

clusters beyond this point does not significantly improve clustering performance.

In Eq (4), after reaching the true cluster count, the SSE still decreases, but the rate of reduction slows significantly, indicating diminishing returns from adding more clusters.

Gap statistic analysis: The gap statistic, developed by Tibshirani, determines the optimal number of clusters in datasets with unknown classifications. It uses Monte Carlo sampling to create reference distributions, which benchmark the sum of squared Euclidean distances within clusters [32,33]. By comparing these results to a zero-mean reference distribution, the optimal number of clusters is identified. The calculation is as follows:

$$\text{Gap(k)}=E[log(wk)]-log(wk). \tag{5}$$

In the above Eq (5), $E[log(wk)]$ represents the expected value of the logarithm of the within-cluster dispersion wk for k clusters. The within-cluster dispersion wk typically measures how compact the clusters are, often quantified by the sum of squared distances of points within each cluster to their centroid. $log(wk)$ is the logarithm of the actual within-cluster dispersion wk for k clusters. To use the gap statistic in practice, wk is calculated for a range of k values, $E[log(wk)]$ is estimated (often using a Monte Carlo simulation approach), and then Gap(k) is computed for each k. The k value where Gap(k) is maximized or shows a clear peak should be chosen as the optimal number of clusters for the dataset.

This comprehensive review, summarized in Table 1, examines the prevalent methods used in the K-means algorithm, highlighting the inherent limitations of each method and their suitability for specific dataset types. It is evident that traditional methodologies often fail when applied to different datasets.

Table 1. Comparative analysis of methods for determining limitations in K-means clustering.

Method/Reference	Limitations
Outliers' detection in K-means clustering algorithm	
Winsorizing to handle outliers [34]	Outliers and unbalanced data complexity. Variability in human development index cases.
Entropic outlier sparsification (EOS). Mixture of spherically symmetric Gaussians [35]	Outliers limit dataset usefulness in real-life scenarios. Challenges in finding a general method for different datasets.
Self-adaptive mixture similarity function based on geometric distance and S divergence [36]	Geometric distance-based similarity function struggles with massively overlapped data. Divergence-based similarity function fails to distinguish disjointed uncertain data.
Non-spherical data in K-means clustering algorithm	
Original dataset clustered into high-density sub-clusters [14]	K-means less effective for clustering non-spherical data. Connectivity among sub-clusters evaluated by density and nearest distance class.
Incorporate Elkan and Hamerly accelerations. Work directly with cosines instead of Euclidean distances [37]	Acceleration techniques for Euclidean distances do not easily translate. Spherical K-means uses cosine similarities for computational efficiency.
K-means spherical clustering [38]	Limited to earthquake events in Bengkulu Province and surroundings. Analysis based on data from 1970 to 2019.
Machine learning [39]	Acceleration techniques for Euclidean distances do not easily translate.
Selection of optimal number of clusters in K-means algorithm	
AutoML procedure that combines numeric and categorical dataset [40]	Selecting optimal K in K-means algorithm is a challenge. AutoML procedure combines numeric and categorical datasets for analysis.
Elbow method, gap statistics method, and Silhouette method, agglomerative hierarchical clustering (AHC) algorithm, and K-means method [41]	The Elbow method may not always give clear optimal number clusters. Gap statistics method can be computationally expensive for large datasets.
Optimal cluster number estimation algorithm (OCNE) [42]	No need to specify maximum or range of k values. Does not require knee point detection in the graph.
Elbow method, Silhouette method, gap statistic method, variance difference method [43]	Energy efficiency, coverage difficulties, and network lifespan challenges. Accuracy diminishes with increasing clusters until it reaches zero.
Euclidean distance, Manhattan distance, Chebychev distance, Minkowski distance, and estimated gap [44]	Initial selection of K is a significant concern. Existing algorithms require scalable solutions for large datasets.
Davies Bouldin index for determining optimal number of clusters [45]	K-means clustering may not be optimal for large datasets.
Davies-Bouldin index, Calinski-Harabasz index, and silhouette plot [30]	KH algorithm prone to local optima, weak searchability. K-means affected by initial clustering center selection.

3. Methodology

This study addresses three primary challenges associated with the K-means clustering algorithm: the detection and handling of outliers, the transformation of non-spherical data into spherical form, and the selection of the optimal number of clusters. The methodologies developed to tackle these challenges are detailed as follows:

Detection and handling of outliers: One of the major limitations of the K-means algorithm is its sensitivity to outliers. To address this issue, this paper employs the winsorization method, as discussed in Section 3.1.

Conversion of non-spherical data to spherical form using KROMD method: K-means performs poorly in the presence of non-spherical data. To mitigate this problem, this paper introduces the KROMD method, which combines Manhattan distance with a Gaussian kernel. The details of this approach are explained in Section 3.2.

Selection of the optimal number of clusters by enhancing the gap statistic: Selecting the optimal number of clusters is a challenging task in K-means clustering. This paper enhances the gap statistic by standardizing expected reference data to overcome this limitation. The detailed methodology is provided in Section 3.3.

Figure 1(a) illustrates the process of detecting outliers and handling them by applying winsorization methods. Figure 1(b) depicts the process of converting non-spherical data into a spherical form using KROMD methods, which combine ROD and Gaussian kernel techniques. Figure 1(c) outlines the method for selecting the optimal number of clusters in K-means by enhancing the gap statistic method through the standardization of reference data. The methods calculate the range of values within cluster sum of square for both the original and reference data. In place of expected values, this algorithm applies standardization methods that clearly and effectively select the optimal number of clusters, mathematically discussed in Section 3.3.6.

Sequential breakdown of the flowchart in Figure 1:

1) Winsorization of outliers

Input: data, =dataset. Load-x=data [:2!]

Output: *Winsorization of outliers*

1: *Choose methods*
z-score or IQR

2: *Reviewer outliers*=[];

3: *Set thresholds values*=[];

4: *Set lower and upper bounds* do

5: $\boldsymbol{IQR} = Q_3 - Q_1$ *(I,Q.R!),*

6: *Lower bound*=$Q_1 - 1.5 * IQR$
Upper bound=$Q_3 + 1.5 * IQR$

7: *Transform each data point for each value of x apply the following transformation*

8: $X_{Winsorization}$

9: $\begin{cases} Lower\ Bound\ if\ X < lower\ Bound \\ Upper\ Bound\ if\ X > upper\ Bound \\ X \qquad\qquad\qquad\qquad Otherwise \end{cases}$

10: winsorization outliers.

2) Conversion to spherical data

Input: Irregular dataset containing high level of noise.

Output: Provide a cluster set 'C' and an "un-grouped" cluster 'Cun'.

1: Initialize clusters 'C' as {C1,C2,...,CN} of data according to their similarity basis of dataset. Ranking of dataset according to their ascending or descending order.

2**: Repeat the following steps:**

3: For all pairs (Cj, Ci) in 'C', do the following:

4: Calculate Ranking of dataset

5: Calculate rank order Manhattan distance

6: Identify ⟨Ci, Cj⟩ as a candidate merging pair.

7: applying Gaussian kernel

7**: End the conditional statement.**

8: The process stops until all the data is converted from non-spherical to spherical form.

3) Optimal number of clusters

Input: *data, =dataset.load-x=data[:,2!]*

Output: *K, (Number of Optimal Cluster K in K-Mean)*

1: *def Sample Num, P, MaxK,u, Sigma;*

2: *SampleSet=[];*

3: *Size(u)=[Um,];*

4: **for** *i=1: Um do*

5: *SampleSet= [sampleSet; munrud (u(I,!), Sigma, fix(SampleNum/Um))]*

6: *Wk= log(compwWk(SampleSet, Maxk));*

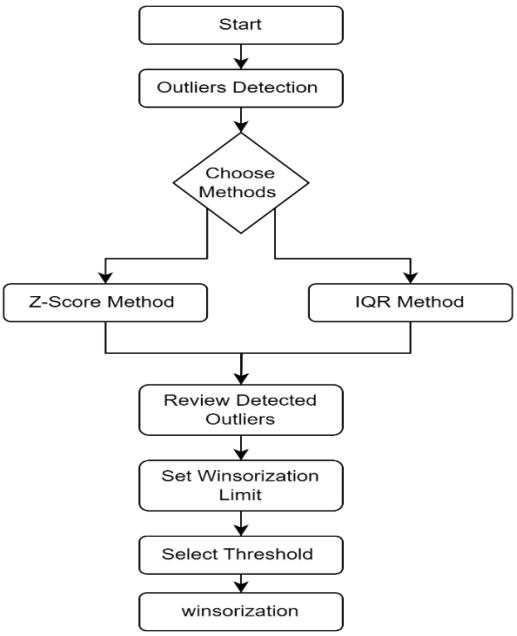
7: **for** *b=1 :P do*

8: *Wkb= log(CompuWk(RefSet(!, ! b) Maxk);*

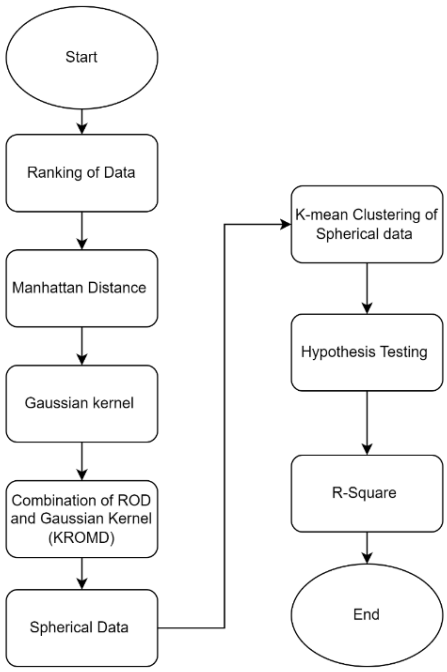
9: **for** *k=1: Maxk, OptimumUsk=1*

10: $EGSk=\frac{\log\left(W_{kb}^{*}\right)-(-\gamma-\log(\lambda))}{=\frac{\pi}{\sqrt{6}}+\sqrt{1+\frac{1}{B}}}$

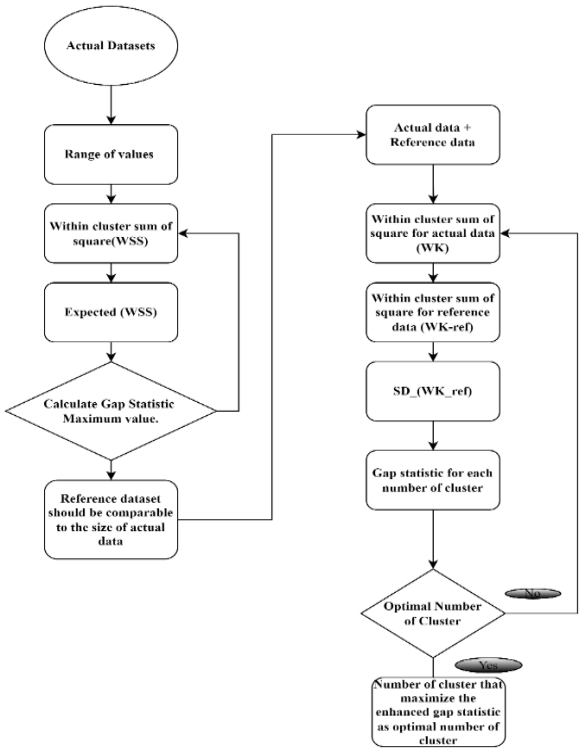
11: *EGSk optimal value for large k value.*



(a)



(b)



(c)

Figure 1. Flowchart for the K-means clustering algorithm.

3.1. Outlier detection and handling

Outliers are data points that differ significantly from most observations, arising from natural variability or data collection errors. It is essential to identify and manage outliers, as they can greatly impact analysis results, potentially causing misleading conclusions. Winsorization is a statistical technique that reduces the influence of extreme values by adjusting outliers to the nearest specified percentile values. This method helps to lessen the effect of potentially erroneous outliers [46]. Winsorization of outliers: Outliers are identified within the dataset using the interquartile range (IQR) method, which allows for the detection of data points lying outside the typical range. Subsequently, the winsorization technique is employed to handle these outliers, whereby extreme values are replaced with the nearest value within a specified percentile range, ensuring a more balanced and representative dataset for analysis [47].

$$\mathrm{IQR} = \mathrm{Q}_3 - \mathrm{Q}_1. \tag{6}$$

Where Q_1 is the first quartile (25th percentile) and Q_3 is the third quartile (75th percentile). Winsorization upper and lower bounds are defined as follows:

$$\text{Lower bound} = \mathrm{Q}_1 - 1.5 \times \mathrm{IQR},$$

$$\text{Upper bound} = \mathrm{Q}_3 + 1.5 \times \mathrm{IQR}.$$

Each data point is transformed for each value of X by applying the following transformation:

$$\mathrm{X}_{\mathrm{Winsorization}} = \begin{cases} \text{Lower Bound} & \text{if } \mathrm{X} < \text{lower Bound} \\ \text{Upper Bound} & \text{if } \mathrm{X} > \text{upper Bound}. \\ \mathrm{X} & \text{Otherwise} \end{cases} \tag{7}$$

In the above Eq (7), to mitigate the impact of outliers on the dataset, values below the lower bound are set to the lower bound and values above the upper bound are set to the upper bound, while values within the bounds remained unchanged. This approach aided in reducing outliers using the winsorization technique. Outliers were identified by calculating the interquartile range (IQR) to gauge the data spread. Data points beyond 1.5 times the IQR were flagged as outliers, and their values were adjusted to extreme values outside the normal range before reintroducing them into the dataset. This method significantly influenced the clustering process, enabling a robust evaluation of winsorization combined with the K-means method.

3.2. Conversion of non-spherical data to spherical form

In the presence of non-spherical data, the K-means algorithm often underperforms. To address this challenge, this paper proposes a novel approach called KROMD, which combines the ROD (rank order distance) technique with a Gaussian kernel method. The performance of KROMD is evaluated by comparing it with established methods such as KNN (K-nearest neighbors), spectral clustering, PCA, and ROD.

3.2.1. Kernelized rank order Manhattan distance (KROMD)

To effectively cluster non-spherical data with high noise levels, we introduce a novel similarity measure called Kernelized rank order Manhattan distance (KROMD). This measure integrates rank order distance (ROD) with a Gaussian kernel. Non-spherical data refers to clusters that deviate from spherical shapes, while high noise indicates numerous data points between clusters, causing them to overlap [14].

3.2.2. ROD for noise tolerance

Traditional ROD has limited capability in handling high noise levels. We enhance ROD by selectively considering only two distance ranks for each sample pair, thus reducing the influence of noise-related ranks [48]. The refined ROD between samples *a* and *b* is defined as:

$$R(a,b) = R_a(b) + R_b(a). \tag{8}$$

In the above Eq (8), rank order distances $R_a(b)$ and $R_b(a)$ quantify the dissimilarity between points a and b based on their ranks within the dataset. In this Eq (8), ROD demonstrates greater resilience to noise, enabling more effective structure detection in non-spherical, noisy data.

3.2.3. Gaussian kernel

The enhanced ROD demonstrates improved resilience to high noise levels, effectively capturing structures in noisy, non-spherical data. To reinforce cluster structures, KROMD integrates this improved ROD with a Gaussian kernel (Eq (3)), which efficiently brings samples within the same cluster closer together. This kernel is commonly used in clustering methods for non-spherical data. The Gaussian kernel between points a and b is computed as follows:

$$k(a,b) = e^{\frac{-d(a,b)^2}{u^2}}. \tag{9}$$

In the above Eq (9), the Gaussian kernel $k(a,b)$ measures the similarity between two points, a and b, based on their distance $d(a,b)$. $e^{\frac{-d(a,b)^2}{u^2}}$ ensures that closer points [small $d(a,b)$] receive higher similarity values, as $e^{\frac{-d(a,b)^2}{u^2}}$ approaches 1 for small distances. Conversely, points that are farther apart [large $d(a,b)$] receive lower similarity values, approaching 0. μ is a tunable parameter, often referred to as the bandwidth or scale parameter. It controls the width of the Gaussian kernel and significantly impacts the transformation of data. This μ in the Gaussian kernel is essential for controlling the extent of influence between data points, smoothing out noise, and converting non-spherical data into a more spherical form, thus enhancing the performance of clustering algorithms like K-means.

3.2.4. KROMD calculation

KROMD combines the ROD and Gaussian kernel to provide a robust similarity measure. The KROMD between samples a and b is calculated as:

$$\mathrm{KROMD} = \mathrm{R(ab)} \times \frac{1}{\mathrm{K(a,b)}} = \mathrm{R_a(b)} + \mathrm{R_b(a)} \times \frac{1}{e^{-\frac{d(a,b)^2}{u^2}}}. \tag{10}$$

In the above Eq (10), rank order distances $\mathrm{R_a(b)}$ and $\mathrm{R_b(a)}$ quantify the dissimilarity between points a and b based on their ranks within the dataset. Gaussian Kernel term $e^{-\frac{d(a,b)^2}{u^2}}$ modifies the contribution of $\mathrm{R_b(a)}$ based on the distance $d(a,b)$ between points a and b. It emphasizes similarity when $d(a,b)$ is small (points are close) and reduces similarity when $d(a,b)$ is large (points are far apart).

3.3. Optimal number of cluster selection

Determining the optimal number of clusters in K-means is challenging for researchers. Therefore, this paper enhanced the gap statistic by standardizing the expected value of the reference dataset using an exponential distribution.

The enhanced gap statistic (EGS) builds upon the traditional gap statistic [44,49] by incorporating an adjustment factor that considers the standard deviation of the within-cluster sum of squares from the reference dataset. This factor addresses variations within the reference dataset, resulting in a more accurate determination of the optimal number of clusters (ONC). By applying an exponential distribution to the standardization process, EGS enhances the robustness, efficiency, and accuracy of clustering results, particularly in the presence of outliers. In the above Eq. 5, we standardize $E[log(wk)]$ using an exponential distribution. The probability density function (PDF) of an exponential distribution is expressed as $f(w^*) = \lambda e^{-\lambda w^*} d(w^*)$, $f, w^* \geq 0$. Therefore, the probability density $f(w^*)$ of an exponential random variable w^* with rate parameter λ is provided in Eq (11) as follows.

$$E[\log(w^*)] = \int_0^\infty \log(w^*)\, \lambda e^{-\lambda w^*} dw^*. \tag{11}$$

Let $\mu = \lambda w^*$, then $w^* = \frac{\mu}{\lambda}$, $dw^* = {du}/{\lambda}$. Substituting this into the integral, Eq (12) is obtained.

$$E[\log(w^*)] = \int_0^\infty \log\left({\mu}/{\lambda}\right) \lambda e^{-\mu} \frac{du}{\lambda} = \int_0^\infty (\log(u) - \log(\lambda)) e^{-u}\, du,$$

$$E[\log(w^*)] = \int_0^\infty \log(u)\, e^{-u}\, du - \log\lambda \int_0^\infty e^{-u} du. \tag{12}$$

In Eq (12), $\int_0^\infty \log(u)\, e^{u} du$ is known to be the derivative of the gamma function $\Gamma(s)$. The gamma function $\Gamma(s)$ at s=1 and $\Gamma(1^-) = -\gamma$ where γ: Euler Mascheroni constant

$$\int_0^\infty \log(u)\, e^{-u} du = -\gamma, \tag{13}$$

$$-\log\lambda \int_0^\infty e^{-u} du = -\log\lambda(1). \tag{14}$$

Substituting Eqs (13) and (14) into Eq (12), $\int_0^\infty e^{-u}du = 1$,

$$E[\log(w^*)] = \int_0^\infty \log(u)\, e^{-u}\, du - \log\lambda \int_0^\infty e^{-u}du = -\gamma - \log(\lambda), \tag{15}$$

$$E[\log^2(w^*)] = \int_0^\infty \log^2(w^*) f(w^*) dw^*. \tag{16}$$

Substituting $u = \lambda w^*$, $dw^* = \frac{du}{\lambda}$ and $w^* = \frac{u}{\lambda}$, $\log(w^*)\log\left(\frac{u}{\lambda}\right) = \log(u) - \log(\lambda)$ into Eq (16), then

$$\begin{aligned} E[\log^2(w^*)] &= \int_0^\infty [\log(u) - \log(\lambda)]^2 \lambda e^{-\lambda u/\lambda} \frac{du}{\lambda} \\ &= \int_0^\infty [\log(u) - \log(\lambda)]^2 e^{-u}\, du. \end{aligned} \tag{17}$$

Expanding and integrating Eq (17),

$$[\log(u) - \log(\lambda)]^2 = \log^2(u) - 2\log(u)\log(\lambda) + \log^2(\lambda)$$

The integration becomes

$$= \int_0^\infty \left[\left(\log^2(u) - 2\log(u)\log(\lambda) + \log^2(\lambda)\right)\right] e^{-u} du$$

$$E[\log^2(w^*)] = \int_0^\infty 2\log(u)\log(\lambda)\, e^{-u} du + \int_0^\infty \log^2(\lambda)\, e^{-u} du \int_0^\infty (\log^2(u)\, e^{-u} du. \tag{18}$$

In the Eq (18),

$$\int_0^\infty (\log^2(u)\, e^{-u} du = \frac{\pi^2}{6}, \tag{19}$$

$$\int_0^\infty 2\log(u)\log(\lambda)\, e^{-u} du = 2\log(\lambda) \int_0^\infty \log(u)\, e^{-u} du = 2\log(\lambda)(\gamma) = 2\gamma\log(\lambda), \tag{20}$$

$$\int_0^\infty \log^2(\lambda)\, e^{-u} du = \log^2(\lambda) \int_0^\infty e^{-u} du = \log^2(\lambda). \tag{21}$$

Substituting Eqs (19)–(21) into Eq (18) will obtain Eq (22) as follows:

$$E[\log^2(w^*)] = \frac{\pi^2}{6} + 2\gamma\log(\lambda) + \log^2(\lambda) = \frac{\pi^2}{6} + \left(\gamma + \log(\lambda)\right)^2. \tag{22}$$

To calculate variance, subtracting Eq (21) with the square of Eq (15) and obtaining Eq (23) as follows:

$$\text{Variance}(\mathrm{w}^*) = \mathrm{E}[\log^2(\mathrm{w}^*)] - (\mathrm{E}[\log(\mathrm{w}^*)])^2$$

$$= \frac{\pi^2}{6} + \left(\gamma + \log(\lambda)\right)^2 — \gamma - \log(\lambda)\, 2,$$

$$\text{variance}(\log(\mathrm{w}^*)) = \frac{\pi^2}{6}. \tag{23}$$

Taking square roots of Eq (23) to find the standard deviation as Eq (24):

$$\mathrm{S.D}(\log(\mathrm{w}^*) = \sqrt{\frac{\pi^2}{6}} = \frac{\pi}{\sqrt{6}}. \tag{24}$$

Now, subtracting $\log(\mathrm{w}^*)$ with Eq (15) and dividing by Eq (24) to standardize as in Eq (25).

$$\text{standardization} = \frac{\log(\mathrm{w}^*) - \mathrm{E}[\log(\mathrm{w}^*)]}{\mathrm{S.D}[\log(\mathrm{w}^*)]} = \frac{\log(\mathrm{W}^*_{\mathrm{kb}}) - (-\gamma - \log(\lambda))}{= \frac{\pi}{\sqrt{6}}}. \tag{25}$$

Substituting Eq (25) in the place of $E[log(wk)]$ in Eq (5), we get standardizations of reference data in gap statistic in the form of Eq (26),

$$\mathrm{EGS_k} = \frac{\log(\mathrm{W}^*_{\mathrm{kb}}) - (-\gamma - \log(\lambda))}{= \frac{\pi}{\sqrt{6}} + \sqrt{1 + \frac{1}{\mathrm{B}}}} - \log(\mathrm{wk}). \tag{26}$$

The scaled gap statistic EGS_k evaluates cluster validity by comparing the within-cluster sum of squares for cluster k in the bootstrap dataset $log(W^*_{kb})$ to the original dataset wk. It includes adjustments using the Euler-Mascheroni constant γ and a scaling parameter λ from an exponential distribution. The standard deviation σ is estimated as $\frac{\pi}{\sqrt{6}}$ and scaled by $\sqrt{1 + \frac{1}{B}}$ to account for variability, where B is the number of bootstrap samples. This statistic measures the standardized difference between the log-transformed within-cluster sums of squares from the bootstrap and original datasets, providing a robust measure of cluster consistency and distinctiveness. Table 1 below summarizes the discussed methods and their limitations in determining the limitations in K-means clustering.

4. Experimental setup

In this section, the limitations of K-means clustering are addressed and the experimental setup is described using a well log dataset consisting of 13 features: borehole size (BS), caliper log (CALI), corrected caliper log (CALS), density correction (DRHO), sonic travel time (DT), gamma ray (GR), deep laterolog resistivity (LLD), shallow laterolog resistivity (LLS), micro spherical focused log (MSFL), neutron porosity (NPHI), photoelectric effect (PEF), bulk density (RHOB), and spontaneous potential (SP). Each feature contains 2435 observations, sourced from Kaggle (https://kaggle.com/search?q=well+logs). The overall statistical summary is presented in Table 3.

Outlier detection and handling: Outliers are detected using the interquartile range (IQR) method.

These outliers are then handled using the winsorization technique to mitigate their impact on the clustering results.

Non-spherical data: To address this issue, an enhanced version of the rank order distance (ROD) method, termed Kernelized rank order distance (KROMD) is used. This new method combines ROD with a Gaussian kernel, effectively transforming non-spherical data into a spherical form suitable for K-means clustering.

Optimal cluster selection: The paper also enhances the gap statistic method to better determine the optimal number of clusters. The enhanced gap statistic (EGS) standardizes the reference data using an exponential distribution. This standardized approach more effectively identifies the optimal number of clusters for K-means clustering. After addressing these issues, the enhanced methods are applied to a well log dataset for lithology identification.

Table 2 utilizes descriptive statistics to effectively summarize the dataset and identify key characteristics. By presenting the mean, standard deviation, kurtosis, skewness, and count, researchers can clearly communicate their findings, setting the stage for more advanced analysis and interpretation.

Table 2. Descriptive statistic of well log dataset.

Data/measurement	Mean	Standard deviation	Kurtosis	Skewness	Count
BS	9.875	0	0	0	2435
CALI	10.83655	1.450557	0.267313	0.99747	2435
CALS	11.00636	1.495335	-0.26492	0.834237	2435
DRHO	0.063434	0.06435	1.131263	1.217352	2435
DT	75.45593	8.95537	-0.0947	0.498675	2435
GR	44.72057	26.60035	0.721314	1.082106	2435
LLD	8.602187	93.10094	1236.184	33.75293	2435
LLS	5.044212	10.50682	61.88977	6.26828	2435
MSFL	3.33086	13.09329	262.2452	14.8937	2435
NPHI	0.14008	0.067792	1.224401	0.438204	2435
PEF	3.654996	1.005118	-1.49849	0.085748	2435
RHOB	2.448815	0.163972	12.22944	-0.81415	2435
SP	-45.9933	3.431153	-0.85737	0.245552	2435

In Table 2, statistical measures are computed using 13 different well log data parameters: BS, CALI, CALS, DRHO, DT, GR, LLD, LLS, MSFL, NPHI, PEF, RHOB, and SP. Detailed data analysis for each parameter is provided in Table 3.

Table 3. Interpretation for each well log dataset given in Table 2.

Data	Mean	S.D	Kurtosis	Skewness	Interpretation
Bs	9.875	0	0	0	All observation are identical
CALI	10.83655	1.450557	0.267313	0.99747	Moderate variability, positively
CALS	11.00636	1.495335	-0.26492	0.834237	Moderate variability, positively
DRHO	0.063434	0.06435	1.131263	1.217352	Moderate variability, positively
DT	75.45593	8.9553	-0.0947	0.498675	Moderate variability, slightly right-skewed, close to normal distribution.
GR	44.72057	26.60035	0.721314	1.082106	High variability, positively
LLD	8.602187	93.10094	1236.184	33.75293	Extremely high variability, extremely heavy tails, and a long right tail.
LLS	5.044212	10.50682	61.88977	6.26828	High variability, heavy tails, and a long right tail.
MSFL	3.33086	13.09329	262.2452	14.8937	High variability, extremely heavy tails, and a long right tail.
NPHI	0.14008	0.067792	1.224401	0.438204	Moderate variability, slightly right skewed with heavier tails.
PEF	3.654996	1.005118	-1.49849	0.085748	Low variability, nearly symmetrical, and a flatter distribution.
RHOB	2.448815	0.163972	12.22944	-0.81415	Low variability, left-skewed with heavy tails.
SP	-45.9933	3.431153	-0.85737	0.245552	Moderate variability, slightly right skewed, and flatter distribution.

Variables show a mix of positive and negative skewness, mostly right skewed. Some variables (e.g., LLD, MSFL) exhibit high variability and heavy tails. BS shows no variability, while others range from low to extremely high variability.

5. Results and discussion

In this section, we initially identify outliers within the dataset. Subsequently, we address these outliers through the implementation of the winsorization technique. Following winsorization, we employ the KROMD method to transform non-spherical data into a spherical form, as outlined in the methodology section. Next, we determine the optimal number of clusters using the enhanced gap statistic method. Finally, the paper delves into a comprehensive discussion of the results obtained.

5.1. Outlier detection and handling by using the winsorization method

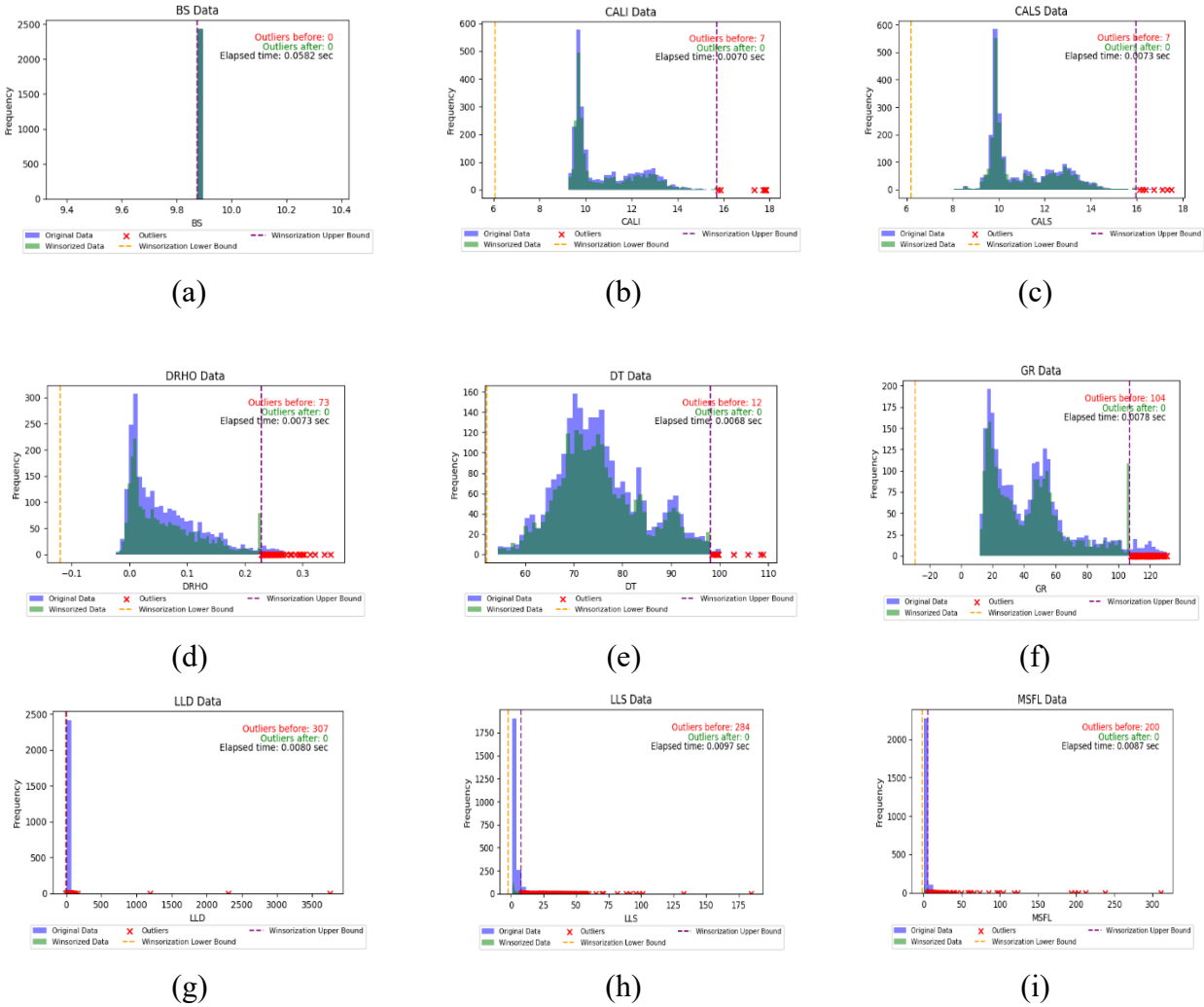
Outlier detection using the interquartile range (IQR) identifies extreme values based on the spread of the dataset. Winsorization handles outliers by replacing extreme values with the nearest non-outlier values within a specified range, helping to maintain the overall distribution of the data.

In Figure 2,

- Subplots (a), (k), and (m) illustrate the datasets for BS (elapsed time = 0.0582), PEF (running time = 0.0095), and SP (running time = 0.0067), respectively, without any identified outliers.

- Subplots (b) and (c) highlight the datasets for CALI (running time = 0.0070) and CALS (running time = 0.0073), where 7 outliers are observed in each dataset.
- Subplot (d) shows the dataset for DRHO (running time = 0.0073), with 78 outliers detected.
- Subplot (e) displays the dataset for DT (running time = 0.0068), revealing 12 identified outliers.
- Subplot (f) presents the dataset for GR (running time = 0.0078), indicating the presence of 104 outliers.
- Subplots (g), (h), and (l) represent the datasets for LLD (running time = 0.0080), LLS (running time = 0.0097), and RHOB (running time = 0.0096), respectively, exhibiting the highest number of outliers among all variables, with 307, 284, and 147 outliers, respectively.
- Subplot (j) denotes the dataset for NPHI (running time = 0.0075), indicating the presence of 20 outliers.

After detecting these outliers, the winsorization technique was applied to handle them. Figure 2 represents the winsorization of outliers using IQR mentioned in Eqs (6) and (7).



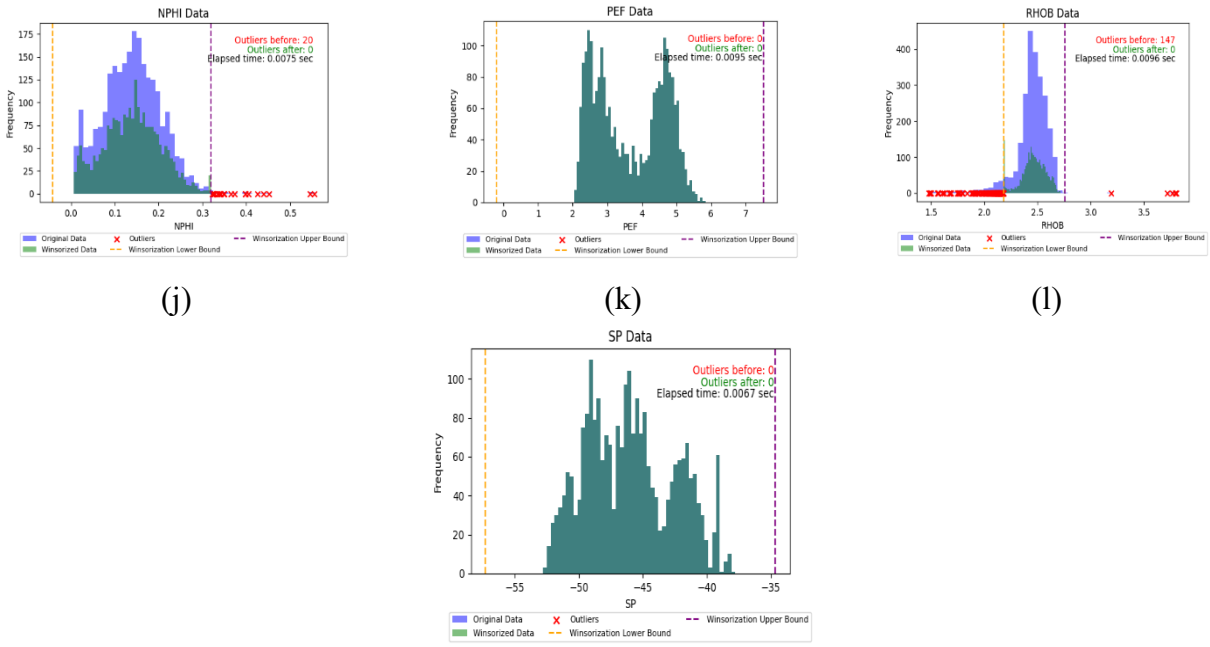


Figure 2. Detection of outliers and handling using the winsorization method.

5.2. Transformation of non-spherical data to spherical form

The transformation of non-spherical data into spherical form is achieved using the kernelized rank order Manhattan distance (KROMD) method, which integrates the Gaussian kernel with the rank order distance (ROD) method. The accompanying graph compares various methods: K-nearest neighbors (KNN), spectral clustering, PCA, ROD, and the newly developed KROMD, for converting non-spherical data into a spherical form.

Figure 3 provides a comparative analysis of different methods for transforming non-spherical data into spherical form. Each subfigure highlights a specific method:

- **Original data (a):** Serves as a baseline, showcasing the non-spherical nature of the dataset.
- **KNN (b):** Converts the data into a spherical form but may not capture the intrinsic structure as effectively as other methods.
- **Spectral clustering (c):** Demonstrates a more sophisticated approach, capturing the underlying patterns in the data better than KNN.
- **PCA (d):** Reduces dimensionality while attempting to preserve the data's variance, transforming it into a spherical form.
- **ROD (e):** Uses rank order distance to achieve the spherical transformation, providing an improved structure over traditional methods.
- **KROMD (f):** The novel KROMD method integrates Gaussian kernel and rank order distance, offering a clear and effective spherical transformation, outperforming other methods in preserving the dataset's intrinsic characteristics.

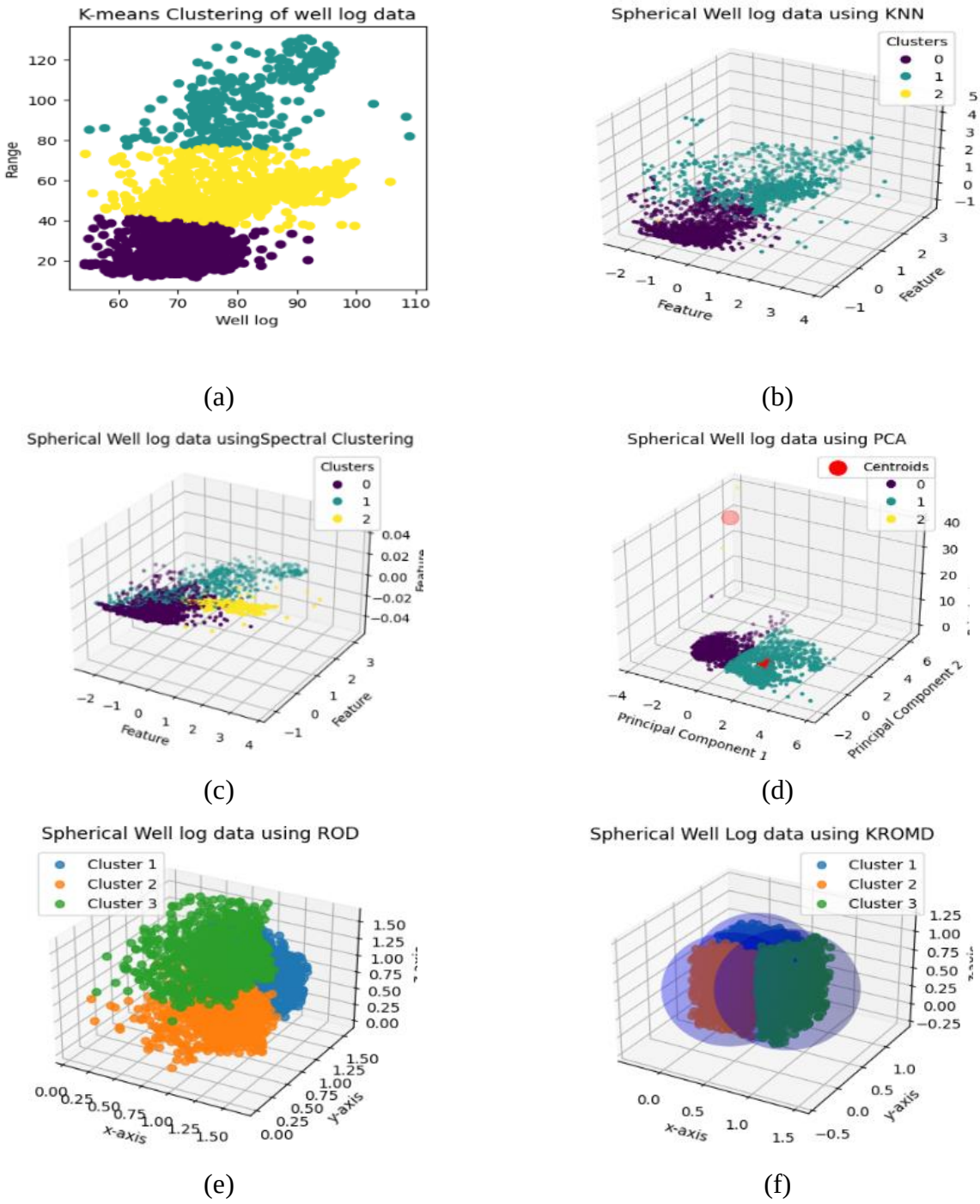


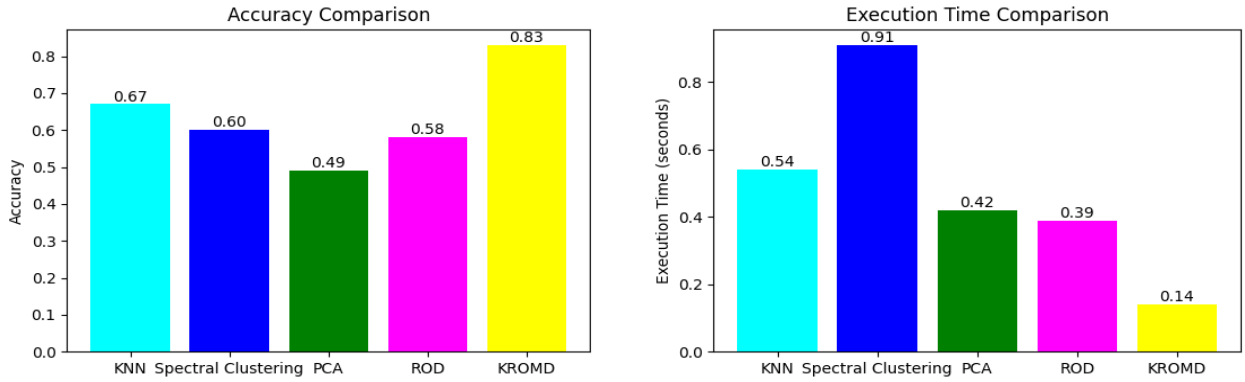
Figure 3. Conversion of non-spherical data to spherical form.

Figure 4 illustrates the accuracy levels and execution times for converting non-spherical data into spherical form using KNN, spectral clustering, PCA, ROD, and KROMD methods.

- Subfigure (a) shows the accuracy levels of each method.
- Subfigure (b) presents the execution times for each method.

The results demonstrate that the newly developed KROMD method excels in both accuracy and efficiency, as summarized in Figure 4.

KNN has an accuracy of 67%, making it the second-best performer in terms of accuracy. Its execution time is moderate at 0.52 seconds, slower than KROMD but faster than spectral clustering. **Spectral clustering** achieves a moderate accuracy of 60%. However, it has the longest execution time at 0.91 seconds, indicating lower efficiency compared to the other methods. **PCA** is the least accurate method, with an accuracy of 49%. Its execution time is relatively efficient at 0.42 seconds, but still not as efficient as KROMD or ROD. **ROD** provides an accuracy of 58%, slightly better than spectral clustering and PCA. With an execution time of 0.39 seconds, it is the second fastest method after KROMD, showing good efficiency. **KROMD** achieves the highest accuracy at 83%, significantly outperforming all other methods. It is also the most efficient, with the shortest execution time of 0.14 seconds. In conclusion, KROMD demonstrates a significant advancement over existing methods for converting non-spherical data into spherical form, offering the best combination of high accuracy and low execution time. This makes KROMD the most effective and efficient method among those compared.



(a) (b)

Figure 4. Accuracy and execution time.

5.3. Optimal number of clusters

In Figure 5, several methods are used to determine the optimal number of clusters for K-means clustering of well log datasets: (a) Davies-Bouldin index, (b) Calinski-Harabasz index, (c) silhouette plot, (d) elbow method, (e) GS, and (f) EGS. Subfigures (g) and (h) show the performance of K-means clustering after the optimal number of clusters has been selected. The results reveal that the enhanced gap statistic (EGS) method performs better than the other methods.

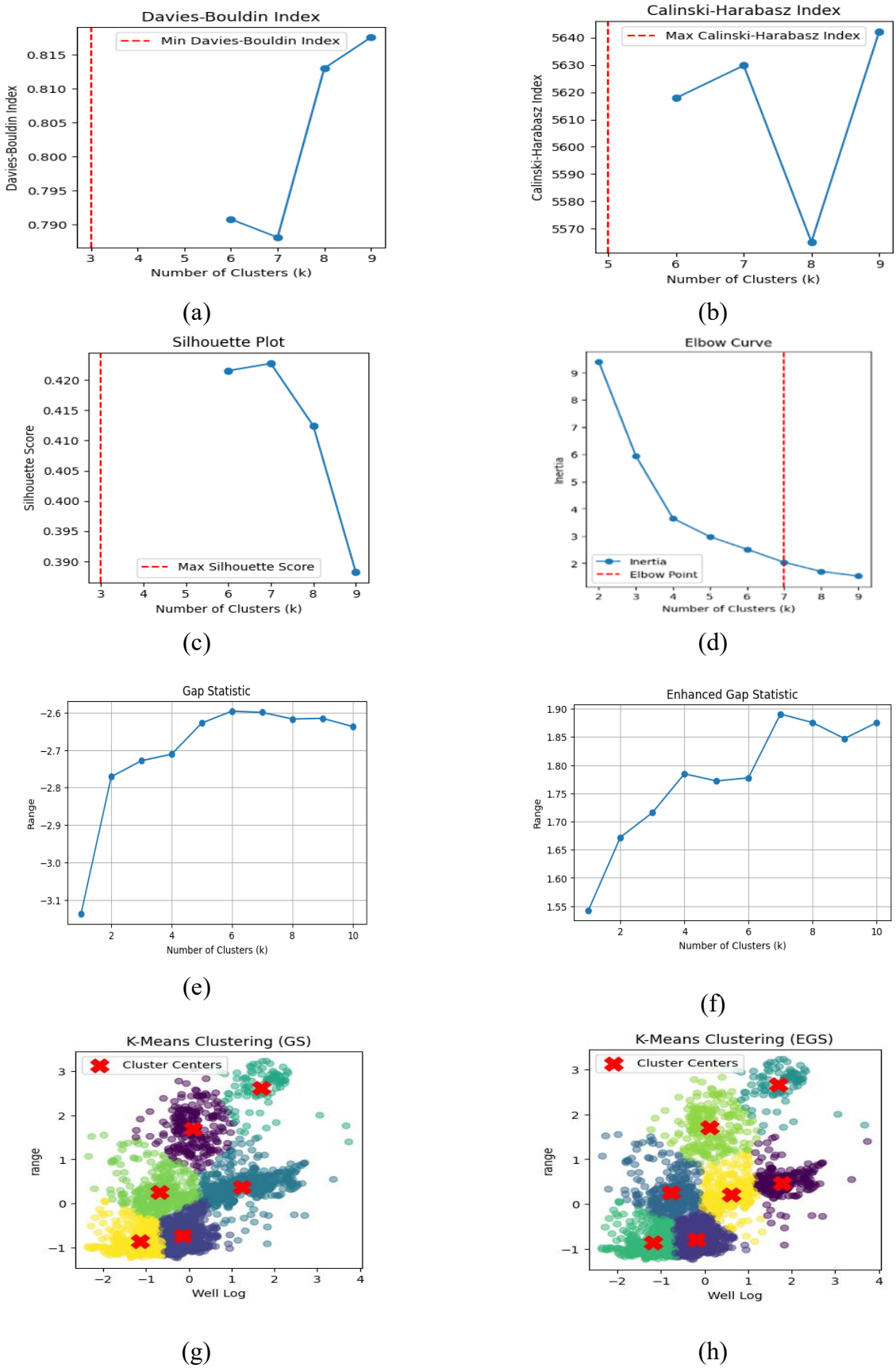


Figure 5. Optimal number of clusters selection.

Figure 6 provides an analysis of different methods for selecting the optimal number of clusters in K-means clustering:

- Figure 6 (a) displays various methods for determining the optimal number of clusters.
- Figure 6 (b) shows the execution times of these methods.

Figure 6 highlights the performance of various methods used to determine the optimal number of clusters in K-means clustering.

Davies-Bouldin (D-B) index. Accuracy: High at 91.88%, making it a robust method for cluster analysis. Execution time: 0.1431 seconds, demonstrating quick computational performance.

Calinski-Harabasz (C-H) index. Accuracy: Moderate at 65.39%, suitable for cluster evaluation. Execution time: 0.0936 seconds, the fastest among the methods.

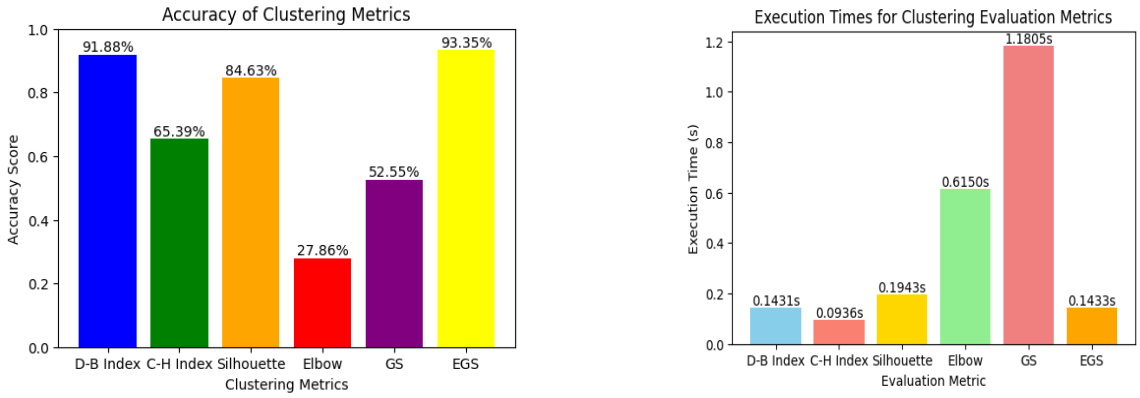
Silhouette plot. Accuracy: Respectable at 84.63%, showing reliable cluster assessment. Execution time: 0.1943 seconds, moderately efficient.

Elbow method. Accuracy: Lowest at 27.86%, suggesting limited effectiveness in cluster identification. Execution time: 0.6150 seconds, slower compared to other methods.

Gap statistic (GS). Accuracy: Relatively lower at 52.55%, indicating less optimal cluster determination. Execution time: 1.1805 seconds, the slowest among all methods.

Enhanced gap statistic (EGS). Accuracy: Highest at 93.35%, demonstrating superior performance in cluster selection. Execution time: Efficient at 0.1433 seconds, indicating fast processing speed.

In conclusion, the enhanced gap statistic (EGS) method emerges as the most effective choice for selecting the optimal number of clusters in K-means clustering, offering both high accuracy and efficient execution time.



(a) (b)

Figure 6. Accuracy and execution times.

5.4. Research contributions

This research significantly advances the K-means clustering algorithm by addressing three primary limitations. First, outlier detection and management were tackled through the implementation of the winsorization method. Second, a novel approach was introduced that combined the rank order distance (ROD) technique with a Gaussian kernel to effectively transform non-spherical data into a

spherical form. Last, a gap statistic method was defined for determining the optimal number of clusters in K-means by standardizing reference data using an exponential distribution. These enhancements have demonstrated superior performance compared to conventional methods, making a substantial contribution to the field of clustering algorithms.

6. Conclusions

This research focused on enhancing the foundational task of clustering, with a particular emphasis on K-means clustering. This paper identified three critical limitations of the K-means algorithm: sensitivity to outliers, difficulties with non-spherical data, and challenges in selecting the optimal number of clusters. To address these issues, this paper proposed innovative solutions:

Mitigating outliers: Winsorization was employed to effectively manage the influence of outliers on the clustering process.

Handling non-spherical data: Kernelized rank order distance (KROMD) was introduced to transform non-spherical data into a spherical form, enhancing clustering accuracy.

Determining optimal clusters: The gap statistic method was improved to provide a more reliable approach for selecting the optimal number of clusters.

Extensive experimentation demonstrated that our proposed methods outperformed traditional approaches, showing superior performance in handling outliers, non-spherical data, and determining the number of clusters. By addressing these critical challenges, this research significantly advances the effectiveness and applicability of K-means clustering across various domains. The paper offers practical solutions to enhance clustering performance, providing a more robust framework for data analysis and decision-making processes. For future work, we encourage the application of these algorithms to different datasets, focusing on each limitation individually, and comparing their performance with other methods in terms of accuracy and execution time.

Author contributions

Iliyas Karim Khan, Hanita Binti Daud, Nooraini Binti Zainuddin, Rajalingam Sokkalingam, Abdussamad, Abdul Museeb, and Agha Inayat: The responsibilities included algorithm development, software creation, numerical example preparation, original draft writing, and review and editing of the manuscript. All authors contributed equally to this work and have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This project received funding from National Collaborative Research Fund with cost center 015MC0-036 and OPEX Incentive for Center of Intelligent Asset Reliability (IAR) with cost center

015LB0-117 at Universiti Teknologi PETRONAS, Malaysia. The authors wish to thank the anonymous reviewers and the editor for their detailed evaluation of this paper and their valuable suggestions and insights.

Conflicts of interest

The authors declare no conflicts of interest.

Reference

1. X. Du, Y. He, J. Z. Huang, Random sample partition-based clustering ensemble algorithm for big data, *2021 IEEE International Conference on Big Data (Big Data)*, 2021, 5885–5887. https://doi.org/10.1109/BigData52589.2021.9671297
2. B. Huang, Z. Liu, J. Chen, A. Liu, Q. Liu, Q. He, Behavior pattern clustering in blockchain networks, *Multimed. Tools Appl.*, **76** (2017), 20099–20110. https://doi.org/10.1007/s11042-017-4396-4
3. Y. Djenouri, A. Belhadi, D. Djenouri, J. C. W. Lin, Cluster-based information retrieval using pattern mining, *Appl. Intell.*, **51** (2021), 1888–1903. https://doi.org/10.1007/s10489-020-01922-x
4. C. Ouyang, C. Liao, D. Zhu, Y. Zheng, C. Zhou, C. Zou, Compound improved Harris hawks optimization for global and engineering optimization, *Cluster Comput.*, 2024. https://doi.org/10.1007/s10586-024-04348-z
5. J. Xu, T. Li, D. Zhang, J. Wu, Ensemble clustering via fusing global and local structure information, *Expert Syst. Appl.*, **237** (2024), 121557. https://doi.org/10.1016/j.eswa.2023.121557
6. W. L. Zhao, C. H. Deng, C. W. Ngo, K-means: a revisit, *Neurocomputing*, **291** (2018), 195–206. https://doi.org/10.1016/j.neucom.2018.02.072
7. J. Qi, Y. Yu, L. Wang, J. Liu, *K*-means: an effective and efficient K-means clustering algorithm*, *2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, IEEE, 2016. https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.46
8. X. Wu, H. Zhou, B. Wu, T. Zhang, A possibilistic fuzzy Gath-Geva clustering algorithm using the exponential distance, *Expert Syst. Appl.*, **184** (2021), 115550. https://doi.org/10.1016/j.eswa.2021.115550
9. Y. Liu, Z. Liu, S. Li, Y. Guo, Q. Liu, G. Wang, Cloud-cluster: an uncertainty clustering algorithm based on cloud model, *Knowl.-Based Syst.*, **263** (2023), 110261. https://doi.org/10.1016/j.knosys.2023.110261
10. M. Ahmed, R. Seraj, S. M. S. Islam, The K-means algorithm: a comprehensive survey and performance evaluation, *Electronics*, **9** (2020), 1295. https://doi.org/10.3390/electronics9081295
11. T. M. Ghazal, Performances of K-means clustering algorithm with different distance metrics, *Intell. Autom. Soft Comput.*, **30** (2021), 735–742. https://doi.org/10.32604/iasc.2021.019067
12. Z. Zhang, Q. Feng, J. Huang, Y. Guo, J. Xu, J. Wang, A local search algorithm for K-means with outliers, *Neurocomputing*, **450** (2021), 230–241. https://doi.org/10.1016/j.neucom.2021.04.028

13. E. Dandolo, A. Pietracaprina, G. Pucci, Distributed K-means with outliers in general metrics, In: J. Cano, M. D. Dikaiakos, G. A. Papadopoulos, M. Pericàs, R. Sakellariou, *Euro-Par 2023: Parallel Processing. Euro-Par 2023*, Lecture Notes in Computer Science, Cham: Springer, **14100** (2023), 474–488. https://doi.org/10.1007/978-3-031-39698-4_32
14. H. He, Y. He, F. Wang, W. Zhu, Improved K-means algorithm for clustering non-spherical data, *Expert Syst.*, **39** (2022), e13062. https://doi.org/10.1111/exsy.13062
15. J. Heidari, N. Daneshpour, A. Zangeneh, A novel K-means and K-medoids algorithms for clustering non-spherical-shape clusters non-sensitive to outliers, *Pattern Recogn.*, **155** (2024), 110639. https://doi.org/10.1016/j.patcog.2024.110639
16. T. M. Kodinariya, P. R. Makwana, Review on determining number of cluster in K-means clustering, *Int. J. Adv. Res. Comput. Sci. Manage. Stud.*, **1** (2013), 90–95.
17. B. Sowan, T. P. Hong, A. Al-Qerem, M. Alauthman, N. Matar, Ensembling validation indices to estimate the optimal number of clusters, *Appl. Intell.*, **53** (2023), 9933–9957. https://doi.org/10.1007/s10489-022-03939-w
18. J. Rossbroich, J. Durieux, T. F. Wilderjans, Model selection strategies for determining the optimal number of overlapping clusters in additive overlapping partitional clustering, *J. Classif.*, **39** (2022), 264–301. https://doi.org/10.1007/s00357-021-09409-1
19. Z. Hao, Z. Lu, G. Li, F. Nie, R. Wang, X. Li, Ensemble clustering with attentional representation, *IEEE Trans. Knowl. Data Eng.*, **36** (2023), 581–593. https://doi.org/10.1109/TKDE.2023.3292573
20. Z. P. Zhang, S. Li, W. X. Liu, Y. Wang, D. X. Li, A new outlier detection algorithm based on fast density peak clustering outlier factor, *Int. J. Data Warehous. Mining*, **19** (2023), 1–19. https://doi.org/10.4018/IJDWM.316534
21. W. Wang, Y. Ren, R. Zhou, J. Zhang, An outlier detection algorithm based on probability density clustering, *Int. J. Data Warehous. Mining*, **19** (2023), 1–20. https://doi.org/10.4018/IJDWM.333901
22. Y. Liu, Z. Liu, S. Li, Z. Yu, Y. Guo, Q. Liu, et al., Cloud-vae: variational autoencoder with concepts embedded, *Pattern Recogn.*, **140** (2023), 109530. https://doi.org/10.1016/j.patcog.2023.109530
23. J. Li, X. Zhao, B. Du, Landslide induced seismic signal clustering with outlier removal, *IEEE Geosci. Remote Sens. Lett.*, **20** (2023), 1–5. https://doi.org/10.1109/LGRS.2023.3327044
24. H. Wang, P. Xu, J. Zhao, Improved KNN algorithms of spherical regions based on clustering and region division, *Alex. Eng. J.*, **61** (2022), 3571–3585. https://doi.org/10.1016/j.aej.2021.09.004
25. W. Xiong, J. Wang, Gene mutation of particle morphology through spherical harmonic-based principal component analysis, *Powder Technol.*, **386** (2021), 176–192. https://doi.org/10.1016/j.powtec.2021.03.032
26. T. Huang, S. Wang, W. Zhu, An adaptive kernelized rank-order distance for clustering non-spherical data with high noise, *Int. J. Mach. Learn. Cyber.*, **11** (2020), 1735–1747. https://doi.org/10.1007/s13042-020-01068-9
27. H. Xin, Y. Lu, H. Tang, R. Wang, F. Nie, Self-weighted Euler K-means clustering, *IEEE Signal Proc. Lett.*, **30** (2023), 1127–1131. https://doi.org/10.1109/LSP.2023.3305909
28. T. Simmons, M. Daghooghi, I. Borazjani, Dynamics of non-spherical particles resting on a flat surface in a viscous fluid, *Phys. Fluids*, **35** (2023), 043334. https://doi.org/10.1063/5.0145221

29. F. Ros, R. Riad, S. Guillaume, PDBI: a partitioning Davies-Bouldin index for clustering evaluation, *Neurocomputing*, **528** (2023), 178–199. https://doi.org/10.1016/j.neucom.2023.01.043
30. I. F. Ashari, E. D. Nugroho, R. Baraku, I. N. Yanda, R. Liwardana, Analysis of elbow, silhouette, Davies-Bouldin, Calinski-Harabasz, and rand-index evaluation on K-means algorithm for classifying flood-affected areas in Jakarta, *J. Appl. Inform. Comput.*, **7** (2023), 95–103. https://doi.org/10.30871/jaic.v7i1.4947
31. E. Schubert, Stop using the elbow criterion for K-means and how to choose the number of clusters instead, *ACM SIGKDD Explor. Newsl.*, **25** (2023), 36–42. https://doi.org/10.1145/3606274.3606278
32. N. T. M. Sagala, A. A. S. Gunawan, Discovering the optimal number of crime cluster using elbow, Silhouette, gap statistics, and NbClust methods, *ComTech: Comput. Math. Eng. Appl.*, **13** (2022), 1–10. https://doi.org/10.21512/comtech.v13i1.7270
33. R. G. Ribeiro, R. Rios, Temporal gap statistic: a new internal index to validate time series clustering, *Chaos Soliton. Fract.*, **142** (2021), 110326. https://doi.org/10.1016/j.chaos.2020.110326
34. S. Demir, E. K. Sahin, Application of state-of-the-art machine learning algorithms for slope stability prediction by handling outliers of the dataset, *Earth Sci. Inform.*, **16** (2023), 2497–2509. https://doi.org/10.1007/s12145-023-01059-8
35. I. Horenko, E. Vecchi, J. Kardoš, A. Wächter, O. Schenk, T. J. O'Kane, et al., On cheap entropy-sparsified regression learning, *Proc. Natl. Acad. Sci.*, **120** (2023), e2214972120. https://doi.org/10.1073/pnas.2214972120
36. K. K. Sharma, A. Seal, Outlier-robust multi-view clustering for uncertain data, *Knowl.-Based Syst.*, **211** (2021), 106567. https://doi.org/10.1016/j.knosys.2020.106567
37. E. Schubert, A. Lang, G. Feher, Accelerating spherical K-means, In: N. Reyes, R. Connor, N. Kriege, D. Kazempour, I. Bartolini, E. Schubert, et al., *Similarity search and applications. SISAP 2021*, Lecture Notes in Computer Science, Cham: Springer, **13058** (2021), 217–231. https://doi.org/10.1007/978-3-030-89657-7_17
38. D. S. Rini, I. Sriliana, P. Novianti, S. Nugroho, P. Jana, Spherical K-means method to determine earthquake clusters, *J. Phys.: Conf. Ser.*, IOP Publishing, **1823** (2021), 012043. https://doi.org/10.1088/1742-6596/1823/1/012043
39. N. Ukey, Z. Yang, B. Li, G. Zhang, Y. Hu, W. Zhang, Survey on exact knn queries over high-dimensional data space, *Sensors*, **23** (2023), 629. https://doi.org/10.3390/s23020629
40. O. Koren, M. Koren, A. Sabban, AutoML–optimal K procedure, *2022 International Conference on Advanced Enterprise Information System (AEIS)*, IEEE, 2022, 110–119. https://doi.org/10.1109/AEIS59450.2022.00023
41. P. Patel, B. Sivaiah, R. Patel, Approaches for finding optimal number of clusters using K-means and agglomerative hierarchical clustering techniques, *2022 international conference on intelligent controller and computing for smart power (ICICCSP)*, IEEE, 2022, 1–6. https://doi.org/10.1109/ICICCSP53532.2022.9862439
42. Jayashree, T. Shivaprakash, Optimal value for number of clusters in a dataset for clustering algorithm, In: M. Pandit, M. K. Gaur, P. S. Rana, A. Tiwari, *Artificial intelligence and sustainable computing*, Algorithms for Intelligent Systems, Singapore: Springer, 2022, 631–645. https://doi.org/10.1007/978-981-19-1653-3_48

43. M. S. Girija, B. R. Tapas Bapu, D. Magesh Babu, A variance difference method for determining optimal number of clusters in wireless sensor networks, *Res. Square*, 2023. https://doi.org/10.21203/rs.3.rs-1984952/v1
44. A. M. El-Mandouh, L. A. Abd-Elmegid, H. A. Mahmoud, M. H. Haggag, Optimized K-means clustering model based on gap statistic, *Int. J. Adv. Comput. Sci. Appl.*, **10** (2019), 183–188. https://doi.org/10.14569/IJACSA.2019.0100124
45. E. Purwaningsih, E. Nurelasari, Implementasi metode K-means clustering Dengan Davies Bouldin index pada analisis faktor penyebab perceraian, *J. Inform. Manag.*, **7** (2023), 134–143. https://doi.org/10.51211/imbi.v7i2.2307
46. G. Gan, M. K. P. Ng, K-means clustering with outlier removal, *Pattern Recogn. Lett.*, **90** (2017), 8–14. https://doi.org/10.1016/j.patrec.2017.03.008
47. F. Zubedi, B. Sartono, K. A. Notodiputro, Implementation of Winsorizing and random oversampling on data containing outliers and unbalanced data with the random forest classification method, *J. Nat.*, **22** (2022), 108–116. https://doi.org/10.24815/jn.v22i2.25499
48. L. Guo, X. Zhang, Q. Wang, X. Xue, Z. Liu, Y. Mu, Joint enhanced low-rank constraint and kernel rank-order distance metric for low level vision processing, *Expert Syst. Appl.*, **201** (2022), 116976. https://doi.org/10.1016/j.eswa.2022.116976
49. S. Yue, P. Wang, J. Wang, T. Huang, Extension of the gap statistics index to fuzzy clustering, *Soft Comput.*, **17** (2023), 1833–1846. https://doi.org/10.1007/s00500-013-1023-9

AIMS Press