Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol. 6, No. 2, September 2023, pp. 154 – 178
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Trends and Advances on The K-Hyperparameter Tuning Techniques In High-Dimensional Space Clustering

[1*]Rufus Gikera, [2]Jonathan Mwaura, [3]Elizaphan Maina, [4]Shadrack Mambo
[1]School of Computing Sciences, Riara University, Nairobi - Kenya
[3]Khoury College of Computer Sciences, North Eastern University, Boston - USA
[3]Departement of Computing, Kenyatta University, Nairobi - Kenya
[4]School of Engineering & Technology, Kenyatta University, Nairobi - Kenya
Email: [1]rgikera@riarauniversity.ac.ke, [2]jtmwaura@gmail.com,
[3]maina.elizaphan@ku.ac.ke, [4]mambo.shadrack@ku.ac.ke

Article Info
Article history:
Received: May 14th, 2023
Revised: Jun 20th, 2023
Accepted: Aug 4th, 2023

Keyword:
Clustering
High-dimensional space
K-hyperparameter tuning
Unsupervised Learning

ABSTRACT
Clustering is one of the tasks performed during exploratory data analysis with an extensive and wealthy history in a variety of disciplines. Application of clustering in computational medicine is one such application of clustering that has proliferated in the recent past. K-means algorithms are the most popular because of their ability to adapt to new examples besides scaling up to large datasets. They are also easy to understand and implement. However, with *k*-means algorithms, *k*-hyperparameter tuning is a long standing challenge. The sparse and redundant nature of the high-dimensional datasets makes the *k*-hyperparameter tuning in high-dimensional space clustering a more challenging task. A proper *k*-hyperparameter tuning has a significant effect on the clustering results. A number of state-of-the art *k*-hyperparameter tuning techniques in high-dimensional space have been proposed. However, these techniques perform differently in a variety of high-dimensional datasets and data-dimensionality reduction methods. This article uses a five-step methodology to investigate the trends and advances on the state of the art *k*-hyperparameter tuning techniques in high-dimensional space clustering, data dimensionality reduction methods used with these techniques, their tuning strategies, nature of the datasets applied with them as well as the challenges associated with the cluster analysis in high-dimensional spaces. The metrics used in evaluating these techniques are also reviewed. The results of this review, elaborated in the discussion section, makes it efficient for data science researchers to undertake an empirical study among these techniques; a study that subsequently forms the basis for creating improved solutions to this *k*-hyperparameter tuning problem.

Corresponding Author:
Rufus Gikera,
School of Computing Sciences,
Riara University,
Nairobi - Kenya.
Email: rgikera@riarauniversity.ac.ke

DOI: http://dx.doi.org/10.24014/ijaidm.v6i2.22718

1. INTRODUCTION

The clustering process aims at gaining a deeper insight into a set of unlabeled dataset, grouping similar features into a common cluster [1]. A plethora of clustering algorithms has since been proposed with the *k*- means clustering algorithm being one of most popular clustering algorithm[2] .K-means clustering algorithms are widely used in many areas because they are relatively easy to understand and implement [3]. However, the clustering results obtained from *k*-means clustering algorithm heavily depend on the *k*-hyperparameter value [2], [4]. Tuning this *k*-value correctly goes along way with improving the quality of

clustering results[5]. With high-dimensional datasets, tuning this *k*-value correctly poses great challenges to the data scientists due to the sparse and redundant nature of such datasets, among other issues [[1],[6]]. The high-dimensional datasets, generated massively by the modern technology trends, has attracted immense interest from the world of scholars and data scientists, inspired by the need to find out the best mapping in low dimensional spaces while at the same time maintaining the nature of the original high-dimensional datsets [6]. For this reason, the adoption of the automated *k*-hyperparameter tunig techniques to aid in the optimal selection of this *k*-hyperparameter value is critical to the performance of the high-dimensional *k*-means algorithms [4], [6], [7].

Although several *k*-hyperparameter tuning techniques have been proposed, identifying the optimal *k*-hyperparameter in a specific high-dimensional space remains challenging, intractable and an open research issue [6]. For instance, the auto elbow *k*-hyperparameter tuning technique has the limitation of the fact that the auto-elbow graph may depict a smooth elbow with some imbalanced high-dimensional datasets; the sharp elbow point is normally used to identify the right *k*-hyperparameter value for a specific high-dimensional dataset [8].

The rest of this paper is organized as follows: in the second section, a succinct review on the existing *k*-hyperparameter tuning techniques in high-dimensional space clustering is done. In the third section, results and discussions based on the review analysis is done and finally, in section four, the conclusions and recommendations for future research, based on the results of the review process are stated.

2. RELATED WORK

2.1. K-Means Architecture

K-means is a clustering method that divides *n* data points into *k* number of clusters with each data point getting into the cluster that posses the nearest mean [7]. A pseudo code for the *k*-means clustering algorithm can be represented as follows:

Input:

1. D = {d1, d2,. ,dn} // n data items' set.
2. K = number of desired clusters

Output: A set of *k* clusters.

The steps in this pseudo code are as follows:

1. Whimsically choose *k* data-items from D as initial centroids;
2. Repeat
   a. Assign each item di to the cluster which has the closest centroid *k*;
   b. Calculate the new mean for each cluster;
   c. Until convergence criteria is met [9].

K-means clustering algorithm has two separate phases i.e. the first phase and the second phase [7] The first phase focuses on the definition of *k*-centroids, one for each cluster while the second phase focuses on taking each point belonging to the given dataset and associating it to the nearest centroid. In order to determine the distance between data points and the centroids, Euclidean distance is generally considered [10]. Once all the points are included in some clusters, the first step is finished and an early grouping is done [11]. At this juncture, new centroids are recalculated because the inclusion of new points can lead to a change in the cluster centroids [12]. Once the new *k* centroids are found, a new binding is created between the same data points and the nearest new *k* centroid, generating a kink [9], [13]. The *k* centroids may change their position in a step by step manner, as a result of this *k* in *k* [13]. Finally, a situation is arrived at where the centroids do not move any further, an indication for the convergence criterion for clustering [14].

2.2. Description of A High-Dimensional Dataset and Its Cluster Analysis Challenges

The modern technology trends have resulted to massive high-dimensional datasets [15].High dimensional statistics and the related study have attracted keen interest from a plethora of data scientists [15]. The dynamism in regards to sparsity and redundancy of the high-dimensional space pose great data mining challenges to data scientists [16]. High-dimensional datasets refer to those datasets whose number of features / attributes, P, is greater, in one or several orders of magnitude, than the number of instances / observations, N, i.e. P>N [15]. Mathematically, "orders of magnitude" refers to a system of classification determined by size, typically in powers of ten [17]. According to [15], it is mostly common to find high dimensional datasets in the field of medicine [[15],[6]]. An example is where the number of attributes for a particular patient are many i.e. body-mass index, blood pressure values, diagnosis history, family history on illnesses, height, weight, status of immune system etc[15].In genomics and proteomic, each sample can be defined by multiple measurements up to a thousand [[15], [6], [18]].

Table 1. An example of a high-dimensional dataset comprised of three patients (*N*) and multuple features (*P*) [19]

P / *N*	*BP*	*Height*	*Weight*	*Diagnosis*	…	…
Patient 1						
Patient 2						
Patient 3						

The process of unveiling meaningful and hidden patterns when clustering high-dimensional datsets, poses a number of challenges for the data scientists [18].Such challenges mainly include: curse of dimensionality, presence of noise, tuning for the optimal *k*-hyperparameter value, presence of outliers and redundant features in a dataset[18]. The curse of dimensionality as well as tuning for the optimal *k*-hyperprameter value from a high-dimensional datasets makes most of the high-dimensional *k*-means algorithms sensitive to the clustering perfomance [14], [18]. Data dimensionality reduction has been greatly used to solve the challenges of curse of dimensionality during the high-dimensional cliuster analysis [14]. Such data dimensionality reduction aim at efficiently representing a high-dimensional dataset in low-dimensions but with as much minimal information loss, as possible [14] . Some *k*-means algorithms are sensitive to both the dimensionality redundancy and biasing and the need to invent algorithms that are independent of such limitations is critical to the success in this area[20]. Lastly, identification of the optimal *k*-hyperparameter from a high-dimensional space is still a challenging task and there is need for researchers to invent mechanisms that aid in automated and accurate methods of identifying this value [13]. Identification of this value, correctly, has a significant effect on the performance of the *k*-means models [18].

2.3. Performance and Statistical Metrics for Evaluating Quality of Clusters from K-Means Clustering Algorithms

In unsupervised clustering algorithms like the *k*-means, the ground truth about the *k*-hyperparameter value, the number of clusters on a specific dataset, relies on the prior knowledge of the problem [21] .In most cases, there is no prior knowledge or intuition about the clustering dataset, at hand, and at times, the domain knowledge is required [22]. For this reason, it is important to use the metrics that give some intuition about the best or the optimal value of *k* on any clustering high dimensional dataset [3],[23]. Such a standard cluster validation process and set of internal validation metrics, is highly critical to assessing the quality of the k clusters generated as the output from the high-dimensional *k*-means algorithms [24], [25]. The optimal *k*-value, in *k*-means, dictates the best clustering results [26]. At this optimal, the variance within a cluster is normally low, while the separation between clusters is normally high [27], [15]. Some of the most commonly applied metrics include:

2.3.1. Internal Validation Indexes

The choice of the internal validation metrics, as opposed to the external and relative validation metrics, is based on the fact that the internal validation metrics are purely based on the information intrinsic to the data alone with no clue on prior information about the dataset [11]. The Internal indexes are known to be better while applied in the determination of the quality of the clustering results because they are purely based on the information intrinsic to the data alone [11], [28]. The most commonly used internal validity metrics, in the clustering literature; include Dunn index, calinski harabsz index, Davies Bouldin index, Silhouette index, bayesian information criterion, point bi-serial and sum-of-squares. Comparing the scores of the different pairs of internal validation metrics, for one dataset, as well as comparing their consistency using Kendall's index, each at a time, would be computationally expensive [28]. For this reason, we propose to adopt an ensemble validation metric whose components exercise equal sensitivity to the varied conditions present in the high dimensional datasets. This type of ensemble could either be bootstrap aggregating or (bagging) or boosting [11]. The most commonly used internal validity metrics, in the clustering literature, include:

1. **Dunn Index (DI)**
   Dunn index, an internal validity metric, indicates a high degree of compactness of the objects belonging to the same cluster and a high degree of separation between objects in different clusters [29]. Dunn index is defined mathematically as follows:

$$DI = \frac{m1 \le i \le j \le m^{Min\delta(C_i, C_j)}}{\max_{1 \le k \le m} \Delta_k} \quad (1)$$

Where distance between clusters i and j be denoted by δ (C_i , C_j) and the Δk is the size of cluster [30]. Higher values of the Dunn index indicate the minimum intra-cluster distances as well as the maximum inter cluster distance [31]. The *k*-means algorithms have successfully used the Dunn index to validate the clustering results that they generate [31].

2. Calinski-Harabasz Index (CH)

Calinski-Harabasz index (CH), an internal validity metric, is referred to as the ratio between the "sums of between-clusters dispersion" and "inter-cluster dispersion" for all clusters [32], [11]]. Calinski-Harabasz index, as follows, mathematically:

$$\text{CH(K)} = \frac{B(K)(N-K)}{W(K)(K-1)} \quad (2)$$

$$\text{B(K)} = (\Sigma_{k=1}^{k} \; a_k \parallel \bar{x}_k - \bar{x} \parallel^2) \quad (3)$$

$$W(K) = (\Sigma_{k=1}^{k} \Sigma_{c(j)=k} \| x_j - \overline{x_k} \|^2) \quad (4)$$

Where *k* is the corresponding number of clusters, B(K) is the inter-cluster divergence, also called the inter-cluster covariance, W(K) is the intra-cluster divergence, also called the intra-cluster covariance, and N is the number of samples [31] The larger the B(K) is, the higher the degree of dispersion between clusters is [33]. The smaller the W(K) is, the closer the relationship in the cluster [1]. Higher CH values are better because they are an indication of a good quality clustering performance and results [32].

3. Davies Bouldin Index (DB)

Davies-Bouldin index (DB), an internal validity metric, is used to identify cluster overlap by measuring the ratio of the sum of the "within-cluster scatters" to the "between-cluster separations" [34][35]. Davies-Bouldin index is defined as follows:

$$\text{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \ne i} \left(\frac{\bar{c}_i + \bar{c}_j}{\|w_i - W_j\|_2} \right) \quad (5)$$

A DB index that is close to zero (0) is an indication that the clusters are compact and far from each other [26]. The implementation of *K*-Medoids algorithm with Davies-Bouldin-Index evaluation for Clustering Postoperative Life Expectancy in Patients with Lung Cancer is an example of an algorithm that has applied Davies Bouldin index in its cluster analysis [32].

4. Silhouette Index (SI)

Silhouette index, an internal validity metric, is referred to as the optimal clustering number derived from the difference between the average distance within the cluster and the minimum distance between the clusters [21] 51, 83,100]. Silhouette index is defined mathematically as follows:

$$\bar{s} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \right) \quad (6)$$

Where a(*i*) represents the average distance of sample *i* to other samples in the cluster, b(*i*) represents the minimum distance of the sample from the sample i to the other clusters . The Silhouette index for determining optimal k-means clustering on images in different color models is an example of an algorithm that has applied silhouette index in its cluster analysis [26], [36].

However, it is important to note that calinski harabsz index, silhouette index, Dunn index and Davies Boulldin index poses some challenges at the individual level. These challenges mainly include: Sensitivity to Cluster Density, dependency on cluster size, lack of ground truth labels, interpretation challenges and sensitivity to data scaling and data dimensionality [32]. For example, the calinski harabsz indexs tend to favor clusters with similar densities [11]. If the clusters have significantly different densities, the index may not accurately capture the clustering quality [32]. It may give higher scores to clusters that are denser, even if they are not necessarily well-separated. The index is

sensitive to the number of clusters and the size of the dataset. It tends to favor solutions with a larger number of clusters, which may lead to overfitting or over-segmentation of the data. This can result in inflated index values, making it challenging to determine the optimal number of clusters [11]. These indexes do not rely on ground truth labels [32]. They assess the clustering quality based on the internal structure of the data, without considering the true underlying classes [11]. Therefore, their effectiveness may be limited when compared to metrics that utilize ground truth information. Similar to the Dunn index, interpreting the absolute value of the Calinski-Harabasz index can be difficult [32]. It lacks a clear threshold or guideline to determine what constitutes a good or bad clustering result. It is often used comparatively, comparing different clustering solutions or tuning the number of clusters to find an optimal solution [32]. The Calinski-Harabasz index can be sensitive to the scaling of the data and the number of dimensions [11]. Inconsistent scaling or high-dimensional data can impact the distances used in the calculation and lead to biased results [11]. Proper data preprocessing and dimensionality reduction techniques may be necessary to address these issues.It is therefore important to consider these challenges when using the each of these internal validation indexes and complement them with other indexes inorder to gain a more comprehensive understanding of the clustering quality.

5. Bayesian Information Criterion (BIC)

Bayesian information criterion (BIC), an internal validity metric, is referred to as a strategy for model selection among a finite set of models [21]. The model with the lowest BIC value is the most preferred as it is an indication of good clustering results [21]. Bayesian information criterion index is calculated as follows:

$$BIC = 2\log(L) + q\log(N) \tag{7}$$

L is the maximum likelihood function of the model and N is the number of data points in a dataset [26]. In evaluating clustering quality using the bayesian information criterion, some challenges crop up [26] These include: sensitivity to model complexity, limited consideration of cluster structure, dependency on initialization and algorithm choice and limited comparability across datasets [21]. BIC penalizes complex models to avoid overfitting, determining the appropriate number of clusters where it the BIC tends to favor more clusters as it penalizes model complexity, potentially leading to over-segmentation or identifying spurious clusters [21]. Interpreting the BIC values and identifying the appropriate number of clusters often requires domain knowledge and additional analysis. The BIC primarily focuses on the goodness of fit and model complexity but may not fully capture the underlying structure of the clusters [21]. It does not explicitly account for cluster separability, irregular cluster shapes, or overlapping clusters [26]. Therefore, the BIC alone may not provide a comprehensive assessment of clustering quality, and it is advisable to consider other metrics or visual inspections to validate the results [21]. The BIC can be sensitive to the initialization of clustering algorithms and the choice of algorithm itself [26]. Different initializations or algorithms may yield different BIC values and clustering results [21]. It is important to be aware of this dependency and perform multiple runs with different initializations or algorithms to obtain more robust results [26]. The BIC values may not be directly comparable across different datasets due to variations in data characteristics and underlying distributions [21]. It is more appropriate to compare BIC values within the same dataset or similar datasets rather than between different datasets [26]

6. Point Bi-Serial

This looks for the difference between the mean intra-cluster distance and the mean inter-cluster distance [21]. The point bi serial's formula is formulated as follows:

$$\bar{d}_s - \bar{d}_c * \frac{\sqrt{(\alpha * \beta / x^2}}{\alpha} \tag{8}$$

d_c is the distance from each data point and every other data point within the cluster while d_s refer to the distance from each data point and every other data point that is not within its cluster [21]. The α refers to the intra-cluster distances while the β refers to the number of the inter-cluster distances [26]. The x refers to the actual number of the point-pairs within a clustering dataset [21]. The σ refers to the standard deviation of all the distances [21]. Point bi-serial internal validation metric resembles the popular silhouette metric, except the fact that it computes the separation from all the non-cluster sharing points, instead of only those that have the closest cluster [26]. When using point bi-serial to

assess quality of clusters, some challenges come up. These include: limited applicability to clustering, lack of ground truth labels, inadequate representation of cluster quality and difficulty in Interpretation [21]. Point bi-serial is primarily designed to evaluate relationships between binary and continuous variables [26]. It may not directly apply to assessing clustering results, which involve grouping similar data points together [21]. Clustering typically deals with unsupervised learning and does not involve predefined binary variables that it requires [26]. Point bi-serial requires a binary variable as a reference point to calculate the correlation coefficient [21]. However, in clustering, there are typically no ground truth labels available to construct such a binary variable [21]. Clustering is an unsupervised learning task, and the absence of ground truth labels makes it challenging to apply point bi-serial directly [26]. Point bi-serial assesses the strength and direction of the relationship between a binary variable and a continuous variable [21]. While this can be useful in certain analyses, it may not adequately capture the quality of clustering results [26]. Clustering quality evaluation often focuses on factors such as compactness, separation, or similarity within and between clusters, which are not directly addressed by it [21]. Lastly, the interpretation of the point-biserial can be challenging, as it measures the strength and direction of a relationship. In clustering, the goal is to assess the quality of clusters rather than the correlation between variables [21]. Therefore, the interpretation of point bi-serial in the context of clustering may not provide meaningful insights.

7. **Sum of Squares**

This method adapts to the Calinski-Harabasz Index of the CH method:

$$\frac{trace(WCSM)}{trace(BCSM)} * k \tag{9}$$

Therefore, this metric is a reverse of the separation-compactness relationship [26]. For this reason, the change on the factor of normalization factor is drastic when the value of the *k* increases [21]. Just like the Davies Bouldin index, the sum of squares metric divides compactness by separation [26]. Lower values in this metric indicate better clustering on a particular dataset [37]. Challenges of suing the sum of squares to assess the cluster quality include: sensitivity to cluster size and dimensionality, lack of normalization, dependency on initialization as well as the insensitivity to cluster shape [26]. The sum of squares is influenced by the number of data points in each cluster and the dimensionality of the data. In *k*-means clustering, for instance, clusters with a larger number of data points tend to have higher sum of squares values [21] . Similarly, in high-dimensional data, the sum of squares can be inflated due to the curse of dimensionality [26]. As a result, the sum of squares may not accurately reflect the quality of clusters in scenarios where the number of points or the dimensionality varies significantly [21]. The sum of squares does not inherently account for the scale or variance of the data [26]. It treats each feature equally and does not consider differences in magnitude or variability between features [21]. Consequently, clusters may be biased towards variables with larger scales or higher variances, potentially leading to a misleading assessment of cluster quality [26]. In iterative clustering algorithms like *k*-means, the initialization of cluster centroids can significantly affect the resulting sum of squares [21]. Different initializations can yield different cluster assignments and, consequently, different sum of squares values [26]. Consequently, the choice of initialization can impact the interpretation of cluster quality based on the sum of squares [21]. The sum of squares primarily measures the dispersion of data points within clusters [26]. However, it does not explicitly capture the shape or structure of the clusters [26]. Clusters with different shapes, such as elongated or irregular clusters, may have similar sum of squares values despite their inherent structural differences [26]. Therefore, the sum of squares alone may not provide a comprehensive evaluation of cluster quality in terms of shape or compactness [26]. To address these challenges, it is often recommended to use the sum of squares in combination with other clustering evaluation metrics.

2.3.2. Clustering Accuracy

In machine learning, clustering algorithms are often evaluated using different metrics rather than a single accuracy score. Clustering is an unsupervised learning task, meaning that there are no ground truth labels available to directly calculate accuracy. Instead, various evaluation measures are used to assess the quality of clustering results. Here are some commonly used metrics:

1. **Adjusted Rand Index (ARI)**

Adjusted rand index measures the similarity between the true cluster assignments and the predicted clusters, considering all pairs of samples. ARI ranges from -1 to 1, where a higher value indicates better clustering quality.

$$ARI = (RI - E) \,/\, (\max(RI) - E) \tag{10}$$

2. **Normalized Mutual Information (NMI)**
   Normalized Mutual Information computes the mutual information between the true labels and the predicted clusters, normalized by entropy measures. NMI ranges from 0 to 1, with 1 indicating perfect clustering.

$$NMI = 2 * \frac{h*c}{h+c} \tag{11}$$

3. **Homogeneity, Completeness, and V-measure**
   These three metrics provide a more detailed evaluation of clustering. Homogeneity measures the extent to which each cluster contains only samples from a single class. Completeness measures the extent to which all samples from a given class are assigned to the same cluster. V-measure is the harmonic mean of homogeneity and completeness.

$$h = 1 - \frac{H(C \backslash K}{H(C)} \tag{12}$$

$$H(C \setminus K) = -\sum_{c,k} \frac{n_{ck}}{N} \log \left(\frac{n_{ck}}{n_k}\right) \tag{13}$$

It's important to note that the choice of metric depends on the specific problem and the nature of the data. Additionally, these metrics may not always capture all aspects of clustering quality, so it's often recommended to consider multiple metrics and interpret the results collectively.

2.3.3. Jaccard Coefficient

Jaccard coefficient is a cluster quality assessment metric that shows the degree of closeness between the clustered values and the actual values and evaluates the ability of the clustering algorithm [15]. Jaccard coefficient is used to investigate similarities between data points and the evaluation based on these similarities [8].

Jaccard Index = (the number in both sets) / (the number in either set) * 100

$$J(X,Y) = |X \cap Y| \,/\, |X \cup Y| \tag{14}$$

When using the Jaccard coefficient to assess cluster quality, some challenges prop up. These include: binary data requirement, sensitivity to set size, lack of ground truth labels, interpretation challenges and limited consideration of cluster structure [8]. The Jaccard coefficient assumes that the data is binary or can be converted into binary form. It calculates the similarity between sets by measuring the intersection over union [15]. If the data is not naturally binary or cannot be converted to binary representation, the Jaccard coefficient may not be applicable [8]. The Jaccard coefficient is sensitive to the size of the sets being compared [8] It tends to yield higher similarity values for smaller sets, even if they share a relatively small number of elements [8]. As a result, the Jaccard coefficient may bias towards smaller clusters or clusters with fewer data points [15]. The Jaccard coefficient, like other unsupervised clustering evaluation metrics, does not rely on ground truth labels [8]. It assesses the similarity between clusters without considering the true underlying classes [15]. This can limit its effectiveness as a standalone measure of clustering quality, as it does not account for the true clustering structure [8]. Interpreting the absolute value of the Jaccard coefficient can be challenging [15]. It measures the similarity between sets, ranging from 0 to 1, where 1 indicates identical sets [8]. However, determining an appropriate threshold or guideline to define good or bad clustering results based on the Jaccard coefficient alone can be subjective and context-dependent[15]. The Jaccard coefficient focuses on measuring the similarity between sets or clusters without considering the internal structure of the clusters [8]. It does not explicitly account for factors such as compactness, separation, or cluster shapes. Therefore, using the Jaccard coefficient alone may not provide a comprehensive assessment of clustering quality.

2.3.4. F1-Score

F1 score refers to the clustering metric that assesses the clustering algorithm's accuracy on a dataset [20]. F1 score can be applied in the assessment of the systems used in binary classifications that cluster data points into two, for example, e.g. true / false or yes / no [38]. Larger F1 scores are better than lower F1 scores, on a range of 0 and 1 [39]. The F1 score formula is as follows:

$$F - score = 2 * (precision * recall) \,/\, (precision + recall) \quad (15)$$

Some of the challenges posed with the use of the F1-score in assessing the quality of clusters include: dependency on predefined metrics, sensitivity to imbalance, inability to capture cluster structure and difficulty in interpretation [38]. To calculate the F1 score for clustering, it is necessary to define criteria for true positives, false positives, and false negatives [20]. This requires specifying rules or thresholds to determine if two clusters should be considered as matching or not [38]. Selecting appropriate criteria can be subjective and may vary depending on the specific clustering task and the nature of the data [20]. The F1 score is influenced by the balance or imbalance of cluster sizes [20]. If the clusters are imbalanced, with significantly different numbers of data points, it can affect the precision and recall values and consequently impact the F1 score [20]. Imbalanced clusters can lead to biased F1 score results, as the metric may be more influenced by the larger clusters [38]. The F1 score evaluates the agreement between the predicted and ground truth cluster assignments based on a flat comparison [20]. It does not explicitly consider the structure, shape, or relationships between clusters [38]. Consequently, the F1 score may not fully capture the quality of clustering in terms of compactness, separation, or cluster interdependencies [20]. Similar to other evaluation metrics, interpreting the absolute value of the F1 score can be challenging [20]. There is no universally defined threshold or guideline to determine what constitutes a good or bad clustering result based on the F1 score alone [20]. The interpretation of the F1 score should be performed in conjunction with other metrics, domain knowledge, and visual inspection to gain a comprehensive understanding of clustering quality [38]

2.3.5. Cochran's Q Score

Cochran's Q test is a non-parametric statistical test applied in heterogeneous meta-analyses [40]. Cochran's Q score is based on the chi-square distribution [41]. It creates a probability that when maximized, it indicates high variation across the subjects of study as opposed to the variations of the subjects within a study [40]. In the evaluation process of clustering algorithms, Cochran's Q score can be used to investigate if different algorithms lead to different quality of clusters on the same dataset [42]. Cochran;s Q statistic is computed as follows:

$$T = k(k-1)\frac{\sum_{j=1}^{k}\left(X_{\ell j}-\frac{N}{k}\right)^2}{\sum_{i=1}^{b} X_{io}(k-X_{io})} \quad (16)$$

Where:

- k : number of treatments
- x• j : column total for the jth treatment
- b : number of blocks
- Xi : row total for the ith block
- N : grand total

When using Coshran's Q score to assess the quality of clusters, some challenges crop up. These include: difficulty in interpreting significance, sensitivity to sample size and cluster imbalance and insensitivity to cluster structure: Cochran's Q test determines whether there are significant differences between groups [42]. However, it does not provide insights into the nature or magnitude of these differences. Interpreting the significance of the test results in the context of clustering quality can be challenging without additional information or domain knowledge. Cochran's Q test can be sensitive to the sample size and the distribution of data across clusters [40]. Unequal cluster sizes or imbalanced data can influence the test results and potentially bias the assessment of clustering quality [42]. Cochran's Q test primarily evaluates the differences between groups without explicitly considering the structure or internal characteristics of clusters [42]. It does not account for factors such as compactness, separation, or cluster shapes [40]. Therefore, relying solely on Cochran's Q score may not provide a comprehensive assessment of clustering quality in terms of these important cluster characteristics [42]. Given these challenges, Cochran's Q test may not be the most suitable method for assessing the quality of clusters.

2.3.6. Chi-Square

Chi-square is a non-parametric statistical metric used to measure of the variance between the observed and expected recurrence of the results of a variables set [43]. Chi-square is important in the analysis of such differences in categorical variables of nominal in nature [44]. A chi square is computed as follows:

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (17)$$

Where:

- The subscript "c" is the degrees of freedom
- "O" is your observed value
- E is your expected value

2.3.7. T-test

T-test refers to a statistical test applied in the comparison of two categories [28]. In clustering algorithms, t-tests are normally applied in hypothesis testing to find out if an algorithm actually has an effect on a dataset, or whether two clustering algorithms are dissimilar from each other in terms of the performance and evaluation metrics used to in the assessment process in clustering a specific dataset [42]. T-test is computed as follows:

$$t = (X^- - \mu 0) \,/\, (s/\sqrt{n}) \quad (18)$$

Where:

- X^- is the sample mean
- $\mu 0$ represents the population mean
- *s* is the standard deviation of the sample
- *n* stands for the size of the sample

When using the Chi squre and T-test metrics to assess thequality of clusters, some challenges come up. These include: interpretation challenges, sensitivity to sample size and cluster imbalance and limited consideration of cluster structure[44]. These tests assess the independence between categorical variables or groups [44]. While statistical significance can be determined, interpreting the practical significance and meaningfulness of the results in the context of clustering quality can be challenging [43]. The tests provide insights into the presence or absence of associations between variables but do not provide direct information about the quality of clustering [43]. These tests can be influenced by the sample size and the distribution of data across clusters [44]. Small sample sizes or imbalanced data may affect the test's power and potentially bias the assessment of clustering quality [43]. Moreover, unequal cluster sizes can impact the statistical significance of the the test results [44]. These tests focus on measuring associations between categorical variables but do not explicitly consider the structural characteristics of clusters [44]. They do not account for factors such as compactness, separation, or cluster shapes [43]. Therefore, relying solely on the chi-square or t-test may not provide a comprehensive evaluation of clustering quality in terms of these important clusters attributes. Considering these challenges, the chi-square test may not be the most appropriate method for assessing the quality of clusters**.**

2.3.8. Sum of squared error

The sum of squared error refers to the variation between the perceived values and the foreseen values [39]. Sum of squared error can also be referred difference of foreseen values from the actual values [40].In the evaluation of clustering algorithms, an example of the sum of squared error would be identifying the variation between the expected running time values of a clustering algorithms against the actual running time values of the same algorithm [45]. Sum of squared error is computed as follows:

$$SE = \sum ni = 0\big(yi - f(Xi)\big)2 \quad (19)$$

Where:

- yi is the *i*th value of the variable to be predicted
- *f(xi)* is the predicted value
- *xi* is the *ith* value of the explanatory variable

When using the sum of squared error, some challenges come up. These include: sensitivity to cluster size and density, dependency on initialization, lack of normalization, Insensitivity to cluster shape and structure and lack of external validation [40]. The SSE is influenced by the number of data points in each cluster [45]. Clusters with larger numbers of data points tend to have higher SSE values, even if the clustering is of good quality [45]. Consequently, the SSE may bias towards larger clusters and may not accurately reflect the overall clustering performance, especially in scenarios with varying cluster sizes or

densities [40]. The SSE is sensitive to the initialization of cluster centroids, particularly in iterative algorithms like *k*-means clustering [45]. Different initializations can lead to different SSE values and, consequently, different cluster assignments [45]. The choice of initialization can impact the interpretation of clustering quality based on the SSE, making it less reliable as a standalone metric [40]. The SSE does not inherently account for the scale or variance of the data [40]. It treats each feature equally and does not consider differences in magnitude or variability between features[45]. Therefore, the SSE may be biased towards variables with larger scales or higher variances, potentially leading to a distorted assessment of clustering quality [40]. The SSE measures the dispersion of data points within clusters but does not explicitly capture the shape, structure, or interrelationships between clusters [40]. Consequently, clusters with different shapes, densities, or structural complexities may have similar SSE values, even if they differ significantly in terms of their quality or characteristics [45]. The SSE is an internal evaluation metric that assesses the compactness of clusters based on the distances between data points and their assigned cluster centroids [40]. However, it does not consider external validation or ground truth labels [40]. Without external validation, it is challenging to determine if the obtained clustering solution is meaningful or corresponds to the true underlying structure of the data [45]. To overcome these challenges, it is recommended to use the SSE in conjunction with other clustering evaluation metrics, such as silhouette coefficient, adjusted Rand index, or other suitable measures.

3. METHODOLOGY

The systematic review process of the *k*-hyperparameter tuning techniques followed the five-step methodology as proposed by Khan, Kunz, Kleijnen, and Antes for conducting critical review on the existing literature [6]. The five-step methodology involves: framing the questions for the review, identification of relevant literature, assessment of the quality of articles, critical review summary of the reviewed literature, and the interpretation of results [18]. Based on this five-step methodology, we first explain how the research questions for the review were framed, followed by the identification of the relevant literature [46]. Next, the criterion used to perform quality assessment of the articles is done [20]. Thirdly a critical review summary of the reviewed literature is done and discussion of the results is done as the culmination of the five-step methodology. After the review process, a number of recommendations for the future research directions, in the last section were proposed. This provides a guide /foundation for further work on solving the tuning problems faced with the *k*-hyperparameter of the *k*-means algorithms in high-dimensional space.

3.1. Framing of The Review / Research Questions

The research questions in this study were formulated as follows:

1. ***RQ1***: Which *k*-hyperparameter tuning techniques and data dimensionality reduction methods are used in high-dimensional spaces?
2. ***RQ2***: What is the nature and dimensionality of the input datasets used with the existing *k*-hyperparameter tuning techniques in high-dimensional spaces?
3. ***RQ3***: What are the key algorithm performance and statistical metrics for evaluating the existing *k*-hyperparameter tuning techniques in high-dimensional spaces?

In the first research question, we conduct a succinct review of the existing *k*-hyperparameter tuning techniques, data dimensionality reduction methods used with these techniques as well as their tuning strategies and limitations. The results are then tabulated as shown in table 1. In the second research question, the analysis of the nature, description and dimensionality of the datasets used with the *k*-hyperparameter tuning techniques is done. The results are tabulated as shown in table 2. In the third research question, we investigate the key algorithm performance and statistical metrics for evaluating the existing *k*-hyperparameter tuning techniques in high-dimensional spaces. The results on these evaluation metrics are reported in table 3.

3.2. Literature Identification

The meta-search based strategy for identifying the relevant literature focused on the *k*-hyperparameter tuning techniques on the high-dimensional *k*-means clustering algorithms. The keywords used include "*k* in k-means". "Optimize" AND "*k*-means" was also used. Optimize was also replaced with its synonyms like "efficient" and "improved" in order to generate more results that are relevant to the literature on the *k*-hyperparameter tuning techniques. The databases incorporated an in-depth search for articles from the Google Scholar, ACM, Research Gate, IEEE Xplore digital library, Springer, among others. We identified a total of 26 articles, published between 2013 and 2022.

3.3. Assessment on Quality and Criteria for Selection

The quality assessment and selection criteria took a narrow focus on the review of the abstracts of all the 26 papers. After assessment on criteria process, we identified 16 papers that were relevant to the *k*–

hyperparameter tuning techniques on the *k*-means clustering algorithms. These papers were reviewed, analyzed and discussed under the section on the results.

3.4. Inclusion Criteria

The inclusion criterion that was used in obtaining the relevant papers for this study include; articles that addressed the *k*-hyperparameter tuning in high-dimensional spaces, those that described original work with the actual *k*-hyperparameter tuning tool having been developed and validated in high-dimensional spaces. The papers included in this research review article is within the last five years in order to give the most recent status on the research progress of the *k*-hyperparameter tuning as opposed to reference to the old literature on these techniques. Such information, based on the current status, is critical to the recommendations for future research in this area.

3.5. Results

The results section describes the various techniques used in the *k*-hyperparameter tuning in high-dimensional spaces. Table 1 presented the summary results of the names of the state of the art *k*-hyperparameter tuning techniques, and a brief description of each technique including its strengths and limitations. Table 2 presented the data dimensionality reduction methods for each and every *k*-hyperparameter tuning technique. Table 3 presented the names and nature of the high-dimensional datasets used with the *k*-hyperparameter tuning techniques. The nature of the data set include: synthetic/real, number of features (*P*), number of records (*N*), categorical / numerical etc. Lastly, table 4 presented a summary of the algorithm's performance and statistical metrics as well as the specific metrics 'scores for each and every *k* hyperparameter tuning.

In Table 1, the name of the state of the art *k*-hyperparameter tuning technique is given with a short description of how it achieves the tuning process, including the name of the dimensionality reduction method used with each technique. The strengths and weaknesses of individual technique are also stated with the proposal for the limitation given to act as the foundation for future research work in this tuning problem.
In table 2, we present the name and the nature of the high-dimensional dataset used by each of the state of the art tuning techniques. The nature of the dataset encompasses the data type, the level of dimensionality, number of instances as well as the number of attributes / variables. The dimensionality of the dataset is the order of magnitude on the number of attributes on the dataset, usually measured in powers of ten.

In Table 3, we present the performance of each and every *k*-hyperparameter tuning technique based on the laid out statistical and performance metrics. It is however worth to note that these metrics scores are picked as reported by the authors.

4. RESULTS AND DISCUSSIONS

4.1. Results

Each of the result findings was aligned to the research questions in this study.

RQ1*: Which k-hyperparameter tuning techniques and data dimensionality reduction methods are used in in high-dimensional spaces?*

Table 2. Description of the *k*-hyperparameter tuning techniques, data dimensionality reduction methods used with these techniques as well as their tuning strategies and limitations.

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Description, tuning strategy and limitations	Data dimensionality reduction method
[8]	Onumanyi et., al. (2022)	AutoElbow	Elbow graph is normalized using lowest and highest values along the coordinates of both the ordinate and abscissa. The estimated elbow, i.e. the *k*-hyperparameter, is the point that maximizes the distance between each point on the graph to the minimum, maximum reference points as well as the "heel" of the elbow graph. Although the technique relatively well, it has the limitation of the fact that the auto-elbow graph may not depict a sharp elbow with some imbalanced high-dimensional datasets	Principal Component Analysis (PCA)
[47]	Yan et. al., 2019	Adaptive Multi-view Subspace Clustering for High-dimensional Data	This technique is an extension of the canonical *k*-means algorithm where feature learning mechanism is integrated in order to handle the high-dimensional space. Although the experimental results with the with four different datasets shows that the technique is relatively effective, the limitation with this *k*-hyperparameter tuning technique is the fact that handling high-dimensional space with heterogeneous features is challenging.	Principal Component Analysis (PCA)

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Description, tuning strategy and limitations	Data dimensionality reduction method
[15]	Tao et. al, 2018	An Intelligent Clustering Algorithm for High-Dimensional Multiview Data in Big Data Applications	This technique uses intelligent weighting *k*-means clustering approach to deal with the challenges of having to consider all features of a high-dimensional dataset with equal relevance. At first, the coupling degree between clusters is presented in the clustering model in order to increase the level of dissimilarity among the clusters. Several features are applied in the computation of the weighting distance function used to identify the objects' clusters. In the second step, swarm intelligence is used to minimize the sensitivity of the initial cluster centers and the weights of features via a global search. Clustering high-dimensional datasets with heterogeneous features is a draw back for this technique.	Principal Component Analysis (PCA)
[48]	Xia et. al, 2020	Ball *k*-means	The technique uses a ball to describe a cluster with the intention of minimizing the point-centroid distance calculation. Each cluster is separated into stable area and active area; active area is then subdivided into equal portions of annular area. This *k*-tuning technique uses the ball clusters and neighbor searching strategy along with a number of novel stratagems to lower the computations of the centroid distances. This technique's iteration time complexity drops to sub linear levels as progress on the iterations is made. This makes it more efficient. However, the performance degradation on some datasets with high distance computations is a draw back with this *k*-hyperparameter tuning technique.	Principal Component Analysis (PCA)
[32]	Wang et. al, 2019	Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data	In this *k*-hyperparameter tuning technique, an adaptive loss is created to give an adjustable cluster indicator computation approach in order to handle high-dimensional datasets that possess different distributions. In this technique, the feature selection processes as well as the clustering process are done simultaneously. However, with this *k*-hyperparameter tuning technique, excessive feature reduction on the original high-dimensional datasets degraded the quality of clustering results.	Principal Component Analysis (PCA)
[30]	Lu, 2019	Improved K-Means Clustering Algorithm for Big Data Mining under Hadoop Parallel Framework	At first, the data points' density is computed and each cluster contains center points whose density is not lower than the threshold and the supplied density range. The basic cluster is combined in relation to the distance between the two cluster centers while the points that are not divided into any clusters are divided into the clusters near to them. However, the process of selecting the initial cluster centers is a draw back with this technique.	Principal Component Analysis (PCA)
[31]	Xie et. al, 2019	Improving K-means clustering with enhanced Firefly Algorithms	In this *k*-tuning technique, two variants of firefly based algorithms are proposed. These include the inward intensified exploration firefly algorithm as well as the compound intensified exploration firefly algorithm. These variants are meant to solve the challenges of initialization sensitivity and trappings of the local optima on *k*-means. A matrix-based search parameters and dispersing strategies are used with the two firefly models in order to improve the capability of exploitation and exploration processes. The attractiveness coefficient with a randomized control matrix in the inward intensified exploration firefly algorithm model is first replaced in order to release it from the biological law constraints. The exploitation in the neighborhood is lifted from a one-dimensional to multiple dimensional search strategy with improved diversification in terms of search scopes, scales, as well as directions. A dispersing strategy in the compound intensified exploration Firefly Algorithm is employed to generate similar fireflies to new positions out of the close neighborhood in order to execute the global exploration process. In order increase the efficiency in the search process, a sufficient variance between fireflies is created. The strength of the use of firefly algorithms in the *k*-hyperparameter tuning process is the fact that firefly algorithms possess attraction movements that help swarm to subdivide into smaller groups, automatically. In this case, each smaller group swarms around one mode or a local optimum solution. However, the presence of redundant, noisy, and irrelevant features in this technique severely affects the model's performance. Future works must therefore address this.	Principal Component Analysis (PCA)
[45]	Rustam et. al,	Kernel	In this *k*-hyperparameter tuning technique, the popular	Kernel Principal

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Description, tuning strategy and limitations	Data dimensionality reduction method
	2019	Spherical K-Means and Support Vector Machine for Acute Sinusitis Classification	machine learning based Kernel Spherical K-Means as well as the Support Vector Machine have been utilized. The technique was improved through the modification of the inner product with the kernel function so as to make sure that the separation of the linear data in high dimensions is achieved and thereby enhance the performance of this technique. On the other hand, the support vector machine is a binary based classification strategy that assists in developing models that poses good generalization ability. The results, evaluated and compared on a number of datasets is a confirmation that this *k*-hyperparameter tuning technique is superior as compared to the baseline models. Both the clustering accuracy and the running time were higher as compared to the baseline models. However, the tuning process of the kernel parameters using the grid search method is a challenging task when handling dataset of high-dimensionality. For such relatively high-dimensional spaces, it would be important to apply more efficient search methods as opposed to the grid search.	Component Analysis
[6]	Hussain & Haris, 2018	K-means based Co-clustering Algorithm for Sparse, High Dimensional Data	The uniqueness and significance of this *k*-hyperparameter tuning technique is that it embeds the higher order statistics as well as the co-clustering strategies in its tuning process as opposed to just using them as an external distance measure. This technique presents a unified framework referred to as "*K*-means based Co-clustering Algorithm for Sparse, High Dimensional Data". The step for the initialization process is modified to incorporate several points representing cluster centers in a way that the within-cluster –points are near each other but far from the points in the other clusters. In an iterative process, the neighborhood based walk statistics is proposed as a semantic similarity technique for both the assignment as well as the re-estimation of the center. The results, on a number of standard datasets, demonstrate the effectiveness of this technique as compared to other baseline models and state-of-the-art improvements. However, the running time is high in cases where the dataset has a large number of clusters.	Principal Component Analysis (PCA)
[49]	Rezaee et. al, 2020	GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game	This *k*-hyperparameter tuning technique utilizes the strength of bargaining game modelling to cluster high-dimensional dataset. With this *k*-hyperparameter technique, the cluster centers (players) compete to attract as many objects as possible to their own cluster. The payoff for the players (cluster centers) is maximized after a successful bargaining with each other. For this reason, these centers keep moving from one position to another in such a way that they possess lower distances with the highest possible data as compared with other cluster centers. The evaluation process demonstrates that this *k*-hyperparameter tuning technique clusters a dataset with relatively higher accuracy as compared to the other baseline models or the state-of-the-art *k*-hyperparameter tuning techniques. However, the continuous movement of the cluster centers (players) is an iterative process that could be computationally expensive for a dataset with extremely high dimensionality.	Principal Component Analysis (PCA)
[18]	Chakraborty & Das 2020	Lasso Weighted *k*-means	The "Lasso Weighted *k*-means" is a *k*-hyperparameter tuning technique that applies L_1 regularization term with feature weights that triggers feature selection within the framework of sparse clustering. A simple block-coordinate descent type algorithm is developed with a time-complexity that resembles Lloyd's strategy with an aim of optimizing the proposed objective. At the same time, strong consistency of the LW-k-means procedure is established. This technique, validated on several datasets via a rigorous experimentation process, shows that this *k*-tuning technique is extremely competitive in performing high-dimensional clustering as compared to the other baseline models as well as the other state-of-the-art models. The scores of both the clustering accuracy and the computational time are relatively good as compared in the other techniques. However, deploying weights on datasets with extremely high dimensionality e.g. gene microarray	t-distributed stochastic neighbour embedding (t-SNE)

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Description, tuning strategy and limitations	Data dimensionality reduction method
			datasets, is a challenge. Therefore, the use of a fuzzy system to help in assigning a probabilistic interpretation of the weights of features could be worth investigating in future research works.	
[20]	Brodinová et. al, 2019	Robust and sparse *k*-means clustering for high-dimensional data	This technique incorporates a weighting function that paves way to an automated assignment of weights on every observation. A high weight on an observation means that a data point belongs to a cluster while low weight means that a data point is a potential outlier. To cope with noisy variables, an objective function referred to as a lasso-type penalty is applied. A framework for determining both the number of clusters, *k*-hyperparameter and variables based on a modified gap statistic is introduced. However, the process of applying weighting functions as well as updating the variable weights is repeated iteratively until there is stabilization of the variable weights. This could be computationally expensive for a data set with relatively high dimensionality, for example, the proteomic and genes based datasets. A technique for pre-processing such a high-dimensional dataset before being clustered could be a solution for this. At the same time, identifying outliers from the noisy variables is challenging as the noisy variables are assigned a weight of zero.	Principal Component Analysis (PCA)
[38]	Orkphol & Yang, 2019	Sentiment Analysis on Micro blogging with *K*-Means Clustering and Artificial Bee Colony	This technique uses frequency–inverse document frequency strategy selecting the important features from a micro blogging high-dimensional dataset. The dimensionality reduction is performed using the singular value decomposition method. In order to search for a global optimum, the popular artificial bee colony is applied in the determination of the best initial centroids. Silhouette internal validation index is then used to determine the optimal value of *k.* However, in order to use silhouette in the determination of the optimal *k*-hyperparameter, several *k*-values have to be supplied in order to compare the one that returns the best silhouette score. This makes the technique computationally expensive. At the same time, inconsistencies may occur where the silhouette generates the best score at an optimal *k*-value that is different from another internal validation index e.g. Dunn index or Davies Bouldin index. In this case, we recommend the use of an ensemble internal validation index, via boosting or bagging, whose components exercise equal sensitivity to the varied conditions present in the high dimensional datasets. The optimal *k*-hyperparameter value would then be reached at through the ensemble's voting scheme i.e. the one that is returned by most of the internal validation indexes.	Singular Value Decomposition (SVD)
[41]	Song et. al, 2021	A Fast Hybrid Feature Selection Based on Correlation-Guided Clustering and Particle Swarm Tuning for High-Dimensional Data	This technique utilizes an integrated three-phase hybrid system using correlation-guided clustering and particle swarm tuning. During the first and second steps, a filter mechanism and a feature clustering-based method, both of low computational costs, are developed to minimize the search space. During the third step, it finds an optimal feature subset using an evolutionary algorithm with the global mechanism for searching. However, this technique faces the drawback of the fact that for data with a huge number of samples, it faces the challenge of extremely high computational cost.	Hybrid
[39]	Dey et. al, 2020	The Sparse Min-Max *k*-Means Algorithm for High-Dimensional Clustering	The sparse Min-Max *k*-Means Clustering strategy reformulates the objective function of the Min-Max *k*-Means algorithm into a new form of weighted between-cluster sum of squares. The sparse regularization is imposed on these weights to make it useful in the clustering of high-dimensional space. The technique, however, degrades in its performance when noisy variables are present in a high-dimensional dataset.	Principal Component Analysis
[40]	Hozumi et. al, 2021	UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets	This technique utilizes the UMAP data dimensionality reduction method to convert the high-dimensional space into low dimensional space. UMAP is nonlinear and uses three assumptions i.e. a dataset has uniform distribution on Riemannian manifold, the Riemannian metric has a local constant, and the there is a local connection of the manifold. UMAP creates a graph representation of each of the original highdimensional space in form of a predefined *k*-dimensional weighted UMAP. This is accomplished in such a way as to	Uniform Manifold Approximation and Projection (UMAP)

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Description, tuning strategy and limitations	Data dimensionality reduction method
			minimize the edge-wise cross-entropy between the weighted graph and the original data. Lastly, the UMAP graph's *k*-dimensional eigenvectors are used to represent each of the original data space. However, this technique does not perform well on a dataset that possess complex non-linear structure or on a dataset that possess non-uniform density.	

RQ2*: What's the nature of the input datasets used with the k-hyperparameter tuning techniques in high-dimensional spaces?*

Table 3. Nature, description and dimensionality of the datasets used with the *k*-hyperparameter tuning techniques

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Dataset name and nature
[8]	Onumanyi et., al. (2022)	AutoElbow	Cleveland heart disease dataset, containing 13 features, 303 instances and the *k*-hyperparameter value is 5. This dataset is of the type text and dimensionality of 10^1.
[47]	Yan et. al, 2019	Adaptive Multi-view Subspace Clustering for High-dimensional Data	Caltech, Jaffe, handwritten and Yale datasets were used with this technique. Caltech contains 8677 images from 101 categories, Jaffe contains 213 samples and 10 classes, handwritten dataset contains 2000 samples and 10 classes while the Yale contains 165 gray images of 15 individuals. These datasets are of the type text and images with a dimensionality of 10^1 and 10^2.
[15]	Tao et. al, 2018	An Intelligent Clustering Algorithm for High-Dimensional Multiview Data in Big Data Applications	Multiple Features (Mfeat) dataset, Internet Advertisement data set, Spambase data set, Segmentation data set and Cardiotocography data set were used with this technique. Mfeat contains 2,000 objects and 649 features in 10 clusters. Internet advertisement dataset contains 2359 objects in 2 clusters and 1557 features. Spam base dataset contains 4601 objects in 2 clusters and 57 features. Image segmentation dataset contains 2310 objects in 7 clusters and 57 features while the cardiotocography contains 2126 objects in 3 clusters and 21 features. These datasets are of the type text and images with a dimensionality of 10^1, 10^2 and 10^3.
[48]	Xia et. al, 2020	Ball *k*-means	Four-class, svmguide1, codma, keg network, epileptic, birch and ijcnn are the datasets used in this technique. Four class dataset is of the size 862 with 3dimensions. Svmguide1 dataset is of the size 7088 and 5dimensions. Codrna dataset is of the size 59535 with 8 dimensions. Kegg-Network dataset is of the size 65554 with 28dimensions.Epileptic is of the size 11500 and 179 dimensions. Birch3 dataset is of the size 100000 and 2dimensions. Ijcnn dataset is of the size 141690 and 22 dimensions while RNA-Seq contains 20531 dimensions. These datasets are of the type text and images with a dimensionality of 10^1, 10^2, $10^3$and 10^4.
[32]	Wang et. al, 2019	Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data	Glass, breast, vehicle, Umist, Yale, WebKB and TD2 are the datasets used in this technique. Glass contains 6 clusters, 214 instances and 9 features. Breast contains 2 clusters, 699 instances and 10 features. Vehicle contains 4 clusters, 846 instances and 18 features. Umist contains 20 clusters, 575 instances and 644 features. Yale contains 15 clusters, 165 instances and 1024 features. WebKB contains 7 clusters, 814 instances and 4029 features. TD2 contains 10 clusters, 653 instances and 36771 features. These datasets are of the type text and images with a dimensionality of 10^1, 10^2, 10^3 and 10^4.
[30]	Lu, 2019	Improved K-Means Clustering Algorithm for Big Data Mining under Hadoop Parallel Framework	HIGGS data set containing 11 million records and 28 features. This dataset is of the type signal images and with a dimensionality of 10^1.
[31]	Xie et. al, 2019	Improving K-means clustering with enhanced Firefly Algorithms	Acute Lymphoblastic Leukaemia (ALL), Sonar, Ozone, Wisconsin breast cancer diagnostic data set (Wbc1), Wisconsin breast cancer original data set (Wbc2), Wine, Iris, Balance, Thyroid, E. coli, Drivface, Micromass, sensor, Human Activity, Skin Lesion, Mice Protein, and Libras are the datasets used with this technique. Sonar contains 60 features, 2 clusters, and 140 instances. Ozone contains 72 features, 2 clusters, and 196 instances. ALL contains 80 features, 2 clusters, and 100 instances. WBC1 contains 30 features, 2 clusters, and 561 instances. WBC2 contains 9 features, 2 clusters, and 683 instances. Wine contains 13 features, 3 clusters, and 178 instances. Iris contains 4 features, 3 clusters, and 150 instances. Balance contains 4 features, 2 clusters, and 576 instances. Thyroid contains 5 features, 3 clusters, and 90 instances. Ecoli contains 7 features, 3 clusters, and 150 instances. Drivface contains 6400 features, 3 clusters, and 81 instances. Micromass contains 1300 features, 5 clusters, and 180 instances. Sensor contains 128 features, 5 clusters, and 415

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Dataset name and nature
			instances. Human Activity contains 560 features, 2 clusters, and 600 instances, Skin Lesion contains 98 features, 2 clusters, and 660 instances, Mice Protein contains 77 features, 2 clusters, and 300 instances. Libras contains 90 features, 2 clusters, and 72 instances. These datasets are of the type text and images with a dimensionality of 10^1, 10^2, and 10^3.
[45]	Rustam et. al, 2019	Kernel Spherical K-Means and Support Vector Machine for Acute Sinusitis Classification	Acute Sinusitis Data which contains 4 features, 200 instances and 2 clusters of acute and non-acute sinusitis.
[6]	Hussain & Haris, 2018	K-means based Co-clustering Algorithm for Sparse, High Dimensional Data	M2, M5, M10, Cornell, Washington and Cora datasets have been used with this *k*-hyperparameter tuning technique. M2 – This dataset contains 20,000 news group documents from 20 different newspapers and has 2 clusters. M5 – This dataset is similar to M2 but with more number of categories and less number of documents as compared to those in M2. M10 – This dataset is similar to M5 but with more number of categories and less number of documents as compared to those in M5 and M2. Cornell – This dataset contains 195 documents containing 1703 words and in 5 clusters. Cora – This dataset contains 2708 documents, 1433 words with each document belonging to one of 6 classes. Washington – This dataset contains 230 documents and has five classes. These datasets are of the type text and with a dimensionality of 10^3.
[49]	Rezaee et. al, 2020	GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game	Australian Credit, Breast Cancer, Breast Wisconsin, Diabetes, Haberman's Survival, Heart Disease, Hepatitis, Ionosphere, Japanese Credit and Mammographic are the datasets used with this *k*-hyperparameter tuning technique. Australian Credit dataset contains 690 instances and 14 attributes. Breast Cancer dataset contains 569 instances and 30 attributes. Breast Wisconsin dataset contains 699 instances and 9 attributes. Diabetes dataset contains 768 instances and 8 attributes. Haberman's Survival dataset contains 306 instances and 3 attributes. Heart Disease dataset contains 303 instances and 13 attributes. Hepatitis dataset contains 155 instances and 20 attributes. Ionosphere dataset contains 351 instances and 34 attributes. Japanese Credit dataset contains 690 instances and 15 attributes. Mammographic dataset contains 961 instances. These datasets are of the type text with a dimensionality of 10^1.
[18]	Chakraborty & Das 2020	Lasso Weighted k-means	Brain, Leukemia, Lung cancer, Lymphoma, Wine, Coil_5, ORL_5, YALE_5, ALLAML, Appendicitis, SuCancer, Iris, Glass, Tae, Zoo, Cleveland, Leaf, Vowel, Ecoli, Hebaerman and the WDBC are the datasets used in this technique. Brain - Brain dataset has 5 clusters and contains 42 instances and 5,597 features Leukemia - Leukemia dataset has 2 clusters and contains 72 instances and 3,571 features Lung cancer – Lung cancer dataset has 2 clusters and contains 181 instances and 12,533 features Lymphoma - Lymphoma dataset has 3 clusters and contains 62 instances and 4,026 features Wine - Wine contains 13 features, 3 clusters, and 178 instances Iris - Iris contains 4 features, 3 clusters, and 150 instances Cleveland - Cleveland heart disease dataset, containing 13 features, 303 instances and the *k*-hyperparameter value is 5 Ecoli - Ecoli contains 7 features, 3 clusters, and 150 instances These datasets are of the type text and images with a dimensionality of 10^1, 10^3 and 10^4.
[20]	Brodinová et. al, 2019	Robust and sparse k-means clustering for high-dimensional data	Synthetic dataset was used in this technique. The synthetic dataset consists of 40 observations, 50 features and 3 clusters. This dataset is of the type text with a dimensionality of 10^1.
[38]	Orkphol & Yang, 2019	Sentiment Analysis on Micro blogging with K-Means Clustering and Artificial Bee Colony	Twitter dataset was used to evaluate this algorithm. This dataset contains 1,000 non-redundant tweets and a polarity of 228 positive tweets, 104 negative tweets, and 668 neutral tweets. This dataset is of the type text with a dimensionality of 10^3.
[41]	Song et. al, 2021	A Fast Hybrid Feature Selection Based on Correlation-Guided Clustering and Particle Swarm Tuning for High-Dimensional Data	Arrhythmia, SCADI, GFE, Prostate, MFD, Coil 20, Yale_64, Colon, SRBCT, WrapAR10P, Leukemia_Small, DBWorld, DLBCL, Drv_face, Leukemia_Big, CNS, Lung and Ovarian were used as the datasets with this technique. Arrhythmia contains 195 features, 452 samples and 16 clusters. SCADI contains 205 features, 70 samples and 7 clusters. GFE contains 301 features, 743 samples and 2 clusters. Prostate contains 339 features, 102 samples and 2 clusters. MFD contains 649 features, 700 samples and 10

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Dataset name and nature
			clusters. Coil 20 contains 1,024 features, 200 samples and 20 clusters. Yale_64 contains 1,024 features, 165 samples and 15 clusters. Colon contains 2,000 features, 62 samples and 2 clusters. SRBCT contains 2,304 features, 83 samples and 4 clusters. WrapAR10P contains 2,400 features, 130 samples and 10 clusters. Leukemia_Small contains 3,571 features, 72 samples and 2 clusters. DBWorld contains 4,703 features, 64 samples and 2 clusters. DLBCL contains 5,469 features, 77 samples and 2 clusters. Drv_face contains 6,400 features, 606 samples and 3 clusters. Leukemia_Big contains 7,128 features, 72 samples and 2 clusters. CNS contains 7,129 features, 60 samples and 2 clusters. Lung contains features 12,600, 203 samples and 2 clusters. Ovarian contains 15,154 features, 253 samples and 2 clusters. These datasets are of the type text and images with a dimensionality of 10^2, 10^3 and 10^4.
[39]	Dey et. al, 2020	The Sparse MinMax *k*-Means Algorithm for High-Dimensional Clustering	Brain, breast cancer, colon cancer, leukemia, lung cancer 1, lung cancer 2, lymphoma, prostate cancer, SRBCT and suCancer datasets have been used in this technique. Brain dataset has 5 clusters and contains 42 instances and 5,597 features. Breast cancer dataset has 2 clusters and contains 276 instances and 22,215 features. Colon cancer dataset has 2 clusters and contains 62 instances and 2,000 features. Leukemia dataset has 2 clusters and contains 72 instances and 3,571 features. Lung cancer 1 dataset has 2 clusters and contains 181 instances and 12,533 features. Lung cancer 2 dataset has 2 clusters and contains 203 instances and 12,600 features. Lymphoma dataset has 3 clusters and contains 62 instances and 4,026 features. Prostate cancer dataset has 2 clusters and contains 102 instances and 6,033 features. SRBCT dataset has 4 clusters and contains 63 instances and 2,308 features. SuCancer dataset has 2 clusters and contains 174 instances and 7,909 features. These datasets are of the type text and images with a dimensionality of 10^3 and 10^4.
[40]	Hozumi et. al, 2021	UMAP-assisted *k*-means clustering of large-scale SARS-CoV-2 mutation datasets	Global SARS-CoV-2 mutation dataset, Coil 20, Facebook network, original MNIST and Jaccard distance based MNIST were used as the datasets with the technique. Global SARS-CoV-2 mutation dataset contains 203,344 features, 203,344 instances and 6 clusters. Coil 20 contains 1,440 grey images, 20 different objects each with an orientation of 72. Each image is 128 x 128 with a total dimensionality of 16384. Facebook network contains 22,470 nodes with a feature size of the same amount. Original MNIST contains a sample of 70,000, 28 x 28 grey scale images with a dimensionality of 784. Jaccard distance based MNIST is similar to the original MNIST. These datasets are of the type text and images with a dimensionality of 10^3, 10^4. And 10^5.

RQ3: What are the key algorithm performance and statistical metrics for evaluating the existing k-hyperparameter tuning techniques in high-dimensional spaces?

Table 4. Performance and statistical metrics and scores of the *k*-hyperparameter tuning techniques

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Algorithm's performance & statistical metrics & scores
[8]	Onumanyi et. al, (2022)	AutoElbow	Clustering accuracy (100%). This has been computed as a percentage of the *number of clusters* generated to the *number of actual clusters (ground truth)*
[47]	Yan et. al, 2019	Adaptive Multi-view Subspace Clustering for High-dimensional Data	Normal mutual information (98.31), accuracy (99.83) as well as the purity(98.83).
[15]	Tao et. al, 2018	An Intelligent Clustering Algorithm for High-Dimensional Multiview Data in Big Data Applications	The Jaccard Coefficient (JC), Rand Index (RI) and Folkes Russe (FS) were used to evaluate this algorithm. Mfeat dataset – RI = 0.9586, JC=0.6820 and FS = 0.8116 Internet Advertisement data set- RI = 0.8179, JC= 0.7868and FS =0.8809 Spambase data set- RI = 0..5225, JC=0.5222 and FS = 0.7225 Segmentation data set - RI = 0.2297, JC=0.8047 and FS = 0.3750 Cardiotocography - RI = 0.3984, JC=0.5576 and FS = 0.5854
[48]	Xia et. al, 2020	Ball *k*-means	Run-time was used to evaluate this technique. Four-class dataset- 0.03 Svmguide1 dataset - 0.24 Codrna dataset – 3.15 Kegg Network dataset – 12.21 Epileptic dataset -15.58 Birch3 dataset -1.18 Ijcnn dataset –9.28 RNA-seq dataset - 500
[32]	Wang et. al, 2019	Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data	Accuracy and NMI were used to evaluate this technique. Glass dataset – Accuracy 49.53% of and NMI of 33.81% Breast dataset- Accuracy 95.57% of and NMI of 71.92% Vehicle dataset- Accuracy of 44.13% and NMI of 17.87%

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Algorithm's performance & statistical metrics & scores
			Umist dataset- Accuracy of 44.35% and NMI of 63.89% Yale dataset- Accuracy of 48.56% and NMI of 54.63% WebKB dataset- Accuracy 67.09% of and NMI of 16.72% TD2 dataset - Accuracy of 36.22% and NMI of 31.13%
[30]	Lu, 2019	Improved K-Means Clustering Algorithm for Big Data Mining under Hadoop Parallel Framework	Clustering accuracy and running time are the metrics used to evaluate this technique. Clustering accuracy – 98% Running time – 22 seconds
[31]	Xie et. al, 2019	Improving K-means clustering with enhanced Firefly Algorithms	Sum of intra-cluster distances, also called fitness scores, accuracy, sensitivity, specificity, and macro-average F-score are used as the performance metrics to evaluate the performance of this *k*-hyperparameter tuning technique. Fitness score- Acute Lymphoblastic Leukemia (293.53), Sonar (160.54), Ozone (514.11), Wisconsin breast cancer diagnostic data set Wbc1 (2280.8), Wisconsin breast cancer original data set Wbc2 (1092.1), Wine (456.78), Iris (130.24), Balance (1002.9), Thyroid (113.26), E. coli (257.63), Drivface (4849.4), Micromass (656.91), sensor (426,26), Human Activity (12785), Skin Lesion (5399.8), Mice Protein (2345.0), and Libras (466.05). Accuracy – Acute Lymphoblastic Leukemia (0.5137), Sonar, Ozone, Wisconsin breast cancer diagnostic data set Wbc1 (0.9147), Wisconsin breast cancer original data set Wbc2 (0.9693), Wine (0.9485), Iris (0.8818), Balance (0.8047), Thyroid (0.8235), E. coli (0.7739), Drivface (0.7687), Micromass (0.8582), sensor (0.8118), Human Activity (0.6436), Skin Lesion (0.7854), Mice Protein (0.7238), and Libras (0.7801). Sensitivity – Acute Lymphoblastic Leukemia (0.5187), Sonar, Ozone, Wisconsin breast cancer diagnostic data set Wbc1 (0.9056), Wisconsin breast cancer original data set Wbc2 (0.9667), Wine, Iris (0.8227), Balance (0.8038), Thyroid (0.8676), E. coli (0.6609), Human Activity (0.6303), Skin Lesion (0.7898), Mice Protein (0.6913), and Libras (0.8342). Specificity – Acute Lymphoblastic Leukemia (0.5087), Sonar, Ozone, Wisconsin breast cancer diagnostic data set Wbc1 (0.8990), Wisconsin breast cancer original data set Wbc2 (0.9667), Wine (0.9618), Iris (0.9113), Balance (0.8056), Thyroid (0.8676), E. coli (0.8304), Drivface, Micromass, sensor, Human Activity (0.6568), Skin Lesion (0.7800), Mice Protein (0.6913), and Libras (0.8342). Macro-average F-score – Acute Lymphoblastic Leukemia (0.5145), Sonar, Ozone, Wisconsin breast cancer diagnostic data set Wbc1(0.9092), Wisconsin breast cancer original data set (Wbc2), Wine, Iris (0.9295), Balance (0.8045), Thyroid (0.7539), E. coli (0.6992). Wilcoxon rank sum test -Acute Lymphoblastic Leukemia (1.18E-04), Sonar (1.07E-08), Ozone (2.88E-11), Wisconsin breast cancer diagnostic data set WBc1 (5.01E-13), Wisconsin breast cancer original data set Wbc2(3.10E-10), Wine (3.49E-08), Iris (1.00E-00), Balance (2.89E-05), Thyroid (2.02E-06), E. coli (2.15E-02), Drivface (3.44E-03), Micromass(3.32E-04).
[45]	Rustam et. al, 2019	Kernel Spherical K-Means and Support Vector Machine for Acute Sinusitis Classification	Clustering accuracy and running time are the two metrics used to evaluate this *k*-hyperparameter tuning technique. Clustering accuracy – 90% Running time – 0.03 seconds
[6]	Hussain & Haris, 2018	K-means based Co-clustering Algorithm for Sparse, High Dimensional Data	Accuracy, NMI, Sum of squared error, running time and the popular statistical t-test were used to evaluate the effectiveness of this technique. Accuracy – M2 (0.92), M5 (0.95), M10 (0.73), Cora (0.48), Cornell (0.63), Washington (0.66). NMI - M2 (0.53), M5 (0.91), M10 (0.69), Cora (0.28), Cornell (0.43), Washington (0.48). Sum of squared error - M2 (385), M5 (368), M10 (370), Cora (1960), Cornell (78.5), Washington (93). Running time (seconds) - M2 (0.05), M5 (0.08), M10 (0.36), Cora (0.72), Cornell (0.01), Washington (0.04). T-test – The technique is statistically significant with a 0.05 significance level on M2, M5, M10, Cora , Cornell and Washington datasets.
[49]	Rezaee et. al, 2020	GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game	F-measure, Dunn index, Rand index, Jaccard index, Normalized Mutual Information, normalized variation of information, measure of concordance and Wilcoxon signed rank test have been used to evaluate this algorithm. F-measure - Australian Credit (0.849), Breast Cancer (0.933), Breast Wisconsin (0.948), Diabetes (0.660), Haberman's Survival (0.518), Heart Disease (0.78), Hepatitis (0.711), Ionosphere (0.759), Japanese Credit (0.839), Mammographic (0.764). Dunn index - Australian Credit (0.059), Breast Cancer (0.076), Breast Wisconsin (0.134), Diabetes (0.100), Haberman's Survival (0.072), Heart

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Algorithm's performance & statistical metrics & scores
			Disease (0.2), Hepatitis (0.443), Ionosphere (0.085), Japanese Credit (0.381), Mammographic (0.25). Rand index - Australian Credit (0.867), Breast Cancer (0.893), Breast Wisconsin (0.890), Diabetes (0.322), Haberman's Survival (0.742), Heart Disease (0.779), Hepatitis (0.848), Ionosphere (0.641), Japanese Credit (0.864), Mammographic (0.757). Jaccard index - Australian Credit (0.863), Breast Cancer (0.996), Breast Wisconsin (1.000), Diabetes (0.974), Haberman's Survival (0.962), Heart Disease (0.510), Hepatitis (0.971), Ionosphere (0.672), Japanese Credit (0.970), Mammographic (0.809). Normalized Mutual Information - Australian Credit (0.42), Breast Cancer (0.670), Breast Wisconsin (0.756), Diabetes (0.481), Haberman's Survival (0.632), Heart Disease (0.24), Hepatitis (0.553), Ionosphere (0.097), Japanese Credit (0.45922), Mammographic (0.245). Normalized variation of information - Australian Credit (0.712), Breast Cancer (0.511), Breast Wisconsin (0.530), Diabetes (0.765), Haberman's Survival (0.633), Heart Disease (0.850), Hepatitis (0.594), Ionosphere (0.951), Japanese Credit (0.702), Mammographic (0.872). Measure of concordance - Australian Credit (1), Breast Cancer (1), Breast Wisconsin (1), Diabetes (1), Haberman's Survival (1), Heart Disease (1), Hepatitis (1), Ionosphere (1), Japanese Credit (1), Mammographic (1). Wilcoxon signed rank test - Australian Credit (), Breast Cancer (), Breast Wisconsin (), Diabetes (), Haberman's Survival (), Heart Disease (), Hepatitis (), Ionosphere (), Japanese Credit (), Mammographic ().
[18]	Chakraborty & Das 2020	Lasso Weighted k-means	Running time in seconds, clustering error rate, rand index and Normalized Mutual Information were used in the evaluation process. Running time in seconds - Brain (2.407632), Leukemia (1.008672), Lung cancer (1.542459), Lymphoma (1.542459), Wine (0.219742), Coil_5 (4.165497), ORL_5 (0.609416), YALE_5 (0.704548), ALLAML (1.008672), Appendicitis (2.421305), SuCancer (Missing), Iris (Missing), Glass (Missing), Tae (Missing), Zoo (Missing), Cleveland (Missing), Leaf (Missing), Vowel (Missing), Ecoli (Missing), Hebaerman (Missing) WDBC (0.510246). Clustering error rate - Brain (0.2254), Leukemia (0.0250), Lung cancer (0.2196), Lymphoma (0.0161), Wine (0.0549), Coil_5 (0.4031), ORL_5 (0.2800), YALE_5 (0.3455), ALLAML (0.2492), Appendicitis (0.1917), SuCancer (0.4770), Iris (Missing), Glass (Missing), Tae (Missing), Zoo (Missing), Cleveland (Missing), Leaf (Missing), Vowel (Missing), Ecoli (Missing), Hebaerman (Missing) WDBC (0.0748). Rand index - Brain (Missing), Leukemia (Missing), Lung cancer (Missing), Lymphoma (Missing), Wine (0.9339), Coil_5 (Missing), ORL_5 (Missing), YALE_5 (Missing), ALLAML (Missing), Appendicitis (Missing), SuCancer (Missing), Iris (0.9495), Glass (0.6983), Tae (0.6181), Zoo (0.8886), Cleveland (0.6677), Leaf (0.9477), Vowel (0.8598), Ecoli (0.7984), Hebaerman (0.6220) WDBC (Missing). Normalized Mutual Information – Brain (0.6263), Leukemia (0.8056), Lung cancer (0.3078), Lymphoma (0.9255), Wine (0.8267), Coil_5 (0.4092), ORL_5 (0.7610), YALE_5 (0.5828), ALLAML (0.4298), Appendicitis (0.2502), SuCancer (Missing), Iris (Missing), Glass (Missing), Tae (Missing), Zoo (Missing), Cleveland (Missing), Leaf (Missing), Vowel (Missing), Ecoli (Missing), Hebaerman () WDBC (0.6215).
[20]	Brodinová et. al, 2019	Robust and sparse k-means clustering for high-dimensional data	Clustering error rate (CER), TPR (True Positive rate) and FPR (False positive rate) were the three metrics that were used to evaluate the performance of this technique. When k=3, CER = 0.01 TPR = 1 FPR = 0 When k=4, CER = 0.03 TPR = 0.85 FPR = 0.15 When k=5, CER = 0.05 TPR = 0.95 FPR = 0.05
[38]	Orkphol & Yang, 2019	Sentiment Analysis on Micro blogging with K-Means Clustering and Artificial Bee Colony	Clustering error rate was used to evaluate this algorithm. CER – 0.191
[41]	Song et. al,	A Fast Hybrid Feature	Clustering accuracy in % and run-time in seconds are the two metrics used to

Refe-rence	Author (s) & Year	Name of the *k*-tuning technique	Algorithm's performance & statistical metrics & scores
	2021	Selection Based on Correlation-Guided Clustering and Particle Swarm Tuning for High-Dimensional Data	evaluate this algorithm. Arrhythmia- Clustering accuracy (67.68), Run time (31.723 seconds) SCADI- Clustering accuracy (89.68), Run time (3.314 seconds) GFE- Clustering accuracy (85.13), Run time (43.571 seconds) Prostate- Clustering accuracy (97.49), Run time (3.550 seconds) MFD- Clustering accuracy (99.40), Run time (94.522 seconds) Coil 20- Clustering accuracy (100), Run time (237.147 seconds) Yale_64- Clustering accuracy (79.52), Run time (35.169 seconds) Colon- Clustering accuracy (92.47), Run time (5.974 seconds) SRBCT- Clustering accuracy (100), Run time (8.716 seconds) WrapAR10P- Clustering accuracy (100), Run time (13.104 seconds) Leukemia_Small- Clustering accuracy (100), Run time (5.274 seconds) DBWorld- Clustering accuracy (9.757), Run time (4.571 seconds) DLBCL- Clustering accuracy (100), Run time (5.821 seconds) Drv_face- Clustering accuracy (98.23), Run time (72.400 seconds) Leukemia_Big- Clustering accuracy (100), Run time (6.119) CNS- Clustering accuracy (8.591), Run time (7.400 seconds) Lung - Clustering accuracy (98.01), Run time (26.316) Ovarian- Clustering accuracy (100), Run time (9.826 seconds)
[39]	Dey et. al, 2020	The Sparse MinMax k-Means Algorithm for High-Dimensional Clustering	Retained features, run-time and Dunn index have been used in the evaluation of this technique. Retained features – Brain (1,810), breast cancer (79), colon cancer (76), leukemia (148), lung cancer 1 (16), lung cancer 2 (5), lymphoma (717), prostate cancer (5,650), SRBCT (1,019) and suCancer (1,370). Dunn index - Brain (0.647), breast cancer (0.197), colon cancer (0.435), leukemia (0.621), lung cancer 1 (0.245), lung cancer 2 (0.548), lymphoma (0.616), prostate cancer (0.393), SRBCT (0.544) and suCancer (0.505). Runtime - Brain (1.386 seconds), breast cancer (7.712 seconds), colon cancer (0.341 seconds), leukemia (0.784 seconds), lung cancer 1 (5.305 seconds), lung cancer 2 (5.137 seconds), lymphoma (1.857 seconds), prostate cancer (4.293 seconds), SRBCT (1.126 seconds) and suCancer (2.918 seconds)
[40]	Hozumi et. al, 2021	UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets	Clustering accuracy and run-time are the three evaluation metrics used with this *k*-hyperparameter tuning technique. Clustering accuracy - Coil20 (0.853), Facebook network (0.786), original MNIST(0.919), Jaccard distanced-based MNIST (0.960), Global SARS-CoV-2 mutation dataset (0.617). Run-time - Coil20 (500 seconds), Facebook network (22,000 seconds), original MNIST(8,000 seconds), Jaccard distanced-based MNIST (25,000 seconds), Global SARS-CoV-2 mutation dataset (45,000 seconds)

4.2. Discussions

From the results, internal validation indexes, external validation indexes, clustering accuracy and the run times are the commonly used metrics for evaluating the existing *k*-hyperparameter tuning techniques. However, in unsupervised clustering, we note that internal validation indexes, as opposed to external validation indexes are the best metrics for use in assessing the quality of clusters of the high-dimensional datasets with unknown number of clusters. Internal validation indexes are based on the previous knowledge about a dataset while the internal validation indexes are based on the information intrinsic to the data alone [50]. Among the internal validation indexes, Silhouette Index, Davies Bouldin index, Calinski Harabsz index as well as the Dunn Index are the most commonly used metrics for evaluating the quality of clusters. This relates to the research done by that explains why the four are the most common internal validation metrics used in evaluation of clusters [50].

Chi-square and T-test are on the other hand the most commonly used statistical tests for hypothesis significance testing with the existing *k*-hyperparameter tuning techniques. In future empirical studies on the evaluation of the state-of-the-art *k*-hyperparameter tuning techniques, it is important to adopt a standard set of the performance and statistical metrics, highdimensional datasets as well as standard set of data dimensionality reduction methods. The datasets used with the existing *k*-tuning techniques are mainly of text and images data types of varying dimensionality levels while the principal component analysis is the popular data dimensionality reduction method applied across a number of the reviewed *k*-hyperparameter tuning techniques. In order to ensure fairness in the evaluation of the different *k*-hyperparameter tuning methods, it is important for the data scientists to use a common set of benchmark high-dimensional datasets during an empirical study. Such future empirical study on this work needs to focus on tuning the *k*-hyperparameter value from popular texts and images datasets of varying dimensionality via different unsupervised data dimensionality reduction methods, and analyze the results. This will inform the data scientists on the most suitable set of the *k*-hyperparameter tuning technique and the data dimensionality reduction method for a specific variety of a high-dimensional dataset. Such a tool box would be beneficial.

The results on the nature, description and dimensionality of the datasets used with the *k*-hyperparameter tuning techniques shows that text and images are the most popular type of the high dimensional datasets used with these techniques. The dimensionality of these high dimensional datasets ranges between 10^1 and 10^5 orders of magnitude, with 10^3 and 10^4 orders of magnitude being the most common. Genomics and proteomics datasets have their order dimensionality of 10^5 and above. Datasets with higher dimensionality showed increased run times. This was evident in the ball *k*-means algorithm when comparing the epileptic dataset and the RNA seq dataset where the latter's run time was higher. This can be alluded to the fact that the number of object to centroid point computations increases proportionally with the increasing dimensionality of a high dimensional dataset. The "Adaptive Multi-view Subspace Clustering for High-dimensional Data" technique performs relatively fast when handling Jaffe and Yale dataset. This is because the technique iteratively updates each cluster centroids in the embedded space as opposed to on the original high dimensional space. When the "Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data" technique is handling dataset with extremely high dimensionality, i.e. TDT2 and WebKB, it has demonstrated relatively faster speed because it performs in the reduced feature space as opposed to in the original feature space. Exploring strategies for reducing the dimensionality of a high dimensional dataset before clustering it is therefore of paramount importance when dealing with high dimensional statistics. However, too much reduction on the number of features degrades the quality of clustering when a few features on the original dataset are preserved.

The superiority of the firefly algorithms, demonstrated in its accuracy, sensitivity, specifity and F-score is ascribed to its strong capability of exploration and exploitation through its search strategies. At the same time, the automated subdivision coupled with global exploration and intensified neighboring lowers the probability of trapping at the local Optima. Any proposed solution for solving *k*-hyperparameter tuning problems should therefore adopt this as a critical success factor. The Wilcoxon rank sum statistical test result of higher than 0.05 for the firefly based algorithm with the IRIS dataset is a demonstration that firefly algorithms are more effective in the *k*-hyperparameter tuning problems in higher dimensional space as opposed to low dimensional space, compared to other baseline models. Across all the datasets used with the firefly algorithm, the fitness scores, accuracy, Fscore, sensitivity and specifity have a significant improvement when the numbers of features in the original dataset are reduced to a lower number, but with minimal information loss. If this reduction has a significant information loss in the original dataset, then these scores degrades due to the degradation of the clustering results.

In the "UMAP assisted k means" technique, the superior scores in the clustering accuracy and run time across a number of datasets are an evidence of high efficiency and stability of the uniform manifold approximation and projection data dimensionality reduction method as opposed to the principal component analysis and the t- distributed stochastic neighbour embedding (t-SNE). This is ascribed to the fact that uniform manifold approximation and projection possess the capability of preserving the global structure of a dataset including its between-points structure and distances. This makes it one of the best options in visualization and exploration and future models geared towards solving the *k*-hyperparameter tuning problems.

In the Sparse MinMax K-means algorithm for high dimensional clustering, all the datasets used with this technique possess a number of features that is greater than the number of instances. The evaluation scores on this technique with these datasets is an indication that this technique is a state of the art in the *k*-hyperparameter tuning of the high-dimensional datasets. However, mechanisms need to be put in place to assist in managing the challenges of noisy variables and redundant features in a high dimensional dataset. Using multiple kernels to handle such challenges could offer solution into this problem.

In the fast hybrid technique based on the feature selection on correlation guided clustering and particle swarm tuning, it is evident that the run times are proportional to the number of features in a dataset. For this reason, it is proposed that the adoption of an efficient data dimensionality reduction method with this technique has a significant effect on its performance. Choosing the most suitable data dimensionality reduction method for such high-dimensional datasets is therefore of paramount importance to data scientists. An empirical study of this technique with varied high-dimensional datasets and data dimensionality reduction methods will inform the data scientists of the best data dimensionality method for a particular variety of high-dimensional dataset while using this *k*-hyperparameter tuning technique.

In the sentiment analysis on micro blogging technique with *k*-means and artificial bee colony, the use of silhouette index only in the determination of the *k*-hyperparameter demonstrated limitations. This is because, inconsistencies may occur where the silhouette index generates the best score at a different optimal *k*-value than the one generated by a different internal validation index like Dunn index or Davies Bouldin index. In such a case, we recommend the use of an ensemble internal validation index, via boasting or bagging, whose components exercise equal sensitivity to the varied conditions within a high dimensional dataset. The optimal *k*-value generated through such an ensemble would be a achieved through the voting scheme i.e. the one that is returned by most of the internal validation indexes.

In the robust and sparse K-means clustering for high dimensional data, having to repeat the process of applying weighting functions and updating the variable weights iteratively until stabilization is attained would be computationally expensive for datasets with extremely high dimensionality. The evaluation results on this algorithm demonstrate a challenge in determining the sparsity parameter *s* for contaminated datasets of high dimensionality. An automated method of determining the sparse parameter value *s* in such high dimensionality datasets is critical to building effective high-dimensional K-means models using this technique.

In the "lasso weighted K-means" technique, the results show that the t-distributed stochastic neighbour and embedding (t-SNE) and principal component analysis (PCA) generate different qualities of clusters with the leukemia dataset. It is therefore important to perform further experiments on a number of other high dimensional datasets with an aim of establishing the best set of data dimensionality reduction methods for a specific variety of high dimensional dataset. At the same time, applying weights on dataset with relatively high dimensionality is challenging when using this technique. The use of automated fuzzy in giving probabilistic interpretation of the weights of features could be worth investigating in the future research work focusing on solving this kind of tuning problem. Lastly, it is a concern that the computations of some metrics in the table of results are missing. For example, the Normal Mutual Information for Cleveland dataset and SuCancer datasets are missing. It is important that all the metrics proposed in the methodology have their results computed in the table of results so that the discussion and conclusion process if fair.

In the GBK means algorithm, we note that the dimensionalities of all the 10 datasets used in the experimentation process are of relatively lower dimensionality as opposed to the datasets used with the other *k*—hyperparameter tuning techniques. It is important to undertake an empirical study using datasets of relatively higher dimensionality and analyze the results in order to have a clearer understanding of the performance of this technique.

In *k*-means based co-clustering algorithm for sparse high dimensional data, its running time shows that it increases when the number of clusters are large as is the case with the M10 and Cora datasets. To solve this challenge, developing parallel systems could significantly improve on the running time on the datasets with large number of clusters. These parallel systems would share the processing workload and subsequently improve on the running times. At the same time, this algorithm performs relatively lower particularly when the dataset is not well separated. For example, with the M2 and M5 datasets, the accuracy values of this algorithm is relatively lower, with other techniques performing significantly much better as compared to this technique. From the experimental results, the techniques that adopted the *k*-Means++ initialization strategy performed relatively better as compared to those that applied random initialization strategy. For this reason, it is important for future researchers focusing on this tuning problem to ensure that the initialization strategies adopted are oriented to finding the most approximate initial centers as opposed to doing this randomly.

In the kernel spherical *k*-means and support vector based clustering method for acute sinusitis, we note that the tuning of the kernel parameters using grid search is not efficient when dealing with dataset of relatively high dimensionality. For this reason, we propose the application of more efficient search methods and strategies that are able to handle high-dimensional spaces. Lastly, in all the *k*-hyperparameter tuning techniques, the correct approximation of the right cluster center goes along way with improving the general clustering performance.

5. CONCLUSION AND RECOMMENDATIONS

This review study presents a number of *k*-hyperparameter tuning techniques in high-dimensional space, data dimensionality reduction methods used with these techniques, their strengths and limitations, performance and statistical metrics as well as the names and nature of the high-dimensional datasets used with these techniques. From this study, it is evident that there is no single *k*-hyperparameter tuning technique that is universally able to return the optimal *k*-hyperparameter across all varieties of the high-dimensional datasets. Their performance is relative to the specific nature and variety of a high-dimensional dataset. This observation is in line with the "no-free-lunch" theorem in machine learning where an algorithm may perform well in one application area and not the other. The results of this study makes it efficient for data scientists and researchers to undertake empirical studies which subsequently aids in coming up with a solution to the *k*-hyperparameter tuning problem. In the future, we propose an in-depth empirical study and analysis on the best performing state-of-the art *k*-hyperparameter tuning techniques using a similar set of standard performance and statistical metrics, data dimensionality reduction methods as well as high-dimensional datasets. Such results will aid in the development of a data scientists' tool box that shows the most appropriate set of data dimensionality reduction method and *k*-hyperparameter tuning technique for a specific high-dimensional dataset. The results can also form the basis for improving an existing *k*-hyperparameter tuning technique or the development of a completely new and enhanced *k*-hyperparameter tuning technique extremely well in one application area and not the other. The results of this study makes it efficient for data scientists and researchers to undertake empirical studies which subsequently aids in coming up with a

solution to the *k*-hyperparameter tuning problem. In the future, we propose an in-depth empirical study and analysis on the best performing state-of-the art *k*-hyperparameter tuning techniques using a similar set of standard performance and statistical metrics, data dimensionality reduction methods as well as high-dimensional datasets. Such results will aid in the development of a data scientists' tool box that shows the most appropriate set of data dimensionality reduction method and *k*-hyperparameter tuning technique for a specific high-dimensional dataset. The results can also form the basis for improving an existing *k*-hyperparameter tuning technique or the development of a completely new and enhanced *k*-hyperparameter tuning technique.

REFERENCES

[1] M. Capó, A. Pérez, and J. A. Lozano, "An efficient K-means clustering algorithm for tall data. ," *Data Min Knowl Discov*, pp. 1–36, 2020.

[2] K. Chowdhury, D. Chaudhuri, A. K. Pal, and A. Samal, " Seed selection algorithm through K-means on optimal number of clusters," *Multimedia Tools*, 2019.

[3] J. Di and X. Gou, "Bisecting K-means Algorithm Based on K-valued Self-determining and Clustering Center Tuning. JCP, 13(6)," pp. 588-595., 2018.

[4] A. Dubey and A. P. D. A. Choubey, "A Systematic Review on K-Means Clustering Techniques," *International Journal of Scientific Research Engineering & Technology (IJSRET)*, pp. 2278–0882, 2018.

[5] M. K. Gupta and P. Chandra, "Pk-means: k- means using partition based cluster initialization method. Available at SSRN 3462549.".

[6] S. F. Hussain and M. Haris, " A k-means based co-clustering (kCC) algorithm for sparse, high dimensional data," *Expert Syst Appl*, vol. 118, pp. 20–34, 2019.

[7] H. Ismkhan, "Ik-means−+: An iterative clustering algorithm based on an enhanced version of the k-means. Pattern Recognition, 79," 2018.

[8] A. J. Onumanyi, D. N. Molokomme, S. J. Isaac, and A. M. Abu-Mahfouz, " AutoElbow: An automatic elbow detection method for estimating the number of clusters in a dataset. ," *Applied Sciences.*, vol. 12, no. 15, p. 7515, 2022.

[9] V. P. Murugesan and P. Murugesan, " A new initialization and performance measure for the rough k-means clustering. Soft Computing," pp. 1-15., 2020.

[10] T. Liu, S. Qu, and K. Zhang, " A Clustering Algorithm for Automatically Determining the Number of Clusters Based on Coefficient of Variation. ," *In Proceedings of the 2nd International Conference on Big Data Research*, pp. 100–106, 2018.

[11] A. R. Mamat, M. A. Mohamed, N. M. Rawi, and M. I. Awang, " Silhouette index for determining optimal k-means clustering on images in different color models. ," *International Journal of Engineering and Technology* , vol. 7, pp. 105–109, 2018.

[12] M. Mughnyanti, S. Efendi, and M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation. ," *In IOP Conference Series: Materials Science and Engineering). IOP Publishing*, vol. 725, no. 1, pp. 012–128, 2020.

[13] J. P. Ortega, N. N. A. Ortega, J. A. Ruiz-Vanoye, S. S. Sánchez, J. M. R. Lelis, and A. M. Rebollar, "A-means: improving the cluster assignment phase of k-means Big Data.," *International Journal of Combinatorial Tuning Problems and Informatics*, vol. 9, no. 2, pp. 3-10., 2018.

[14] C. D. Nguyen and T. H. Duong, " K- means**–a fast and efficient K-means algorithms. ," *International Journal of Intelligent Information and Database Systems*, vol. 11, no. 1, pp. 27–45, 2018.

[15] Q. Tao, C. Gu, Z. Wang, and D. Jiang, " An intelligent clustering algorithm for high-dimensional multiview data in big data applications. ," *Neurocomputing*, vol. 393, pp. 234-244., 2020.

[16] W. . . , and, '' Song, D. Li, Y. Ma, Wu, and D. Ji, "An enhanced clusteringbased method for determining time-of-day breakpoints through process tuning," *IEEE Access*, vol. 6, pp. 29241–29253, 2018.

[17] J. W. Harris and H. Stöcker, *Handbook of mathematics and computational science.* . 1998.

[18] S. Chakraborty and S. Das, "Detecting meaningful clusters from high-dimensional data: A strongly consistent sparse center-based clustering approach. ," *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 6, pp. 2894–2908, 2020.

[19] S. Sun, Z. Cao, H. Zhu, and J. Zhao, " A survey of tuning methods from a machine learning perspective. ," *IEEE Trans Cybern*, 2019.

[20] Š. Brodinová, P. Filzmoser, T. Ortner, C. Breiteneder, and M. Rohm, "Robust and sparse k-means clustering for high-dimensional data. ," *Adv Data Anal Classif*, vol. 13, pp. 905-932., 2019.

[21] S. Pandey and L. kumar Tiwari, "Review of Existing Methods in K-means Clustering Algorithm.," 2018.

[22] C. Patil and I. Baidari, "Estimating the Optimal Number of Clusters k in a Dataset Using Data Depth. ," *Data Sci Eng*, vol. 4, pp. 132–140, 2019.

[23] N. Sandhya and M. R. Sekar, " Analysis of variant approaches for initial centroid selection in K-means clustering algorithm. In Smart Computing and Informatics ," *Springer, Singapore.*, pp. 109–121, 2018.

[24] C. Yuan and H. Yang, "Research on K- value selection method of K-means clustering algorithm.," *J—Multidisciplinary Scientific Journal*, vol. 2, no. 2, pp. 226-235., 2019.

[25] G. Zhang, C. Zhang, and H. Zhang, "Improved K-means algorithm based on density Canopy. ," *Knowl Based Syst*, vol. 145, pp. 289-297., 2018.

[26] S. S. Yu, S. W. Chu, C. M. Wang, C. M. Wang, Y. K. Chan, and T. C. Chang, "Two improved k-means algorithms. ," *Applied Soft Computing*, vol. 68, pp. 747-755., 2018.
[27] S. Hess and W. Duivesteijn, "k is the Magic Number--Inferring the Number of Clusters Through Nonparametric Concentration Inequalities. ," *arXiv preprint arXiv:1907*, p. 02343, 2019.
[28] H. I. Hayatu, A. Mohammed, and A. B. Isma'eel, "Big Data Clustering Techniques: Recent Advances and Survey. In Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics ," *Springer International Publishing: Berlin/Heidelberg, Germany.*, pp. 57–79, 2021.
[29] S. Nawrin, M. R. Rahman, and S. Akhter, "Exploring k-means with internal validity indexes for data clustering in traffic management system," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 3, 2017.
[30] W. Lu, "Improved K-means clustering algorithm for big data mining under Hadoop parallel framework. ," *J Grid Comput*, vol. 18, pp. 239-250., 2020.
[31] H. Xie *et al.*, "Improving K-means clustering with enhanced Firefly Algorithms. ," *Appl Soft Comput*, vol. 84, p. 105763, 2019.
[32] X. D. Wang, R. C. Chen, F. Yan, Z. Q. Zeng, and C. Q. Hong, "Fast adaptive K-means subspace clustering for high-dimensional data. ," *IEEE Access*, vol. 7, pp. 42639-42651., 2019.
[33] P. Roy and J. K. Mandal, "Performance evaluation of some clustering indices," *In Computational Intelligence in Data Mining. Springer, New Delhi.*, vol. 3, pp. 509–517, 2015.
[34] J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen, "Comparison of internal clustering validation indices for prototype-based clustering. Algorithms," vol. 10, no. 3, p. 105, 2017.
[35] M. Hassani and T. Seidl, "Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. Vietnam Journal of Computer Science," vol. 4, no. 3, pp. 171-183., 2017.
[36] M. Jain, M. Jain, T. AlSkaif, and S. Dev, "Which internal validation indices to use while clustering electric load demand profiles ," *Sustainable Energy, Grids and Networks*, vol. 32, p. 100849, 2022.
[37] H. Xiong and Z. Li, "Clustering validation measures. ," *In Data Clustering. Chapman and Hall/CRC*, pp. 571–606, 2018.
[38] K. Orkphol and W. Yang, "Sentiment analysis on microblogging with K-means clustering and artificial bee colony. ," *Int J Comput Intell Appl*, vol. 18, no. 3, p. 1950017, 2019.
[39] S. Dey, S. Das, and R. Mallipeddi, "The Sparse MinMax k- Means Algorithm for High-Dimensional Clustering. ," *In IJCAI* , pp. 2103–2110, Jul. 2020.
[40] Y. Hozumi, R. Wang, C. Yin, and G. W. Wei, "UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets.," *Computers in biology and medicine*, vol. 131, p. 104264., 2021.
[41] X. F. Song, Y. Zhang, D. W. Gong, and X. Z. Gao, " A fast hybrid feature selection based on correlation-guided clustering and particle swarm tuning for high-dimensional data. ," *IEEE Trans Cybern*, vol. 52, no. 9, pp. 9573-9586., 2021.
[42] T. S. Babu, J. P. Ram, T. Dragicevi, M. M. c, F. Blaabjerg, and N. Rajasekar, "Particle swarm tuning based solar pv arrayreconfiguration of the maximum power extraction under partial shading conditions," *IEEE Transactions on Sustainable Energy* , vol. 9, no. 1, 2018.
[43] K. Peng, V. C. Leung, and Q. Huang, "Clustering approach based on mini batch kmeans for intrusion detection system over big data," *IEEE Access*, vol. 6, pp. 11897–11906, 2018.
[44] J. Pérez-Ortega, Almanza-Ortega N.N, A. Vega-Villalobos, R. Pazos-Rangel, C. Zavala-Díaz, and A. Martínez-Rebollar, "The K-means algorithm evolution. ," *Introduction to Data Science and Machine Learning*, 2019.
[45] Z. Rustam, J. Pandelaki, and A. Siahaan, " Kernel spherical k-means and support vector machine for acute sinusitis classification. ," *In IOP Conference Series: Materials Science and Engineering* , vol. 546, no. 5, p. 052011, Jun. 2019.
[46] Ruiz-Vanoye, Socorro Saenz S ´ anchez, Jos ´ e Mar ´, Rodr´ıguez Lelis, and Alicia Mart´ınez Rebollar, "A-means: improving the cluster assignment phase of k-means for big data.," *International Journal of Combinatorial Tuning Problems and Informatics*, vol. 9, no. 2, pp. 3–10, 2018.
[47] F. Yan, X. D. Wang, Z. Q. Zeng, and C. Q. Hong, "Adaptive multi-view subspace clustering for high-dimensional data. Pattern Recognition Letters," vol. 130, pp. 299-305., 2020.
[48] S. Xia *et al.*, " Ball k k-Means: Fast Adaptive Clustering With No Bounds. ," *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 1, pp. 87-99., 2020.
[49] M. J. Rezaee, M. Eshkevari, M. Saberi, and O. Hussain, " GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game. ," *Knowl Based Syst*, vol. 213, p. 106672, 2021.
[50] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *International Journal of computers and communications*, vol. 5, no. 1, pp. 27-34., 2011.

BIBLIOGRAPHY OF AUTHORS

Rufus Gikera is a Computer Science Lecturer at the School of Computing Sciences at Riara University Nairobi – Kenya. He holds Msc Computing degree of Strathmore University and a PhD Computer Science of Kenyatta University, Nairobi, Kenya. Area of Specialization: Machine Learning. Research interests: Computational Medicine, Computational Neuroscience, Computational Proteomics, Computational Genomics, Markov chain Monte Carlo.

Formerly a Professor of Computer Science at the University of Massachusetts, Jonathan Mwaura is currently a Professor of Computer Science at the Khoury College of Computer Science – Boston, USA. He holds a PhD Computer Science from the University of Exeter, United Kingdom. Area of Specialization: Machine Learning. Research interests: Evolutionary Computation, Multimodal Tuning & Robotics.

Elizaphan Maina is a Computer Science Lecturer at the Department of Computing at Kenyatta University Nairobi – Kenya. He holds a PhD Computer Science from the University of Nairobi, Nairobi – Kenya. Area of Specialization: Artificial Intelligent Systems. Research interests: Educational Data Mining & Machine Learning in Education.

Shadrack Mambo is the Dean School of Engineering & Technology at Kenyatta University Nairobi – Kenya and a Lecturer at the Department of Electrical & Electroncs Engineering at Kenyatta University Nairobi - Kenya. He holds a PhD in Electrical & Electronics Engineering from Universite Paris Est Creteil, Paris, France. Area of Specialization: Electronics Engineering. Research interests: Digital Image Processing & Digital Signal Processing.