Information Sciences 622 (2023) 178–210

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data

Abiodun M. Ikotun [a], Absalom E. Ezugwu [a,b,*], Laith Abualigah [c,d,e,f,g], Belal Abuhaija [h,*], Jia Heming [i]

[a] School of Computer Science, University of KwaZulu-Natal, Pietermaritzburg, KwaZulu-Natal, South Africa
[b] Unit for Data Science and Computing, North-West University, Potchefstroom, 1 Hoffman Street Potchefstroom, 2520, South Africa
[c] Prince Hussein Bin Abdullah College for Information Technology, Al Al-Bayt University, Mafraq 130040, Jordan
[d] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan
[e] Faculty of Information Technology, Middle East University, Amman 11831, Jordan
[f] Faculty of Information Technology, Applied Science Private University, Amman 11931, Jordan
[g] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia
[h] Department of Computer Science, Wenzhou-Kean University, Wenzhou, China
[i] College of Information and Engineering, Sanming University, China

ARTICLE INFO

Article history:
Received 5 March 2022
Received in revised form 18 November 2022
Accepted 27 November 2022
Available online 1 December 2022

Keywords:
K-means
K-means variants
Clustering algorithm
Modified k-means
Improved k-means
Perspectives on big data clustering
Big data clustering

ABSTRACT

Advances in recent techniques for scientific data collection in the era of big data allow for the systematic accumulation of large quantities of data at various data-capturing sites. Similarly, exponential growth in the development of different data analysis approaches has been reported in the literature, amongst which the K-means algorithm remains the most popular and straightforward clustering algorithm. The broad applicability of the algorithm in many clustering application areas can be attributed to its implementation simplicity and low computational complexity. However, the K-means algorithm has many challenges that negatively affect its clustering performance. In the algorithm's initialization process, users must specify the number of clusters in a given dataset apriori while the initial cluster centers are randomly selected. Furthermore, the algorithm's performance is susceptible to the selection of this initial cluster and for large datasets, determining the optimal number of clusters to start with becomes complex and is a very challenging task. Moreover, the random selection of the initial cluster centers sometimes results in minimal local convergence due to its greedy nature. A further limitation is that certain data object features are used in determining their similarity by using the Euclidean distance metric as a similarity measure, but this limits the algorithm's robustness in detecting other cluster shapes and poses a great challenge in detecting overlapping clusters. Many research efforts have been conducted and reported in literature with regard to improving the K-means algorithm's performance and robustness. The current work presents an overview and taxonomy of the K-means clustering algorithm and its variants. The history of the K-means, current trends, open issues and challenges, and recommended future research perspectives are also discussed.

* Corresponding authors.
E-mail addresses: Abiodun@ukzn.ac.za (A.M. Ikotun), Ezugwua@ukzn.ac.za (A.E. Ezugwu), babuhaij@kean.edu (B. Abuhaija), jiaheming@fjsmu.edu.cn (J. Heming).

https://doi.org/10.1016/j.ins.2022.11.139
0020-0255/

1. Introduction

Extracting meaningful and tangible information from collected data is the primary goal of data mining [4]. However, most data are collected in arbitrary forms and categories, making such data difficult to analyse, especially when the data objects' features are unknown. Appropriate organization of unlabeled data is an aspect of data mining handled by cluster analysis. The meaningful grouping of such unlabeled data is regarded as data clustering. The goal is to group unlabeled data so that the data objects whose characteristics and attributes are similar are together in a cluster such that the similarities of data objects within the same clusters are higher when compared with other clusters' data objects. In other words, data clustering analysis classifies unlabeled data to ensure higher intra-cluster similarity and lower inter-cluster similarity [59]. The process of clustering analysis can be likened to the learning process, which involves specific predictive behavior associated with unsupervised learning when handling unlabeled datasets [55]. Fig. 1 clearly illustrates this spectrum of different categories of learning problems of interest in pattern recognition and machine learning, as discussed in Jain [95].

Cluster analysis has been successfully applied to address data clustering problems in different domains such as medical science, manufacturing, robotics, the financial sector, privacy protection, artificial intelligence, urban development, aviation, industries, sales, and marketing [61,7,180,59,20,111,49]. Extracting useful information from data in these domains is essential for providing better services and generating more profits [181,148,172]. Real-world data generated are mostly voluminous, unlabeled, and of different dimensions. This makes data clustering difficult. Pre-identifying the number of clusters in a real-world dataset cannot be quickly done. Therefore, determining the optimal number of clusters in a real-world dataset characterized by high density and dimensionality is quite tricky for standard clustering algorithms. This poses a significant challenge to conventional clustering algorithms in which the number of clusters must be specified as input to the algorithm.

Algorithms for data clustering are grouped into two major categories [97,224,68,60], namely, hierarchical clustering algorithms and partitional clustering algorithms. Hierarchical clustering algorithms partition data objects into clusters in a hierarchical form either in a bottom-up approach (agglomerative method) or a top-down approach (divisive method). In the agglomerative method, individual data objects are merged iteratively based on their similarity. In the divisive method, the initial dataset is taken as a single cluster and broken down iteratively using data object similarity until each data object forms a single cluster or a set criterion is met. The hierarchical clustering algorithm produces a dendrogram of merged (agglomerative) or split (divisive) data objects depicting the corresponding cluster hierarchy generated as output for the cluster analysis [60]. The dendrogram is a pictorial representation of the data objects' nested grouping showing the similarity level at which each grouping changes [97].

In the partitional clustering approach, a single partition of the initial dataset is produced instead of a clustering structure of a dendrogram. Clusters are produced in a heuristic approach while optimizing a criterion function defined globally on all the data objects in the set or locally on the subset of the data objects [246,9,189]. Optimizing a criterion function on a set of the data objects using a combinatorial search of all possible values to get the optimum value is computationally prohibitive. Therefore, partitional clustering algorithms require the specification of different k values supplied at different runs to obtain the best configuration to produce the optimum clusters.

K-means clustering algorithm was proposed independently by different researchers, including Steinhaus [203], Lloyd [132], MacQueen [135], and Jancey [98] from different disciplines during the 1950s and 1960s [171]. These researchers' various versions of the algorithms show four common processing steps with differences in each step [171]. The K-means clustering algorithm generates clusters using the cluster's object mean value [197,34]. In the standard K-means algorithm, the cluster number is required as a user parameter and is used in the arbitrary cluster center selection from the dataset. However, the K-means algorithm may converge to a local minimum because of its greedy nature [95]. Therefore, it requires several runs for a given k value with different initial cluster center selections to obtain the optimal cluster result [243,59,19]. In addition, the standard algorithm detects ball-shaped or spherical clusters only because of the use of the Euclidean metric as its distance measure [95]. A typical K-means clustering process is illustrated in Fig. 2. With a set of input data supplied to the K-means clustering algorithm, the centroid vector $C = \{c_1, c_2, ..., c_k\}$ can easily be identified with K being the number of cen-

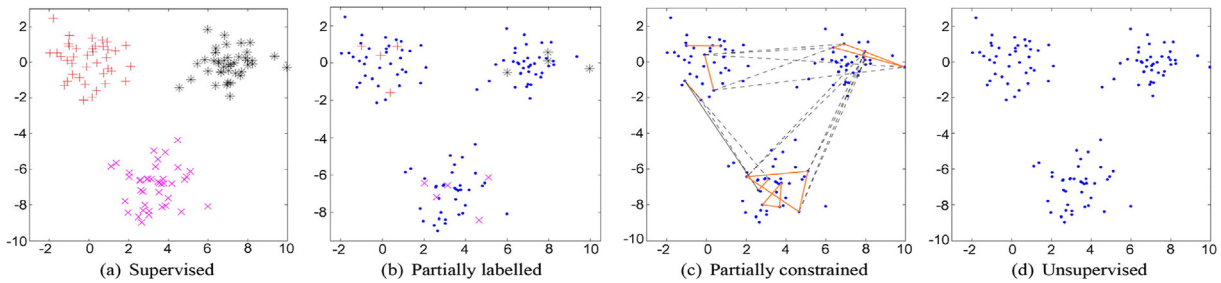

Fig. 1. Clustering analysis considered as a learning problem. The dots on the figure correspond to points without labels. In contrast, points with labels are denoted by plus signs, asterisks, and crosses. In (c), the must-link and cannot-link constraints are denoted by solid and dashed lines, respectively [120,95].

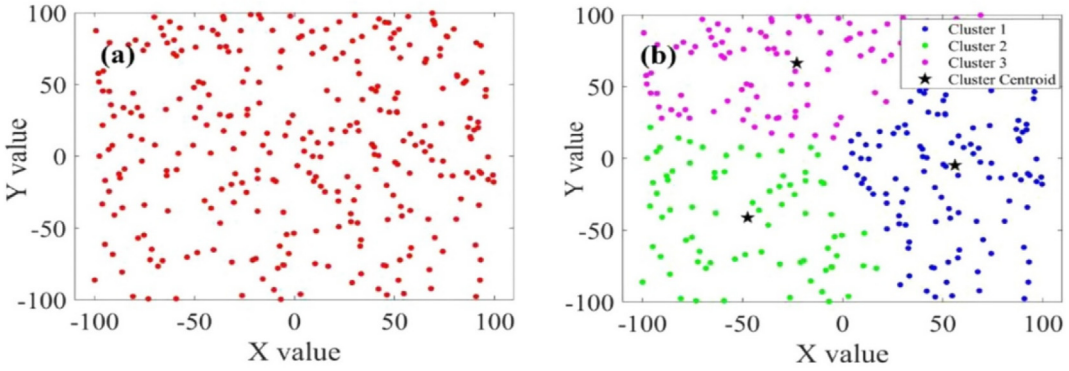

Fig. 2. K-means clustering: (a) randomly distributed datasets and (b) closest cluster centroid with three clusters [142].

troids defined by the user. Fig. 2a illustrates a data set in 2D space distributed randomly with $-100 \leq x_i, y_i \leq 100$, and Fig. 2b presents the K-means clustering result with the number of centroids set to $K = 3$.

Despite these limitations, the K-means clustering algorithm is credited with flexibility, efficiency, and ease of implementation. It is also among the top ten clustering algorithms in data mining [59,217,105,94]. The simplicity and low computational complexity have given the K-means clustering algorithm a wide acceptance in many domains for solving clustering problems. Several K-means clustering algorithm variants have been developed to enhance its performance. This work presents an overview of the K-means clustering algorithm and its variants with a proposed taxonomy for the variants. The algorithm's research progression from its inception, the current trends, open issues, and challenges with recommended future research perspectives are also discussed in detail.

In this paper, the following focal research question was proposed to reflect the purpose of this comprehensive review work:

"*What are the existing variants of K-means algorithms for solving clustering problems since its inception to date.*"

In providing answers to the main research question, the following sub-research questions were considered:

a) *Identify research that has been conducted to improve on the standard K-means clustering algorithm*
b) *What methods have been adopted in the various research found in (a) for improving the performance of the K-means clustering algorithm?*
c) *What are the performances of the reported K-means clustering algorithm variants*?
d) *What are the current research progressions involving the K-means clustering algorithm?*

This review work will be presented from four perspectives: first, a systematic review of the K-mean clustering algorithm and its variants. Second, a presentation of a proposed novel taxonomy of K-mean clustering methods in the literature. Third, verifications of the findings on all aspects of K-means clustering methods through an in-depth analysis. Fourth, an outline of open issues and challenges and recommended future trends. The main idea is to present a comprehensive systematic review that will provide current researchers and practitioners with a pathway for future novel research involving the K-means clustering algorithm. The main contributions of this research work are summarized below:

- A comprehensive review of the K-means algorithm is presented, including a proposed taxonomy of recent variants and trending application areas of the K-means clustering algorithm.
- Open research issues relating to adopting metaheuristic algorithms as automatic cluster number generators to improve the K-means algorithm's performance quality are identified and discussed.
- Finally, research gaps and the future scope of the K-means algorithm in general, particularly in outlining a new perspective for solving the challenges of the K-means clustering algorithm and its variants, are identified.

The rest of the paper is organized as follows: Section 1 introduces the background work on the proposed review study; Section 2 outlines the methodology approach; Section 3 presents a proposed taxonomy of k-mean clustering methods found in the literature, followed by a detailed discussion of the review of the K-means algorithm variants; Section 4 discusses the review findings; Section 5 reports the current trending areas of application of the K-means algorithm; Section 6 outlines the open issues and challenges of K-means clustering methods with recommended future trends; and Section 7 concludes the review.

2. Research methodology

This study aims at conducting a review of the K-means clustering algorithm variants. The research methodology adopted for the study is presented in this section. Kitchenham et al.'s [113] guidelines for a systematic literature review of computer

technology were adopted for the study. Four phases are involved in the review: planning, study search and selection, data acquisition, and data analysis. The planning section reported in section 1 includes establishing the problem statement, the study objectives formulation, the research questions, and the review protocol. The study 'search and selection' procedure phase includes the search keywords and the search queries. The search strategy with the selection criteria will be reported in this section under the corresponding sub-sections. The methodology steps used for the study are presented in Fig. 3.

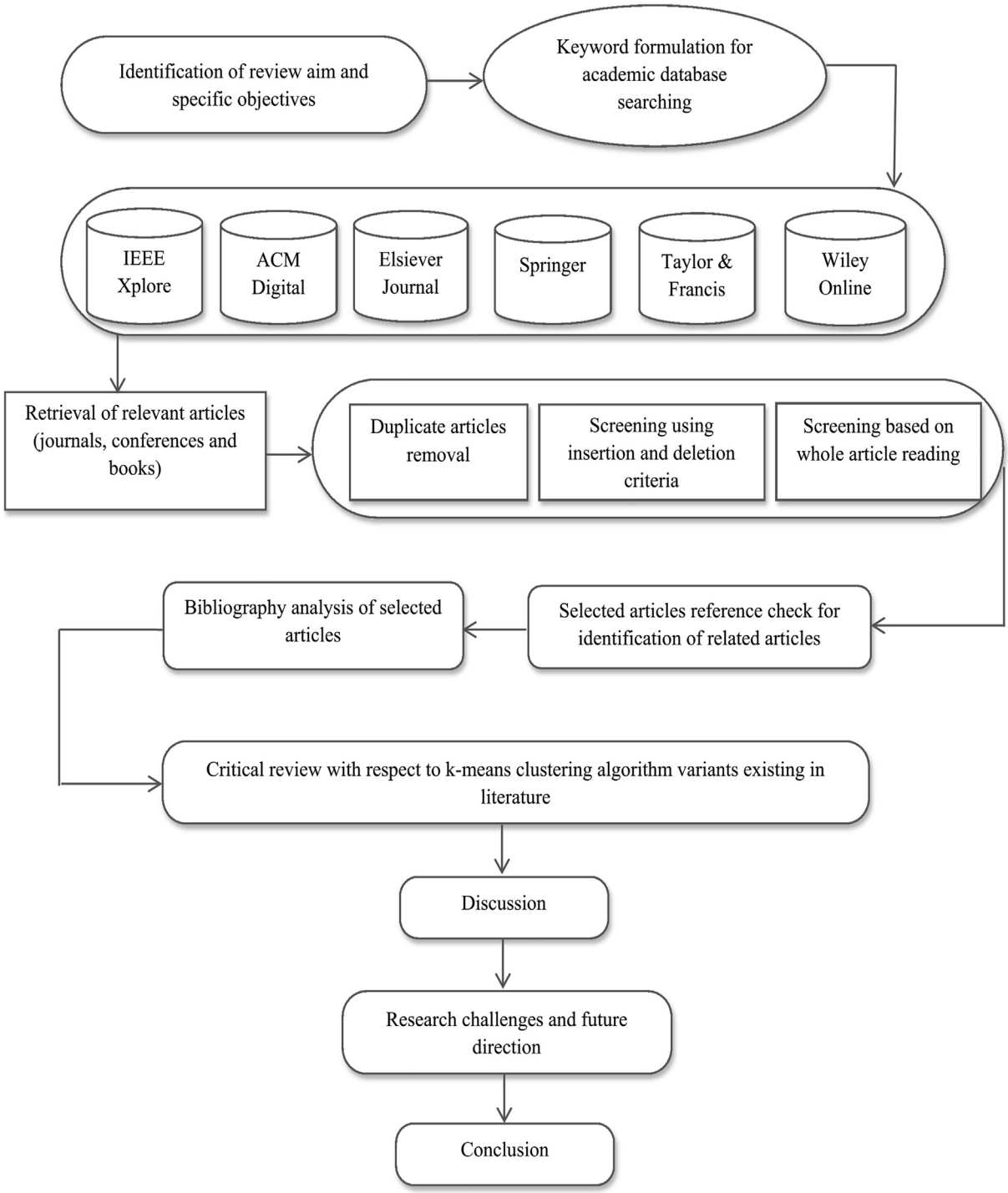

Fig. 3. The methodology processes used for the study.

2.1. Search strategy and keywords for identification of relevant literature

The K-means clustering algorithm is a viral and widely used clustering algorithm, with several publications reporting its application and enhancements. An electronic search was conducted on six major academic databases to find relevant literature for the study. Journal articles, conference proceedings, and edited books published from 1984 to 2021 were considered for the study, with major emphasis from 2010 after Jain's [95] comprehensive survey on clustering algorithms. Different keywords relevant to the study on the "K-means clustering algorithm and its variants" were defined in searching for relevant articles. The search keywords used included "K-means", improved K-means", "variants of K-means", "K-means variants", "modified K-means", "updates on K-means", "variations of K-means", "K-means taxonomy". Additional suitable keywords were generated using the synonyms of the existing keywords, such as "enhancements of K-means", "advances in K-means", "expanded K-means", "new developments in K-means", "innovations of K-means", "progressive K-means", "updated K-means". All English language-based articles were scrutinized to select the most appropriate ones for the study.

2.2. Search results

Relevant academic articles were searched for and retrieved from the six major academic databases listed in session 2.1 based on the queris formulated from the search keywords. A total number of 44 433 articles were retrieved. A thorough search of the stated academic databases was performed, and the results are presented in Table 1. Duplicate copies of retrieved articles were removed, and the remaining distinct copies were used for the subsequent screening.

2.3. Article screening and selection criteria

Three major screening exercises were conducted on the remaining distinct retrieved articles. The first screening involved reading the title, abstract, and keywords to select the most relevant articles, and 96 articles were found to be most appropriate for further screening. The selection criteria were applied to conduct the second screening exercise, and 21 articles were filtered out, leaving 77 articles for the final screening. In the final screening, the references of the selected articles were scrutinized to search for more relevant articles concerning the inclusion criteria. Six new articles were added. Finally, a total number of 83 articles were selected for the detailed study. The inclusion and exclusion criteria of article selection are presented in Table 2. These 83 articles were considered representative in reporting different variants on the K-means clustering algorithm from which the taxonomy was proposed. However, 19 articles from the Neural Information Processing System conference were added to supplement the reported variants extracted from the focused academic databases.

The distribution of the academic database from which the 83 articles were selected is shown in Table 1. Of the selected articles, 72 articles are from journals, six articles are conference proceedings, and 5 are book chapters. The selected articles were reviewed, and the K-means algorithm variants' taxonomy shown in Fig. 5 is proposed as reported in section 3.2. The improvements made to the standard K-means clustering have been grouped into two major categories: the variants based

Table 1
The search and screening results from the six databases.

Database	Result of Database Queries	1st Screening	2nd Screening	3rd Screening
ACM Digital Library	3 667	5	5	6
Elsevier Journal Access	8 464	22	15	20
Wiley Online Library	10 744	8	7	5
IEEE Explore	13 248	40	35	37
Springer	8 128	16	14	14
Taylor & Francis	182	5	1	1
	44 433	96	77	83

Table 2
Inclusion and exclusion criteria.

Inclusion	Exclusion
The main focus of this article is on K-means and its variants	Articles on other clustering algorithms
Articles that compare the standard K-means with the reporting variants	Articles comparing the K-means algorithm with other clustering algorithms
Articles reporting the performance of a variant of K-means	Articles reporting the performance of a variant of any other clustering algorithm
Articles comparing various variants of K-means	Articles reporting comparison between other clustering algorithms and K-means clustering algorithm variants
Published articles from conference proceedings, reputable peer-review journals, and edited books	Articles published as editorials, keynotes, speeches, abstracts and parts of textbooks
English language-based articles only	Relevant articles published in other languages.

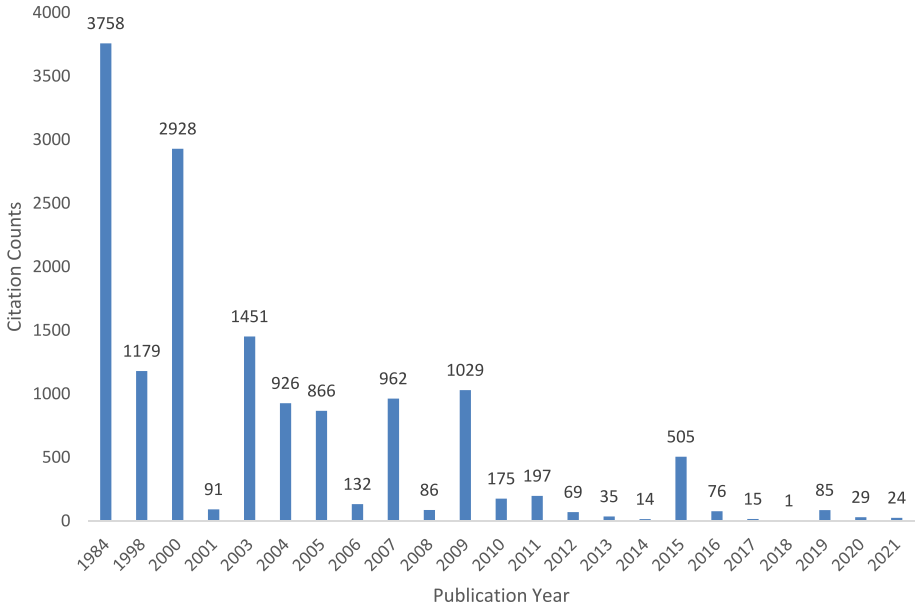

Fig. 4. Citation Count Per Year.

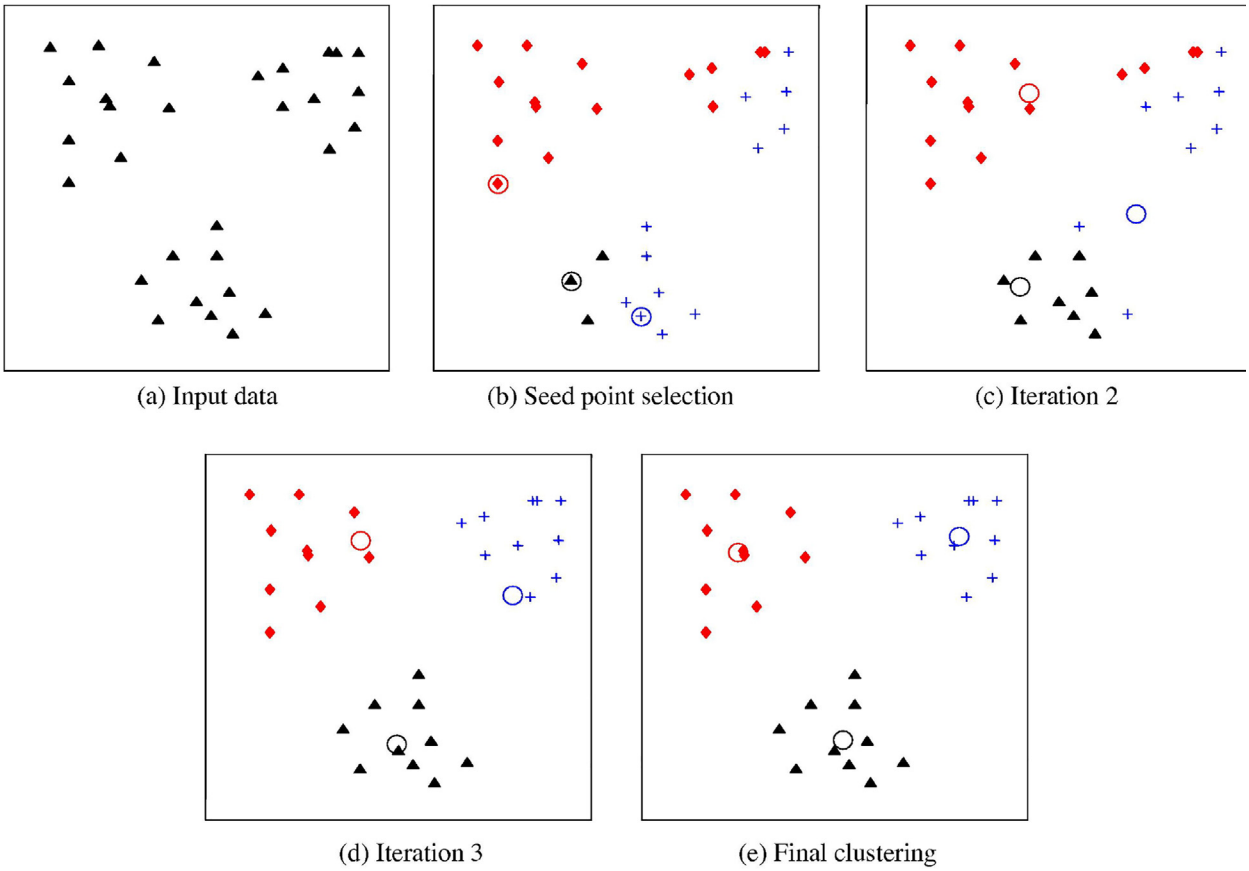

Fig. 5. Illustration of K-means algorithm. (a) Two-dimensional input data with three clusters; (b) three seed points selected as cluster centers and initial assignment of the data points to clusters; (c) and (d) intermediate iterations updating cluster labels and their centers; (e) final clustering obtained by K-means algorithm at convergence (figure taken from [95].

on the modifications to the standard K-means clustering algorithm, and the variants based on the standard K-means algorithm implementation. The standard K-means modification-based variants were further subdivided into four categories: algorithm input modification; algorithm processes modification; modifications for improved algorithm outputs; and algorithm concept modification. The implementation modification-based variants also have four categories: parallel machine implementation; quantum processes implementation; implementation using MapReduce framework; and, other implementations. Section 3.3 and 3.4 provide a detailed discussion of these variants.

The citation count per year is presented in Fig. 4. The vertical axis shows the citation count, and the horizontal axis shows the publication year. 1984 has the optimum yearly citation count, followed by the year 2000. However, high citation counts can be observed in 1984, 1998, 2000, 2003–2005, 2007, and 2009. There were some works of literature published in these years that received very high citation counts, for example, the works of Bezdek, Ehrlich, and Full [26], Huang [89], Kanungo et al. [102], Likas, Vlassis, and Verbeek [127], and Park and Jun [167].

2.4. Comparison with existing survey works

This section compares and discusses the differences between existing review papers and this review work. Many existing reviews of the K-means clustering algorithm concern its application in a specific domain, with few reviews being specific to the variants of the K-means clustering algorithm. The literature reviewed was minimal for those particular to the K-means clustering algorithm and variants. Steinley [205] presented a survey on the K-means algorithm which synthesized the results, methodology, and research conducted on the algorithm in the previous fifty years. It outlined the formulation of the minimum variance loss functions and alternative loss functions, as well as the various methods of specifying the number of clusters and initialization, data reduction schemes, and variable preprocessing methods. Hans-Hermann [81] presented a survey on the K-means algorithm with an historical view of the algorithm in respect of minimizing the sum of square distance measures used in the clustering process for both continuous and discrete variants.

Blömer et al. [28] presented a survey paper on the theoretical analysis of the K-means algorithm and several extensions for its run-time analysis and approximation quality. The scalability of the algorithm concerning big data was also considered. Pérez-Ortega et al. [171] conducted a systematic review of 79 articles on K-means algorithm improvements, classifying and summarizing them based on the algorithm steps, namely, initialization, classification, centroid calculation, and convergence. The reviewed articles include the recent trends in the improvements and the use in other areas. A taxonomy of the algorithm's improvement was not included.

Ahmed et al. [10] presented a simple taxonomy of K-means clustering algorithm variants under three headings: initialization, data types, and applications. This taxonomy is limited and does not cover improvements to the K-means clustering algorithm, such as parallel machine implementation, kernel-based design, and other recent innovations embedded in newer variants. The current systematic review will present a comprehensive review of 83 existing literature items, including state-of-the-art K-means algorithm variants, with extensive taxonomical analysis of the K-means algorithm variants.

Ahmad and Khan [9] reviewed a few K-means variants concerning mixed data clustering under the subsection related to partitional clustering algorithms. The number of variants discussed is limited to those articles that modified K-means for effective clustering of datasets of mixed data types. Table 3 presents a summary of previous surveys compared with the current study of K-means variants.

Table 3
Summary of previous surveys compared with this survey on K-means variants.

Reference	No of Literature Items Reviewed	The Range of Years Covered	Remarks
Stanley (2006)	n/a*	1956–2006	Elucidated the coalescence of information on the K-means algorithm over 50 years, summarizing the research on its performance with suggestions for future research.
Hans-Hermann [81]	n/a	n/a	Reported the origins, mentioning some of the critical K-means extensions.
Blömer et al. [28]	n/a	n/a	A survey on recent results of the K-means running time analysis and quality of approximation with other extensions to the algorithm. Provides insight into how the algorithm can be improved.
Pérez-Ortega et al. [171]	79	n/a	Classification and summary of various improvements to the K-means algorithm based on the algorithm steps.
Ahmad and Khan [9]	n/a	n/a	A taxonomy of the study of mixed data clustering with a few mentions of those involving K-means
Ahmed et al. [10]	27	1998–2019	The number of papers reviewed is minimal compared with existing literature on K-means variants but with a simple taxonomy of K-means variants.
This study	83	1998–2021	An up-to-date comprehensive review of existing literature, including state-of-the art K-means algorithm variants with a proposed extensive taxonomical analysis of the variants

*n/a – not available.

3. Standard K-means clustering algorithm

The K-means clustering algorithm is categorized as a partitional clustering algorithm. Partitioning given datasets into clusters involves finding the minimum squared error between the various data points in the data set and the mean of a cluster, then assigning each data point to the cluster centre nearest to it. Mathematically, given a dataset $X = \{x_i\} where i = 1, 2, \ldots n$ of d-dimension data points of size n,

X Is partitioned into 'k' clusters $C = \{c_j\} where j = 1, 2 \ldots k$

such that

$$J(c_k) = \sum_{x_i \epsilon c_k} ||x_i - \mu_k||^2$$

The K-means algorithm aims to minimize the sum of the square error for each k cluster. That is,

Minimize

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \epsilon c_k} ||x_i - \mu_k||^2$$

Initially, the K-means algorithm randomly chooses a specified k number of centroids from the dataset. Each data point's distance from all the selected centroids is evaluated, with each assigned to the closest centroid as a member of that centroid's cluster. On the assignment of a new member to a cluster, the center of a cluster is re-evaluated. This K-means algorithmic process is iteratively performed until the cluster membership is stable.

The basic steps in K-means algorithm are as follows:

- Initial partition selection with 'k' clusters
- Generate a new partition by assigning each pattern to the nearest cluster center
- New cluster center computation
- Repetition of (2) and (3) until there is stability in the cluster membership

The K-means algorithm is credited with implementation simplicity, and it has been used widely for clustering in many domains ([95]. According to Jain [95] and Drineas et al. [56], the objective function for the cluster formation for any given dataset involves minimizing the sum of square error, which is considered an NP-hard problem. The K-means algorithm may converge to a local minimum [95,10], implying that an increase in the number of clusters implies a decrease in the square error. This invariably implies that optimum results for the objective function can only be obtained for a fixed number of clusters. To avoid this problem, the algorithm is executed with different initial partitions (with randomly selected centroids), and the partition with the least square error is then selected for any specified value of k [95].

In implementing the K-means clustering algorithm, users are required to specify-three input parameters: k – the cluster number to be generated; the cluster centroids corresponding to the number of specified k, and the distance metric to be used. Choosing the appropriate value for k is critical in the K-means algorithm [95,30,2] which, in fact, is a major problem associated with the standard K-means clustering algorithm; the algorithm's performance depends on the specified value of k, with different values of k producing varying results [10]. Moreover, resultant clusters produced are also affected by initial centroid selection. The algorithm chooses initial centroids randomly, and different initial centroids produce different clusters due to the convergence of the K-means clustering algorithm, sometimes to a local minimum [2,95]. Thus, clusters depend on the initial cluster center choices [193,90].

Typically, for the distance calculation of data points from cluster centers, the standard K-means algorithm uses the Euclidean distance metric measure. K-means return spherical or ball-shaped clusters [95,199]. The K-means algorithm's convergence usually requires several iterations of repeating steps, and the accurate number of iterations cannot be determined beforehand. This impacts the algorithm's computational cost and becomes more pronounced with large data sets. The K-means algorithm has also been reported to be sensitive to outliers [199]. A limited number of outliers can significantly influence the mean value in a cluster. The presence of outliers as noise considerably affects the resulting clusters.

Most of the existing clustering algorithms, including K-means, were designed based on a framework that supports natural data classes that are either disjointed or fuzzy. However, in recent times, there are areas of clustering applications such as biology and information retrieval where there is a clear overlap in the natural classes of data [45]. The standard K-means algorithm clusters the data based on the properties of the data shared among the data objects, which naturally results in crisp clusters. As a result, K-means cannot effectively detect overlapping clusters, limiting its clustering performance on datasets in these application areas. The standard K-means clustering algorithm pseudocode is given in algorithm listing 1.

Algorithm 1: Standard K-means clustering algorithm pseudocode

```
Input:    Array X{x_1, x_2, ..., x_n} // Dataset to be clustered
Output:   k// Number of required clusters
          C{c_1, c_2, ..., c_k}// Cluster centroids
1.        A set of k clusters
2:        // Initialize Parameters
3:        X = {x_1, x_2, ..., x_n}
4:        C = {c_1, c_2, ..., c_k}
5:        Repeat
6:        //Distance calculations
7:          fori = 1 to n do
8:          forj = 1 to k do
9:            Compute the Euclidean distance from a data object to all cluster
10:         end j
11:         //Data object assignment
12:         Add data objects to the closest cluster
13:       end i
14:       //Update cluster centroid
15        Compute the new cluster centroid
16:       Until the difference between the cluster centroids of two consecutive iterations remains the same
          End
```

3.1. K-means computational complexity analysis

The time and space complexity analysis of the K-means algorithms is very significant in understanding the scalability of the algorithm's performance across different dimensional datasets and clustering problem variations. Various studies revealed that both the time and space complexities of the K-means and some of its common variant algorithms depend on the size of the input data n of the datasets [10]. Similarly, it is always very complex and challenging to find an optimal solution for the k-means algorithm in the Euclidean space for both the binary and multi-class clustering problems [102,164,154,155]. Moreover, according to Nazeer, Kumar, and Sebastian [155], the standard k-means algorithm has a computational time complexity of $O(nkl)$ because of the algorithm's iterative nature, where n is the number of data-points, k is the number of clusters and l is the number of iterations.

However, subsequent enhancement of the K-means algorithm, as presented in Yuan et al. [232], reveals that the first phase of determining the initial centroids for the K-means takes up to $O(n^2)$ time even though it produces better results than the classical K-means algorithm. More so, other variants of the K-means as proposed by Nazeer, Kumar, and Sebastian [155] derived a new time complexity for finding the initial centroids of a dataset containing n elements, of $O(n(kl + logn))$. Therefore, in summary, it can be assumed that the time complexity of all the K-means variants algorithms is in the order of $O(nlogk_{max})$, where k_{max} denotes the maximum number of clusters in a dataset, while the space complexity for all the variants is $O((n + k)d)$, and where d is the number of features in a dataset [10].

3.2. Taxonomy of K-means variants

Due to the wide acceptability and use of the K-means clustering algorithm and its easy implementation and simplicity, several algorithm variants have been proposed in the literature to address the known limitations of the standard algorithm. The algorithm has been modified in some literature to adequately address the clustering need in a specific domain. In other cases, the implementation technique has been considered to improve the K-means algorithm. Furthermore, hybridizing the standard K-means algorithm with other algorithms has been considered to enhance its performance. Based on these, the K-means algorithms variants are grouped into two major subsections: K-means algorithm design variants and K-means algorithm implementation variants. The various variants reflecting the various improvements made to the standard K-means algorithm are thus represented in the proposed taxonomy presented in Fig. 6.

3.3. K-means algorithm design variants

Most of the improvements made to the K-means algorithm to enhance its performance addressed the various limitations of the algorithm design discussed above. These improvements cut across the three major components of an algorithm: the input, the process, and the output. Some literature reported work on iteration reduction for the algorithm process enhancements, while others improved how data objects are assigned to clusters. Other works identified non-regular cluster shapes

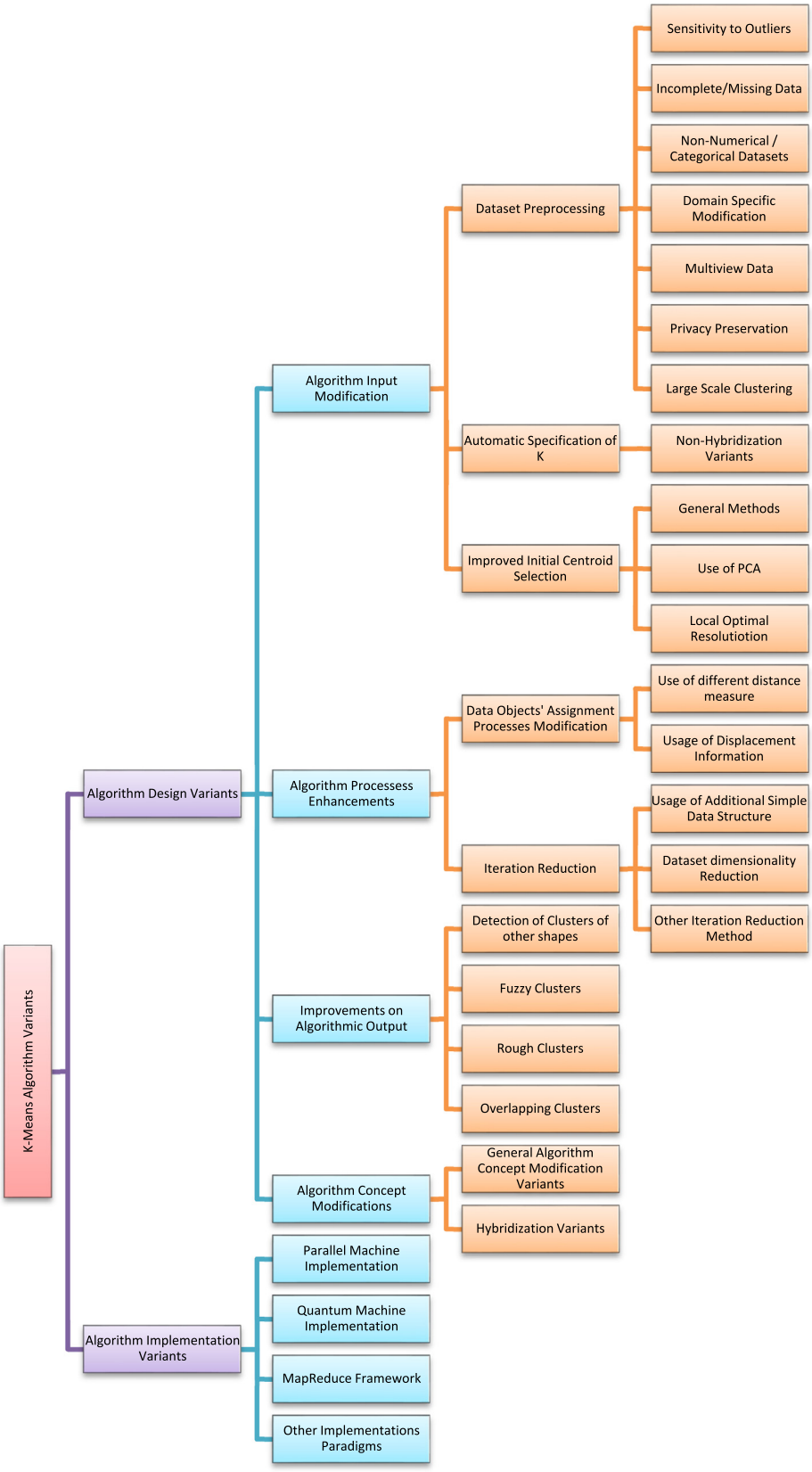

Fig. 6. Proposed taxonomy of K-means clustering algorithm variants.

such as rough, non-spherical, fuzzy, and overlapping clusters. The K-means algorithm has been combined with suitable existing algorithms to enhance the performance of K-means, that is, a hybridization technique has been applied in a bid to automatically generate values of k. Most metaheuristics algorithms were hybridized with standard K-means algorithms for solving mechanical clustering problems and the problem of local minimum convergence. In domain-based variants, attention has mostly been paid to preprocessing the domain dataset for proper clustering.

3.3.1. Algorithm input modification

For the K-means clustering algorithm input modifications, the variants reported in the literature can be subdivided into three categories: variants related to dataset preprocessing, variants addressing automatic specification of *k*, and variants dealing with initial centroid selection. These variants addressed the initialization problems of the standard K-means clustering algorithm. Dataset preprocessing positively impacted the K-Mean algorithm's performance, so some variants in this category performed some preprocessing operations on the data set. Some dataset preprocessing operations involved outlier detection and removal to address the problem of outliers in the data clustering.

a.) Dataset preprocessing

Outliers in datasets degrade data linearity and subsequently affect clustering accuracy [124]. According to Chawla and Gionis [37], the sensitivity of the K-means algorithm to outliers is exceptionally high, and it is essential to consider this when designing algorithms for the K-means objective [79]. Because of this, some of the variants of K-means have sought to address this issue [194,181,162]. In other cases, some datasets may be missing values, rendering them incomplete and adversely affecting the accuracy of the cluster output. Moreover, not all datasets are purely numerical. Some have mixed numeric data types with categorical features on which the standard K-means cannot be used. Some applications require domain-based data preprocessing on the dataset before efficient data clustering can be performed. To adapt the standard K-means to clustering these data sets, some variants have been developed which incorporate some data preprocessing activities on the dataset before eventual clustering. Also, there are other modifications such as the use of binary code for fast clustering [190] and feature ranking [240] in some of the variants for effectively handling large-scale datasets to avoid expensive computation and the high memory costs expected with large scale clustering.

Sensitivity to outliers

The presence of outliers in real datasets affects the optimal performance of the standard K-means algorithm. Olukanmi and Twala [162] state that outliers easily mislead the classical K-means clustering algorithm. Robust K-means algorithms focus on resolving the adverse effects of outliers on K-means clustering. There are several research works reporting on robust K-means clustering [86,63,123]. Georgogiannis [72] presented a theoretical analysis of the robustness and consistency of robust K-means.

Shrifan, Akbar, and Isa [194] propose a modification to the standard algorithm by combining Tukey rules with a new distance metric to improve the accuracy of the clustering algorithm. Outliers in the data set are removed adaptively using a modified Tukey rule in consideration of the distribution of the data either to the left or right or with consideration of the input data's mean value. The modified algorithm eliminates the outliers before finding the cluster centroids to reduce the effects of the outliers. Together with the proposed new distance metric, the centroid convergence and the clustering accuracy of the modified algorithm were significantly improved. Rathore and Shukla [181] proposed and implemented an enhanced K-means algorithm to handle a big data environment clustering problem. In their approach, the outlier points were removed to improve the data quality of the dataset, after which a bi-part method was used to perform the clustering operation. Normalization, outlier detection, and weighted-based methods were introduced to improve the standard algorithm's performance.

Wang and Su [214] presented another improvement to the K-means clustering algorithm through noise data filtering using a density-based noise data detection technique. Discovering noise data and filtering them is embedded into the standard K-means clustering algorithm. The filtering detects and removes noise before dataset clustering so significantly improves the algorithm's performance by producing a more accurate clustering output. K-medoids [167,106] are a variant of K-means for resolving the latter's sensitivity to outliers. In K-medoids, the representative objects are called medoids, representing the most centrally located object in a cluster instead of centroids, as is the case with K-means [167]. Park and Jun [167] proposed a fast and straightforward algorithm for K-medoids that is less sensitive to outliers but simple and efficient.

Chien, Pan, and Milenkovic [43] addressed the problem of approximate K-means clustering with outliers and side information of the same cluster queries and possible noisy answers. They applied a method that relies on the number of points needed for accurate centroid estimation and cluster size estimation based on the novel generalization of the double Dixie cup problem [158,54]. Their method performed approximate clustering and outlier identification simultaneously with order of magnitude performance improvements. Olukanmi and Twala [162] proposed a K-means-sharp that modifies the classical K-means centroid update step to avoid outliers when computing a new centroid. Outliers are detected automatically using a global threshold obtained from the distribution of the point-to-centroid distance in their modified algorithm. Their algorithm was able to detect outliers with high accuracy.

Incomplete or missing data

Considering that some datasets may be incomplete due to missing values, Honda et al. [85] introduced an extension to the principal component analysis-guided K-means (PCA-guided K-means) to address this problem. The estimation of the principal component scores is performed as an iterative process without inputting values, and has the capability of a rotated solution of cluster indicators serving as the initial centroid for K-mean clustering [5]. The authors employed the computa-

tionally efficient algorithm proposed by Shibayama [191] in calculating the principal components. Sarma et al. [187] proposed the leader K-means clustering method with a varying threshold (*lk*-means-CMVT), a prototype-based hybrid approach involving the partitioning of the dataset into varying small-sized partitions, each representing a prototype cluster using modified leaders clustering method. The modified K-means clustering algorithm is then used for partitioning the prototypes set into k number of clusters avoiding the possibility of empty clusters. Each prototype's corresponding set of patterns then replaces the various prototypes in the formed clusters of prototypes to generate patterns. A correcting step is embedded in the algorithm to prevent the generated clusters of patterns from deviating from the obtained partitions of the original K-means.

Huang et al. [88] proposed a robust deep K-means as a simple and effective method of clustering data to avoid the problem associated with the standard single-layer formulations that contain low-level features hindering data clustering based on complex hierarchical information of the dataset. Their proposed algorithm adopted deep learning techniques for extracting deep representations to improve clustering performance using the deep K-means model to learn hidden representations of the implicit lower-level attributes. Lithio and Maitra [131] presented K_m-means algorithm as an efficient variant of the K-means algorithm, which permits the clustering of a dataset with incomplete records. When the dataset has complete records, the algorithm is reduced to the standard K-means algorithm. The K_m-means algorithm is also equipped with initialization strategies and methods for estimating the number of clusters in the dataset. Marom and Feldman [139] proposed a K-means variant for clustering lines for big data. The problem of the K-mean variant arises when there are missing entries in some or all the input vectors or sometimes incomplete information in datasets. An example of this problem is typical in computer vision, where the position of a point or set of k points turn into lines based on their projections on 2D images through a pinhole camera model. In matrix approximation theory and data science, all possible values for missing entries in a database's record are considered, thereby turning a point into a line. The clustering process then considers lines that intersect around the K-means centers.

Non-numerical or categorical dataset

The standard K-means algorithm can only work on a numerical dataset, so some of the variants have sought to address this shortcoming. A modified version of the K-mean clustering algorithm was proposed by Ahmad and Dey [8] which can handle datasets with categorical features and mixed numeric data. They proposed a distance measure and a new cost function based on the co-occurrence of value which considers an attribute significance in the clustering process. The cluster center description of the standard K-means clustering algorithm was modified to bypass the 'numeric data only' feature to characterize clusters better.

To accommodate datasets with multiple, heterogeneous feature spaces, a framework that integrates multiple feature spaces was introduced by Modha and Spangler [145]. Their main goal was to equip the K-means algorithm with features to represent each data object as a tuple of multiple feature vectors, assign suitable distortion measures to feature space, and combine alterations on diverse feature spaces. Their proposed convex K-means was adapted to cluster fixed-weight data objects and determine optimal feature weighting that enhanced the clustering result.

Chen and Yang [39] presented diffusion K-means clustering, which aims to maximize the closeness of data objects within the same cluster based on diffusion distance. It comfortably clusters non-linear datasets and those with a non-Euclidean geometric features in mixed dimensions, such as is the case with manifold clustering. K-mode [89] is a simple but fast extension of the K-means algorithm for handling datasets with categorical attributes [75]. A simple matching dissimilarity measure is used to cluster objects with categorical attributes using modes of clusters instead of means with a frequency-based method for updating the modes during the clustering process. For data objects with mixed numeric and categorical attributes, K-means and K-mode algorithms were further integrated by Huang [89] to produce a k-prototypes algorithm that uses a dissimilarity measure that considers the mixed attributes.

Couto [47] explored the clustering of categorical data using kernel K-means. A new Hamming distance-based kernel function was proposed for embedding categorical data in feature space for clustering. In kernel methods, standard learning machine algorithms (that rely on inner products or distance metrics) are applied to data embedded into a feature space using a kernel function. Data are embedded into a feature space to capture and enhance the regularities and patterns in the data. After embedding the data, the clustering algorithm is used to detect the data's regularities in the feature space. According to Chen et al. [38], kernel-based K-means algorithms enable the handling of non-linearly separable datasets where data points are embedded into a high dimensional nonlinear manifold, and a nonlinear kernel distance function is employed to define their similarity. They proposed a fast kernel K-means that uses incomplete Cholesky factorization (InCF) to generate a low-rank full kernel matrix approximation version with linear K-means for faster kernel K-means with reduced memory space. Dinh, Huynh, and Sriboonchitta [53] proposed K-CMM (clustering mixed numerical and categorical data with missing value) using the mean and kernel-based methods for cluster formulation. The imputation and clustering steps are integrated into a single process for more efficient clustering. Information-theoretic-based dissimilarity measures and squared Euclidean were used for computing the distances between objects and cluster centers. Calandriello and Rosasco [33] proposed the Nyström approach to kernel K-means based on sampling a subset of training set points as an approximate kernel matrix. Their proposed method significantly reduced the computational cost while achieving the same accuracy as kernel K-means.

Domain-specific modification

Some dataset preprocessing is domain-based with specific operations relating to the characteristics of the dataset in the domain. Using an order-constrained solution, Krey, Ligges, and Leisch [114] presented domain-based (musical sound recording features) improvements to the standard K-means clustering algorithm to achieve clustering result stabilization and

improve the clustering's interpretability. Chen, Chen, and Lu [41] proposed the MK-means (multi-pass K-means) algorithm, a variant of the standard algorithm for handling dynamic network clustering. The clustering results of a current time are adjusted using the previous and subsequent time periods. Process adjustment back and forth ensures continual reduction of the temporal cost. Yang and Wang [227] improved the K-means algorithm for tag clustering based on latent semantic analysis using the min–max similarity technique with the most accurate and effective tag cluster result.

Qi et al. [177] introduced a modified K-means called text detector based on modifying K-means (K-text) by the law of universal gravitation with a mechanism for outlier detection and sufficient context information. The K-text algorithm can generate arbitrarily shaped texts with bounding boundaries of word level. It competently handles dense texts that have arbitrary shapes. The standard algorithm was used in most of the variants that concentrated on improving the standard K-means algorithm performance through dataset preprocessing. Most of the standard algorithm challenges are still present in the variants. These works do not address the peculiar problems of user dependence specification of cluster numbers as parameters and other K-means algorithm problems.

Multiview data

Yang and Sinaga [229] proposed feature-reduction multi-view K-means (FRMVK) for K-means-based clustering of multiview data (e.g., webpages, social network data) where different views give different representations. It uses a learning algorithm for automatic computation of individual feature weight and a schema for removing irrelevant features associated with a small weight for feature reduction.

Two-level variable weighting-K-means (TW-K-means) was presented by Chen et al. [40] as a clustering algorithm for multiview data characterized as an automated two-level variable weighting system. The algorithm computes weights for individual variables and views simultaneously to identify the variable's importance and the compactness of the views, respectively. These weights are then used in distance functions to determine an object's cluster. Intensive computation is not required for the two-step operations in the computation of the weights; hence the system is efficient for handling extensive high-dimensional multi-view data. Gönen and Margolin [74] proposed a multiple kernel learning algorithm that extends the kernel K-means clustering to the multiview setting. Their algorithm combined kernels on views in a local way to better capture sample-specific characteristics of the data.

Privacy preservation

The pervasiveness of computing devices contributes continuously to data collection and analysis growth. This data analysis has contributed immensely towards the growth of business, contributing positively to society in various fields. Despite this, storing and transmitting sensitive data poses serious concerns about data privacy [140]. Privacy preservation in general data mining addresses the issue of data privacy such that sensitive raw data are modified or trimmed out from the original database to avoid compromising personal privacy. It also addresses the concern of excluding sensitive knowledge mined from the database since such knowledge can equally compromise the privacy of data [209]. In this same vein, consideration for privacy preservation has been incorporated into extending the standard K-means. Biswas et al. [27] proposed a privacy-preserving approximate K-means clustering algorithm (PPK-means) where the input data is coded so that decoding back to accurate data is difficult while ensuring that the computational clustering result from the encoded data is relatively close to the result obtained from clustering the real data. Their K-means variant requires that the input data be coded in binary format to preserve the privacy of the data, and the algorithm is prevented from accessing the actual data vector during the computation. Their approach offered better performance for privacy preservation when compared with other baseline privacy-preserving clustering approaches. Stemmer and Kaplan [206] presented a differentially private K-means with constant multiplicative error as a privacy-preserving K-means algorithm in response to increased awareness and demand for user privacy. In privacy preservation, differential privacy ensures that the outcome of any analysis on a dataset is not affected substantially by the addition or removal of a database item.

Large scale clustering

The computational and memory cost for large-scale handling datasets is high when using the standard K-means. To address this challenge, some variants to the K-means algorithm have been proposed in the literature. Shen et al. [190] proposed a short variant of the K-means algorithm for large-scale clustering called compressed K-means (CKM). In CKM, the high-dimensional data are compressed into short binary code for fast clustering. There is a significant reduction in the storage requirement due to using binary code for data point representation. The use of Hamming metric between binary codes for the distance computation made the proposed algorithm very efficient. According to Zhang, Lange, and Xu [240], clusters are distinguished by only a few features in many realistic scenarios. Based on this, they proposed a simple and scalable sparse K-means clustering via feature ranking. Their variants addressed the problem of the curse of dimensionality using the idea of clustering under a sparsity assumption. Clustering under a sparsity assumption ignores irrelevant features in datasets to alleviate the memory and computational demands and improve clustering accuracy.

Shindler et al. [192] proposed fast and accurate K-means clustering of large datasets. Their algorithm addressed the clustering problem in a streaming model where a large dataset cannot be stored in the main memory and must be accessed sequentially. They incorporated an approximate nearest neighbor search for the computation of K-means with a better approximation factor and a faster worst-case running time. Ailon et al. [11] proposed a variant that optimizes the K-means objective in a one-pass streaming setting with a very lightweight memory requirement and computational cost without any assumption about data. Their algorithm, termed streaming K-means approximation, used a pseudo-approximation batch algorithm for K-means based on the new improvement on K-means++ (where more than k centers are allowed), with a streaming clustering algorithm with batch clustering performed on small inputs, which are then combined hierarchically.

Sculley [188] proposed a variant to the standard batch K-means clustering algorithm called a mini-batch K-means, characterized by low computational cost with excellent clustering results on large data sets. The mini-batch optimization for K-means clustering was adopted to lower stochastic noise and to reduce the computational cost when datasets grow large. Newling and Fleuret [156] proposed a nested mini-batch K-means to improve the mini-batch K-means further using the approach of distance bounding [57]. In their algorithm, each data sample contributes just once to centroids to avoid biased estimates caused by the unbalanced use of data. For the appropriate choice of mini-batch size, they balanced the premature fine-tuning of centroids with redundancy-induced slow-down. In Table 4, the summary of data preprocessing-based K-means variants is presented.

b.) Automatic specification of K

The standard K-means clustering algorithm requires the cluster number as a user parameter for the algorithm's execution. Determining cluster numbers is generally reported to be a complex problem in clustering [95]. For the K-means algo-

Table 4
Summary of data preprocessing-based variants.

Data Preprocessing Concerns	Reference	Methodology
Outliers Detection	Shrifan, Akbar and Isa [194]	Use of Tukey Rule
	Rathore and Shukla [181]	Outlier detection Module
	Wang and Su [214]	Noise data filtering
	Park and Jun [167]	Use of Medoids
	Chien, Pan, and Milenkovic [43]	Outliers Query
	Olukanmi and Twala [162]	Modified Centroid Update
	Olukanmi et al. [163].	Outlier awareness using the median of absolute deviation
	Zhao et al. [242]	Projected fuzzy K-means for handling noise and outliers
Missing Value/Incomplete Record	Honda et al. [85]	Extended PCA
	Sarma, Viswanath and Reddy (2013)	Prototype based clustering
	Huang et al. [88]	Deep learning techniques
	Lithio and Maitra [131]	
	Marom and Feldman [139]	Matrix approximation theory
Non-Numerical/ Categorical Dataset	Ahmad and Dey [8]	Co-occurrence of value
	Modha and Spangler [145].	Feature spaces integration
	Chen and Yang [39]	Diffusion distance
	Huang [89]	Matching dissimilarity measure
	Goyal and Aggarwal [75]	
	Couto [47]	Hamming distance-based kernel function
	Chen et al. [38]	Incomplete Cholesky factorization
	Dinh, Huynh and Sriboonchitta [53]	Mean and kernel-based methods with information-theoretic based dissimilarity measure
	Calandriello and Rosasco [33]	the Nyström approach to kernel K-means
	Sieranoja and Fränti [195]	Adapting K-means for graph clustering using K-algorithm and M−algorithm
Domain-based Data preprocessing	Krey, Ligges and Leisch [114]	Order constrained solution
	Chen, Chen, and Lu [41]	Process adjustment
	Yang and Wang [227]	Latent semantic analysis
	Qi et al. [177]	Law of universal gravitation
	Ben Gouissem et al. [22]	Grid-based K-means
	Abernathy and Celebi [1]	Partitional color quantization through binary splitting formulation
Multiview Data	Yang and Sinaga [229]	Feature reduction
	Chen et al. [40]	Two-level weighting System
	Gönen and Margolin [74]	Multiple kernel learning
	Lu et al. [133]	Multi-view clustering framework that uses discrete cluster assignment matrix and auto-weighted strategy for views
Privacy Preservation	Biswas et al. [27]	Input data coding
	Stemmer and Kaplan [206]	Use of constant multiplicative error
	Yang, Tjuawinata, and Lam [230]	The use of generalized differential privacy definition for local privacy of user's data.
	Zhang et al. [237]	Use of fully homomorphic encryption (FHE) for secure multi-party K-means
Large Scale Clustering	Shen et al. [190]	Data compression
	Zhang, Lange, and Xu [240]	Feature ranking
	Shindler, Wong, and Meyerson (2011)	Approximate nearest neighbor search
	Ailon et al. [11]	Pseudo approximation batch
	Sculley [188]	Mini batch approach
	Newling and Fleuret [156]	Nested mini batch

rithm, the choice of k is very critical. Therefore, the specification of an accurate number of clusters enhances the algorithm's performance [174]. Several variants have been proposed to determine the number of k for a given dataset automatically.

Non-hybridization variants

Saha and Mukherjee [184] proposed a variant of K-means called cluster number assisted K-means (CNAK), where k can be learned during the ongoing clustering process. Their idea is based on randomly sampled large-sized datasets having the same distribution as the original dataset, such that the number of generated cluster centroids from such a sample is approximately the same as the original dataset. The algorithm successfully detected a single cluster, identified cluster hierarchy, and clustered high-dimensional datasets. It demonstrated robustness in handling noise and cluster imbalance.

Sinaga and Yang [197] constructed an unsupervised learning schema for the K-means algorithm proposing unsupervised K-means (U-K-means) to free the standard algorithm from initialization without parameter selection with the ability to find an optimal number of clusters simultaneously. Sinaga, Hussain, and Yang [198] extended the U-K-means in their proposed entropy K-means with feature reduction. They employed several entropy-regularization terms in creating a learning schema for feature reduction so that the remaining essential features are then used in determining the number of clusters. Irrelevant features are eliminated, and the optimal number of clusters is automatically generated.

A density-based clustering approach, the modified K-means algorithm (MK-means), was introduced by Dashti et al. [51] and used in a guided kernel-based clustering algorithm. It allows clustering without specifying the cluster number beforehand, coupled with the flexibility of merging similar classes to improve the guided K-means clustering algorithm. In the guided K-means, the Euclidean distance is used to calculate the density, and the principal component analysis (PCA) is used to find the dense groups of objects; the PCA finds the appropriate number of object groups, and the information is passed as input into the MK-means.

X-means is a variant of K-means proposed by Pelleg and Moore [168], where the number of clusters is dynamically computed [116]. The algorithm efficiently searches the cluster location space to optimize the Bayesian information criterion (BIC) or the Akaike information criterion (AIC) measure, and dynamically updates the number of clusters in the process. In this way, X-means handles the standard algorithm's user-dependent specification of k and its poor computational problem. The user specified lower and upper bound values are used in the dynamical computation of the number of clusters [116].

Hamerly and Elkan [80] proposed Gaussians-means (G-means) as a variant of K-means. G-means uses a statistical test to decide if a K-means center is to be split into two to discover an appropriate k for any given dataset. The algorithm starts with a small number of K-means centers and adds more centers as appropriate at each iteration. K-means centers are split into two when their data appears not to come from a Gaussian distribution.

Harb et al. [82] proposed a new initialization method that dynamically finds the optimal cluster number in the dataset. The method works assuming that all the data sets are initially in the same cluster. Based on a one-way analysis of variance (ANOVA) model, this method uses dependence between measurements to determine the optimal cluster number from the dataset. Their improved algorithm was adapted to solve similarity aggregation in underwater wireless sensor networks (UWSNs).

c.) Improved initial centroids selection

Randomly selecting the initial centroid is another aspect of the initialization problem shared with the standard K-means algorithm. The standard algorithm randomly generates the initial centroid without considering the position of such centroids in the dataset, leading to unexpected convergence [10,66]. According to Fränti and Sieranoja [66], poor initialization causes the clustering operation to get stuck into a low local minimum. Considering this, many variants of the standard algorithm have concentrated on improving the process of selecting initial centroids.

General methods

Kant and Ansari [101] proposed an improved variant that uses the Atkinson index for initial centroid selection. The Atkinson index is used to measure inequality in the algorithm along with Euclidean distance to select the initial seeds accurately and efficiently. Arthur and Vassilvitskii [16] proposed K-means++, which uses specific probabilities in randomly choosing starting centers. A point p with the probability proportional to its contribution to the overall potential is selected as a center. K-means ++ has a very appealing simplicity and speed compared with the standard algorithm but with no guarantee for accuracy. Wei [216] further studied K-means++, focusing on the setting where the benchmark remains an optimal k-clustering but sampling more than k-clusters to improve the approximation. Makarychev, Reddy, and Shan [137] also studied K-means ++ and K-means ++ parallel, providing novel analyses for guaranteed improved approximation and bi-criteria approximation for the two algorithms. Their experimental result provided a better theoretical justification for the efficient performance of the algorithms in practice.

Ismkhan [94] proposed the K-means-+ (iterative K-means minus-plus) clustering algorithm as an improved version of K-means++. The K-means-+ iteratively enhances the standard algorithm's solution quality by eliminating a cluster (minus) from the set while dividing another one in each iteration, and the clustering process is repeated. Some methods to speed the process of determining which cluster is to be removed and which one is to be divided are embedded within the algorithm. K-means-+ performance doubles that of the standard algorithm and the K-means++.

This problem of the standard algorithm's sensitivity to the selected initial cluster center as well as the effect of outliers on cluster results prompted the proposal of K-means variants by Zhang et al. [239], which uses the Gaussian distance ratio in cluster approximation to set sample weight to be used for dissimilarity evaluation instead of setting weights of all features equally as is shared with the standard algorithm. The self-adaptive weight-based variant computes the weight for all data based on the current clustering state without relying on the initial cluster center and weight.

Zhang and Ma [238] also presented an improved variant that uses the Gaussian function with a weighted distance measure to compute the new center for each cluster instead of using the same weight for all the data objects. Mahmud, Rahman, and Akhtar [136] proposed a variant that heuristically finds better initial centroids given the computational expense. The strong dependence of cluster results on the selected initial centroids produces better cluster results with less computational time and higher accuracy. Mishra et al. [144] presented another variant with better clustering results and reduced computational complexity. In their algorithm, pairs of data points are evaluated to determine the pair with the longest distance apart. These points are selected as the initial cluster centroids, with data points added until a threshold for the maximum number of data points in a cluster is reached. The procedure is repeated for each cluster until the total number of clusters is K or $K-1$. Their proposed algorithm produced more accurate and efficient results when considering larger data sets with more attributes.

Likas, Vlassis, and Verbeek [127] presented another variant of the standard K-means clustering algorithm called the Global K-means algorithm. The algorithm is not dependent on any initial parameter [116]. Instead cluster centers are added dynamically-one at a time in an incremental manner using a deterministic global search procedure while the local search procedure is carried out using the K-means algorithm. Their algorithm also includes a computational load reduction method that does not significantly affect the resulting cluster quality. Fahim et al. [62] introduced the use of the selection rule to acquire a good candidate for the initial center, and erasure rules for deleting one or many unqualified centers further reduced the computation time of their variant. This variant is like the standard K-means algorithm in terms of the mean square error of the cluster produced but with a lower execution time.

Yuan et al. [232] proposed an improved variant of the standard algorithm with a better initial centroid selection. Their proposed algorithm evaluates the distances between every pair of data points to discover similar points consistent with the data distribution. The initial centroids are then selected from these data points. Mirkin [143] introduced intelligent K-means (iK-means), which employs data standardization such that the data points' center of gravity is considered a reference point. The anomalous pattern algorithm is then used to build clusters one by one, starting with the farthest data point from the origin. The process is reapplied to data points that are not clustered among the data points. Clusters that are too small, such as a singleton, are classified as outliers/noise and removed, while the centroid of the remaining clusters is taken as the initial K-means setting.

Vij and Kumar [210] proposed a K-means clustering algorithm variant that systematically determines the initial centroids when clustering a dataset. The variant considers a two-dimensional dataset with attributes that have negative and positive values as input. These values are then mathematically manipulated to generate the initial centroids. Gu [76] presented an approach for improving clustering accuracy through a novel Locality-Sensitive K-means based on Subtractive Clustering (LKSMSC). The proposed algorithm generates the initial cluster centers using subtractive clustering instead of a random selection of the standard algorithm to achieve a more stable clustering performance. In subtractive clustering, data objects with the highest mountain function values are selected as the cluster centers. In locality sensitivity, the data object's distance from the cluster center is used to calculate the object's weight parameter to describe its neighborhood structure. Qi et al. [176] proposed K*-means using a hierarchical optimization principle to initialize k* cluster centers where $k^{*}>k$ for reduction of the randomly seeded selection risk and used a top-n method for merging the nearest cluster based on the shortest n-edges in each iteration until k clusters are achieved.

K-median [108] is another variation of K-means clustering where the centroids of clusters are determined by calculating the median of the data objects instead of the mean. According to Jain and Dubes [96] and Bradley, Mangasarian, and Street [31], the use of median minimizes the overall cluster error when considering the 1-norm distance metric in place of the squared 2-norm distance metric of the K-means. K-median clustering has a better-formulated criterion function that generates more compact clusters than those generated using K-means clustering. Xiong et al. [222] proposed an improved K-means text clustering algorithm that optimizes the initial cluster centers to address the initial cluster center's sensitivity problem. Each data object's density is first calculated in this algorithm to determine isolated ones. A k number of high-density data objects with the most considerable distance is selected as the initial cluster centers. Newling and Fleuret [157] used Clarans – a K-medoids algorithm [159] for K-means initialization. Their proposed variant significantly reduced initialization and final mean square errors.

According to Bachem et al. [18], finding initial clusters is fundamental to obtaining high-quality K-means clustering results. They proposed a simple but fast seedling algorithm that provably produces good clustering results without assumptions on data. Their work is an improvement on the K-means++ seedling that does not perform well in the massive dataset and on the Markov chain Monte Carlo (MCMC) approach, which requires assumptions on the data generating distribution that may not hold.

The use of PCA for cluster centroids

Recent discoveries that the global solution for cluster centroids of a standard algorithm lies in the subspace of PCA [3,226,135,167]. Xu and Tian [224] proposed an adequate search for the standard algorithm using the PCA as a guide in the search. The PCA has a smaller subspace than the original dataset. The PCA-guided process has been more robust in handling initialization problems [85,52].

Ding and He [52] proved the authenticity of the principal components as the indicators for a continuous solution to discrete cluster membership for standard clustering. A new lower bound for K-means objective functions was derived by sub-

tracting the eigenvalues of the data covariance from the total variance. The new bounds were within 0.5–1.5% of the optimal values. It provides an excellent initial guess of centroid values for the K-means algorithm. Min and Siqing [141] proposed another variant that finds the initial cluster center using the adaptive searchability of the Genetic algorithm. They further adopted Li and Qin's [126] method for reducing the impact of isolated points in the dataset.

Resolution of local optimal convergence

As stated earlier, the standard algorithm can get stuck in the local optima of the objective function. Some literature focuses on addressing this problem of sensitivity to initialization and convergence to the local minimum of the standard algorithm. Zha et al. [234] proposed a spectral relaxation for K-means clustering to realize optimal global solutions. In their algorithm, the K-means minimization of the sum of squares cost function was reformulated as a trace maximization problem associated with the Gram matrix of the data vectors. A global optimum clustering solution was obtained by computing partial Eigen decomposition of the Grams matrix, a relaxed version of the trace maximization problem. The computation of a pivoted QR decomposition of the eigenvector matrix was used in assigning data objects to the appropriate cluster. Feng et al. [65] proposed a maximum triangle rule based variant called K-means xaximum triangle rule (KMTR) to solve the problem of optimal local convergence of the standard algorithm. It is believed that the consistency of the spatial distribution of the chosen cluster center and dataset can be guaranteed to some extent. Therefore, the rule was adopted to select the appropriate initial cluster center. There was a significant improvement in the algorithm's performance, and the local optimum convergence problem was solved. Lee and Lin [121] extended the work reported by Fahim et al. [62] to user selection and erasure rules to produce an accelerated K-means clustering algorithm. Vijayaraghavan, Dutta, and Wang [211] presented a natural notion that captures practical instances of Euclidean K-means to define stable instances with unique optimal K-means solutions. Their proposed algorithm recovers optimal clustering for additive perturbation stable instances. Their algorithm is designed to be robust to outliers in some cases where instances have additional separation. Table 5 summarizes the various initialization improvement approaches for the K-means variants.

Table 5
Summary of initialization improvement variants.

Area of Improvement	Reference	Method
Automatic Specification of K	Saha and Mukherjee [184]	Learning K using a randomly sampled dataset
	Sinaga and Yang [197]	Unsupervised learning schema
	Sinaga, Hussain and Yang [198]	Learning schema using entropy-regularization terms
	Dashti et al. [51]	The use of PCA
	Pelleg and Moore [168]	Optimization of BIC or AIC
	Hamerly and Elkan [80]	Statistical text on cluster centers
	Harb et al. [82]	One way analysis of variance
Improved Initial Centroids Selection	Kant and Ansari [101]	Atkinson index
	Arthur and Vassilvitskii [16]	Datapoint's probability contribution
	Wei [216]	Sampling more than k centers
	Ismkhan [94]	Cluster elimination and division
	Zhang et al. [239]	Gaussian distance ratio for dissimilarity evaluation
	Mahmud, Rahman, and Akhtar [136]	Heuristic approach
	Mishra et al. [144]	Pairs with the longest distance
	Likas, Vlassis, and Verbeek [127]	Dynamic adding of cluster center
	Fahim et al. [62]	Selection rule
	Yuan et al. [232]	Consistent similar points between pairs
	Mirkin [143]	Standardization of data for the center of gravity
	Vij and Kumar [210]	Using negative and positive attributes values
	Gu [76]	Locality sensitivity on subtractive clustering
	Qi et al. [176]	Hierarchical optimization principle
	[108]	Using the median of data objects
	Xiong et al. [222]	Optimization of initial cluster centers
	Newling and Fleuret [157]	The use of medoids
	Bachem et al. [18]	
The use of PCA for Cluster Centroids	Ding and He [52]	New lower bound for K-means objective function
	Min and Siqing [141]	Adaptive search of GA
	Zubair et al. [250]	Using PCA and division into percentiles for efficient initial coordinates for centroids
Resolution of local optimal convergence	Zha et al. [234]	Spectral relaxation
	Feng et al. [65]	Maximum triangle rule
	Lee and Lin [121]	Selection and erasure rule
	Vijayaraghavan, Dutta, and Wang [211]	Additive perturbation stability instances

3.3.2. Algorithm processes enhancement

The standard K-means algorithm is run multiple times to avoid convergence to local minimal. Many program iterations are carried out before convergence is achieved in the execution. The number of iterations cannot be determined beforehand, which invariably impacts the computational cost, and this becomes more pronounced with a large data set.

a.) Data objects assignment process modification

In order to reduce the number of program iterations required to avoid convergence into local minimal, the standard process of assigning data objects to clusters is modified in some of the K-means variants to improve the performance of the standard algorithm.

The use of different distance measures

Ichikawa and Morishita [91] presented a powerful heuristic variant for accelerating the large-scale dataset's standard algorithm clustering process. They used the Euclidean and Pearson correlation distances to quantify the dataset's similarities when assigning data points to clusters. They introduced a heuristic method based on an inherent property of Pearson correlation distance to prune redundant computation. The computation time was remarkably reduced. Since the standard algorithm uses the distance factor as the only constraint, there is sensitivity to particular data points. Geng et al. [71] proposed an improved K-means algorithm based on fuzzy metrics to address this challenge. The algorithm introduces a new constraint condition during the clustering process, specifying a new membership equation. A method for choosing the initial cluster center is also given to reduce the random cluster center selection problem. Their fuzzy entropy-based cost function constraints yielded an optimized clustering algorithm with Gaussian distribution.

Usage of center displacement information

Lai, Huang, and Liaw [118] presented fast K-means clustering using center displacement (FKMCUCD), a variant that uses center displacement information between successive partition processes to reject unlikely candidates when allocating data points to clusters. The number of distance calculations and computational time were reduced considerably. The algorithm's linear growth with the dataset dimension instead of the usual computational complexity of the k_d-based algorithms grows exponentially with the data dimension. The remarkable performance of this algorithm is noticeable with high-dimensional datasets and those with higher cluster numbers.

Lee and Lin [122] further worked on the algorithm Lai, Huang, and Liaw [118] proposed. They presented a faster variant using center displacement and norms product (CDNP) deletion. Their variant introduces faster locating of the nearest cluster in the dataset for each data point. The reduction in computational time was achieved through the use of the center displacement method and an erasure test at the second stage of the standard algorithm. The proposed algorithm was three times faster than the FKMCUCD [118], the ancestor method. Lee and Lin [122], in another of their variant algorithm, tried to accelerate the computational speed by combining their previous work [121] with Lai, Huang, and Liaw [118], adding the norms product test for identifying the impossible candidate to be deleted.

b.) Iteration reduction variants

According to Na, Xumin, and Yong [150], the method of clustering algorithm directly influences the clustering results. In the traditional K-means algorithm, the distance between each data object and all cluster centers is calculated in each iteration. This reduces the efficiency of the algorithm. Several options have been proposed in the literature to address this shortcoming of the standard K-means algorithm.

The use of additional simple data structure

Na, Xumin, and Yong [150] proposed a variant that reduces the number of distance calculations required in the standard algorithm, avoiding the repeated computation of the distance of each data object from the cluster center. Two simple data structures were maintained to keep each data object's cluster label and distance to the nearest cluster during each iteration for use in the subsequent iteration to identify which data object's distance from which cluster needs to be calculated. Data objects whose distances from new clusters are smaller than or equal to the distance of old cluster centers are exempted. Fahim et al. [62] proposed another variant with a lesser number of distance calculations and overall computation time. A simple data structure was used to store relevant information in a subsequent iteration. Based on the information, only the data points furthest from the new centroid are evaluated with other clusters and moved to the closest centroid.

Kanungo et al. [102] presented a variant referred to as a filtering algorithm that uses a K_d-tree data structure for the data points instead of the center points, thereby reducing computation since the data points do not vary throughout the computation. As the sizes of the dataset increase, the standard algorithm's cluster re-assignment process becomes prohibitively expensive. To address this issue, Wang et al. [212] proposed a variant that substantially reduces the computational complexity of the standard algorithm data object assignment step. Multiple random spatial partition trees are used to pre-assemble data objects into neighboring point groups to efficiently identify active points on or near clusters' boundaries. A closure for each cluster is constructed using the neighborhood information to minimize the number of cluster candidates to be considered when assigning a data point to the nearest one.

Dataset dimensionality reduction

Xie, Liu, and Wei [221] aimed to solve the K-means algorithm problem of low efficiency in big data clustering by using feature space as a dimension reduction technique for handling the dimensionality problem of big data. They adopted the spherical K-means model using the cosine dissimilarity between the data object features and the K-medoids model to choose the cluster center to achieve a faster and more accurate K-means clustering algorithm. In their proposed efficient K-means algorithm, Capó, Pérez, and Lozano [34] employed a recursive and parallel approximation to the standard algorithm using a small, weighted set of distributive representative points in the dataset instead of analyzing the entire dataset without com-

promising the quality of the approximation. The data objects in the weighted set are chosen from regions where determining the correct cluster assignment of the original instance is considered harder. They achieved a good algorithmic performance tradeoff between the quality of the solution obtained and the number of distance computations performed.

To solve the problem of the curse of dimensionality, the idea of integrating dimensionality reduction and clustering in a joint framework has been proposed in the literature. Discriminative clustering incorporates linear discriminant analysis (LDA) [67] into the clustering framework such that while the clustering provides the labels for LDA, the LDA provides the subspace for clustering. Based on this, Ye et al. [231] proposed disKmeans (discriminative K-means), which performs LDA subspace selection and clustering simultaneously.

Other iteration reduction methods

Moodi and Saadatfar [146] proposed an improved version of K-means for big data clustering with an acceptable precision rate and reduced processing loads. Their method uses the distances between points and the two nearest centroids to consider their variations in the last two iterations. Points within the same equidistance threshold or the acceptable benchmarked equidistance index are not included in the distance calculation. They are added to the same cluster as the pivot point under consideration. Table 6 presents a summary of algorithm processes enhancement variants.

3.3.3. Algorithmic output improvement variants

The standard algorithm is known to perform well with compact and hyper-spherical clusters. However, specifying the cluster number as an algorithm input results in different cluster shapes and outlier effects [10]. Similarly, the simple squared distances in the clustering process model ball-shaped clusters only [224], leaving the standard K-means algorithm struggling to discover clusters of other shapes. Some variants were proposed to address this peculiar problem. Chokniwal and Singh [44] state that cluster shapes depend greatly on the clustering algorithm's distance metric.

a.) Detection of clusters of other shapes

According to Chokniwal and Singh [44], real-world data always follow Gaussian distribution when collected in large quantities. Therefore, they proposed a faster Mahalanobis K-means clustering for Gaussian distributions. Using Gaussian mixture models (GMMs) with Mahalanobis distance enhances the identification of elliptical clusters that are real-life-like in presentation. Berry and Maitra [24] introduced the TiK-means algorithm to extend the standard K-means for partitioning skewed groups. The skewness-transformation parameters are estimated as observations and are assigned to groups during the clustering process. The groups and transformation obtained are presented in the form of generally structured clusters that can be understood through the inversion of the estimated transformation.

Cheung [42] presented the stepwise automatic rival-penalized (STAR). The K*-means variant is a generalized version of the standard algorithm for detecting data clusters with ellipse shapes. It can also perform clustering on the data set without specifying the number of clusters generated. The algorithm has two steps: a preprocessing procedure that assigns at least a seed point to each true cluster, and an update section that updates each seed point based on a winner-takes-all learning rule. Each input is then dynamically assigned to a cluster using the maximum-a-posteriori (MAP) principle. For each input to the cluster, the ideal of the rival penalized competitive learning (RPCL) algorithm [225] was followed while the chances of rival seeds winning were reduced to zero. The algorithm had an outstanding performance compared with the standard algorithm.

Ren and Fan [182] proposed a variant algorithm that uses variation coefficient for similarity measure instead of the Euclidean distance to solve the problem of inaccurate reflection of the similarity among data objects. This decreases the effects of irrelevant data object features, and the clustering results were better.

Table 6
Summary of algorithm processes enhancement variants.

Algorithm Processes Enhancement	References	Methods
Data objects Assignment Process Modification	Ichikawa and Morishita [91]	Heuristic variant based on Pearson correlation distance
	Geng et al. [71]	The fuzzy entropy-based cost function
	Lai, Huang, and Liaw [118]	Center displacement information
	Lee and Lin [122]	Center displacement information and norms product test
	Nie et al. [160].	Reformulation of classical K-means as a trace maximization problem
Iteration Reduction Variants	Na, Xumin, and Yong [150]	Data structures for data objects labels
	Fahim et al. [62]	Use of relevant information for a subsequent iteration
	Kanungo et al. [102]	Filtering algorithm using K_d-tree
	Wang et al. [212]	Multiple random spatial partition trees
	Xie, Liu, and Wei [221]	Dimension reduction technique
	Capó, Pérez, and Lozano [34]	Recursive and parallel approximation
	Ye et al. [231]	Discriminative clustering based on linear discriminant analysis (LDA)
	Moodi and Saadatfar [146]	Variations in distances from two centroids
	Moodi and Saadatfar (2022).	Variations in distances from two centroids along with variation in the previous two iterations

b.) Fuzzy clusters

Bezdek, Ehrlich, and Full [26] presented a variant of the K-means algorithm called the fuzzy c-means (FCM), which generates fuzzy partitions and prototypes of any numerical datasets. A generalized least-squares objective function was employed as the clustering criterion with either Euclidean, Mahalanobis, or diagonal measures as the distance measure among data objects. The algorithm is equipped with an adjustable weighting factor for controlling the algorithm's sensitivity to noise with a platform for specifying variable numbers of clusters and outputs equipped with several measures for cluster validation.

c.) Rough clusters

To efficiently represent incomplete and approximate information, fuzzy and rough set are provided based on fuzzy set theory and rough set theory, respectively [129]. Based on the properties of rough sets, Lingras and West [129] proposed a variant of the K-means algorithm called rough K-means, where clusters are represented as intervals (rough sets). K-means is modified to create cluster intervals to provide an efficient method for representing clusters with vague and imprecise boundaries.

Guo, Han, and Han proposed a new extension of the K-means algorithm called 'k-interval' (2014), with variation in cluster representation as intervals during data objects' similarity computation instead of the center-based representation of the standard K-means. It measures the distance between a data object and a particular cluster as the dissimilarity between the object and the cluster's internal representation. This variant ensures that the simplicity of the standard algorithm is kept and preserves more information about the clusters.

d.) Overlapping clusters

Overlapping clustering presents a tradeoff between crisp and fuzzy clustering and has been found to be a good alternative in some application domains such as biology and information retrieval [46]. Cleuziou [46] presented an extended version of the K-means algorithm called overlapping K-means (OKM) for handling overlapping clustering. A new objective function was defined to be minimized under multi-assignment constraints. It supports data space exploration rather than space partition as in K-means. Table 7 presents a summary of the algorithm output of the various improvement variants of the K-means clustering algorithm.

3.4. Algorithm concept modification

Some reported variants are focused on modifying the standard algorithm's basic concept to reduce the standard algorithm's computational complexity. For instance, Zeebaree et al. [233] reviewed literature that reports the K-means clustering algorithm and genetic algorithm hybridization focusing on the time complexity's efficiency and effectiveness as well as an automatic calculation of the number of clusters in a dataset. Their review discovered and reported that almost all the hybridizations of the K-means algorithm with genetic algorithm have high-performance clustering quality with minimum execution time. Moreover, the evolution process converges faster compared with other techniques.

3.4.1. General algorithm concept modification

Hussain and Haris's [90] variant exploited the dataset's statistical information to guide the algorithm's iterative convergence. They embedded the concepts of higher-order statistics and duality in the data into the standard algorithm to create their K-means co-clustering (kCC) algorithm. The initialization step was modified by representing each cluster with multiple points so that points belonging to the same clusters are closer and far apart from points representing other clusters. Cluster center re-estimation strategy was introduced to maximize the clusters' higher-order walks, replacing the usual re-computation of means of medoids. They were able to record better performance compared with the standard algorithm. Consistency of results obtained and quick convergence and stability were observed in the execution of their variant.

Rezaee et al. [183] proposed game-based K-means (GBK-means), a variant of K-means that leveraged the bargaining game modeling power in the K-means algorithm for clustering data. In this work, each cluster center attracts the largest number of

Table 7
Summary of algorithm output improvement variants.

Improvement Variants	References	Methods
Algorithmic Output Improvement Variants	Cheung [42]	Maximum-A-Posteriori (MAP) principle and rival penalized competitive learning
	Ren and Fan [182]	Variation coefficient for similarity measure
	Chokniwal and Singh [44]	Use of Gaussian mixture models (GMMs) with Mahalanobis distance
	Berry and Maitra [24]	Estimation of skewness-transformation
	Bezdek, Ehrlich and Full [26]	Generation of fuzzy partitions
	Lingras and West [129]	Representation of clusters as intervals
	Cleuziou [46]	Data space exploration using multi-assignment constraints for overlapping clustering
	Guo, Han, and Han [78],	Representation of clusters as intervals during similarity computations
	Wu et al. [218]	Combination of canny algorithm with K-means for cluster pixel recognition

data objects to its cluster through competition with other cluster centers by continually changing its positions to minimize its distances with the maximum possible data compared with other cluster centers. According to Guan et al. [77], the configuration of the standard algorithm is time-consuming due to its iterative nature. Hence, they proposed a reuse-centric K-means configuration to accelerate the algorithm. Techniques such as re-use-based filtering, center re-use, and a two-phase design that capitalizes on the re-use opportunities considering validation, number of clusters, and feature sets, were introduced. From their point of view, effectively reusing computations from prior trials could greatly reduce the configuration time. They were able to achieve standard algorithm typical configuration by 5-9X.

Borlea et al. [29] presented a unified form of K-means and fuzzy c-means algorithm as a single configurable algorithm that facilitates the implementation of either algorithm as desired in a single system. It provides an elegant solution to the challenges of clustering large datasets sequentially. Alguliyev, Aliguliyev, and Sukhostat [13] proposed a parallel batch K-means algorithm for effectively handling big data. The algorithm splits the dataset into equal partitions to reduce the exponential growth of computation and increases the processing speed while preserving the dataset's characteristics. Wu and Wu [219] presented an enhanced regularized K-means type clustering with adaptive weights for differential treatment of different features and handling of the importance of data objects in a dataset while clustering. Adaptive data weights vector and adaptive feature weights matrix are introduced into the K-means algorithm objective function. A significant improvement in clustering performance was observed. Niu et al. [161] proposed a variant of the K-means algorithm termed K-means + to resolve the high complexity problem of low response time requirement in big data clustering. K-means + uses block operation and redesigned distance function to reduce clustering modeling time costs. It uses Manhattan distance in place of Euclidean distance for calculation simplification.

The bisecting K-means (BKM) is a variant of the K-means algorithm proposed by Steinbach, Karypis, and Kumar [202]. It assigns the entire dataset to a single cluster and uses K-means to obtain the best sub-clusters. The process is repeated on the resultant clusters until optimal clusters evolve. BKM is reported to have higher computing efficiency with better cluster quality, and the susceptibility to initial cluster centers is low [249]. Zhuang et al. [249] proposed a limited iteration BKM for fast clustering of large datasets, increasing the efficiency of cluster analysis while maintaining the quality of the resulting clusters. The number of iterations performed on the bisected clusters is limited to three, which achieves the same clustering qualities as the unlimited iteration of the standard BKM.

3.4.2. Hybridization variants

To improve the general performance of the K-means algorithm, the algorithm has been hybridized with several metaheuristics optimization algorithms to improve its performance [6]. In much of the reported literature on K-means hybridization with a metaheuristic optimization algorithm, most of the challenges of the algorithm are also addressed to enhance its general clustering performance. In this era of big data, accurately specifying an optimal number of clusters for the dataset is difficult. Because of this, some of the hybridized algorithms are equipped with the capability for automatic clustering.

Lam, Tsang, and Leung [119] proposed a scheme to improve the particle swarm optimization (PSO)-based K-means algorithm. The clustering problem dimensionality is expanded in the standard PSO-KM hybridized algorithm because the represented clusters' particle sequence is not evaluated. Therefore, the cluster centroid sequence encoded in a particle is matched with the corresponding ones in the global best particle with the closest distance. This ensures the evaluation of the sequence of centroids, optimizing it with the closest distance. Thus, the performance of the particles in collaboratively searching for the optimum is enhanced, the clustering error is effectively reduced, and the convergence rate is improved. Zhang and Zhou [236] developed Nclust to combine a novel niching genetic algorithm with K-means to resolve the same issues of sensitivity to initial cluster seed, determining clusters automatically, and convergence to local optimal. Dai et al. [50], in their proposed parallel genetic algorithm (PGAClust), addressed the issue of sensitivity to the initial cluster center and automatic determination of cluster number; it combined the efficiency K-means algorithm with the parallel genetic algorithm's global optimization ability.

Other K-means are hybridized with metaheuristics algorithms in the literature, including Zhou et al. [244], Sinha and Jana, [201],Kapil, Chawla, and Ansari [104],Rahman and Islam, [179]- Islam et al. [93], Mustafi and Sahoo [149], Xiao et al., [220,103,196]; Hu et al., [87] and Kuo et al. [117]. Ikotun et al. [92] extensively reviewed many nature-inspired metaheuristic algorithms that have been hybridized with K-means. The hybrid algorithms combine the metaheuristics algorithms with the K-means algorithm for improved clustering performance. However, some of the problems inherent in the metaheuristic algorithm, such as high computational complexity and time and the need for parameter tuning for optimal results, affect the overall performance of the resulting hybrid algorithms. Table 8 summarizes the algorithm concept of the K-means modification variants.

3.5. Algorithm implementation variants

In some of the literature, attention is given to enhancing the performance of the K-mean algorithm based on technicalities of implementation without any change to the algorithm structure. Implementation on a parallel machine, the idea of a quantum machine, kernel implementation, and implementation using the map-reduce framework are examples in this category.

Table 8
Summary of Algorithm Concept Modification Variants.

Concept Modification	References	Methods
General Concept Modification	Hussain and Haris [90]	Higher-order statistics and duality concept for Co-clustering
	Rezaee et al. [183]	Bargaining game modeling
	Guan et al, [77]	Reuse-centric configuration
	Borlea et al. [29]	Unified Algorithm combining K-means and C -means
	Alguliyev, Aliguliyev and Sukhostat [13]	Parallel Batch execution
	Wu and Wu [219]	Adaptive weights for differential treatment
	Niu et al. [161]	Block operation with Manhattan distance function
	Steinbach, Karypis and Kumar [202]	Bisection model
	Zhuang et al. [249]	Limited Bisection
	Gocer and Sener [73]	Integrating modified weighted K-means with multi-criteria decision-making tools
Hybridization Variants	Lam, Tsang, and Leung [119]	PSO-based metaheuristic optimization
	Zhang and Zhou [236]	GA-based metaheuristic optimization
	Dai et al. [50]-[173]	GA-based metaheuristic optimization with parallel execution
		PSO, ABC and K-mean hybrid algorithm

3.5.1. Parallel Machine implementation

Kijsipongse and Suriya [112] proposed a variant that adopted parallel implementation of the standard algorithm on graphics processing unit (GPU) clusters utilizing the massive parallelism in GPUs. This speeds up the time-consuming part in each node of the K-mean algorithm. Fatta et al. [64] presented an epidemic variant of the standard algorithm that is fully decentralized to address the problem of communication failure common to a straightforward parallel algorithm in a large-scale geographical distributed system. The epidemic K-means algorithm has a precise parallel formulation that intrinsically faults tolerance without requiring global communication. Its performance is comparable to an ideal centralized algorithm that produces closely desired results. The epidemic K-means algorithm represents a practical and accurate clustering algorithm implemented using a distributed framework suitable for large and highly scaled networked systems.

3.5.2. Quantum Machine implementation

Implementation of the standard algorithm using a quantum machine is another area of implementation variants reported in the literature. Casper et al. [35] explored and evaluated a quantum modeled K-means (QK-means) clustering algorithm in a particular domain (image segmentation) and reported that under specific conditions, there is an improvement in the accuracy and precision of the result obtained compared with the classical counterpart. The probability amplitude values are associated with various self-contained qubit strings defined in the system to generate input into the standard. The convergence rate was controllable, which guaranteed the algorithmic convergence of the algorithm and lowered the calculated variance.

3.5.3. Implementation of MapReduce framework

Implementation of the MapReduce framework is another area for improvement in the performance of the standard algorithm. MapReduce is a robust programming framework in which the job is divided into several tasks and executed in a distributed environment, thus reducing the execution time of an algorithm. With the current era of big data and the attendant challenges in its clustering, the standard K-means is not efficient enough regarding the computational time required when clustering an extensive dataset. Hence adopting the MapReduce programming framework in the implementation of the standard algorithm.

Boukhdhir et al. [30] proposed an improved design of the standard algorithm using the MapReduce programming model improved MapReduced K-means (IM-K means) to achieve efficiency in clustering massive datasets. The MapReduce programming model is a simplified model designed in a parallel environment for handling data-intensive applications. The idea is to adapt the programming model to the standard algorithm for large-scale clustering datasets for execution time reduction. Their work included two other algorithms: one for removing outliers in the dataset and the second for automatically selecting initial centroids to stabilize the result. Their design was implemented on a[1]Hadoop platform. It improves the parallel K-means (PK-means) algorithm based on MapReduce [134], Zhao et al. [241]. The report shows that the execution time for IM-K-means is less than that of the standard algorithm, the variants PK-means, and the fast K-means. The reported limitations are the requirement for the user-dependent specification of parameter k, and it can only work on a numeric dataset.

Van-Hieu and Meesad [208] presented the MapReduce framework for reducing the number of iterations familiar with the standard algorithm. The proposed variant based on MapReduce termed fast MapReduce K-Means (FMR.K-means) reduced the total iteration by 30% with 98% accuracy. According to Cui et al. [48], MapReduce is unsuitable for iterated algorithms because of the need to restart the job repeatedly. As a result, they proposed a novel MapReduce processing model that eliminates iteration dependence resulting in a high-performance, robust and scalable K-means algorithm. Their algorithm employed a sampling technique to select some subset of the big data from which the center set is obtained to cluster the

[1] https://hadoop.apache.org.

original datasets. The PK-means algorithm based on MapReduce by Lv et al. [134] and Zhao, Ma, and He [241] has three functions: the map function, the combine function, and the reduce function. Map function finds each data point's distance to the k center and assigns it to the nearest cluster. The combine function calculates the local centroid and the reduce functions to find the new global centroids. The variant algorithm is very efficient and can be easily implemented. It has a significant improvement in terms of speed and scalability. However, instability and sensitivity to outliers are still a problem to be reckoned with. An adaptively dispersed centroids K-means (ADC-K-means) implemented using the MapReduce model on the Hadoop platform was proposed by Wang et al. [213] to improve the accuracy and stability of clustering results. Dispersed data points based on the distribution of the data points are searched for and selected as the initial centroids for the clusters.

3.5.4. Other implementation paradigms

In this era of big data, effectively clustering massive real-life datasets is very difficult. Hence, Ragunthar et al. [178] proposed the strong reinforcement parallel implementation of the K-means algorithm using a message passing interface. The message passing interface (MPI) programming model is adopted to increase the algorithm's execution speed, scalability, and performance. The MPI is implemented in a parallel environment to enhance the efficiency of the improved algorithm. Other implementation paradigms found in the literature include the work reported by Estlick [58], who explored algorithmic level transformation to map the K-means algorithm to reconfigurable hardware. According to the author, Euclidean distances and floating-point arithmetic are used in the standard K-means algorithm on a general-purpose processor. However, they have speed penalties and occupy a large area when implemented on a field programmable gate array (FPGA). For good performance on an FPGA, the algorithm needs to be transformed. Estlick's [58]. Their implementation on a reconfigurable hardware resulted in approximately 200 times faster than the software implementation in terms of the computational speed.

Benchara and Youssfi [23] presented the distributed service clustering method (DSCM) – a distributed method K-means clustering for a high-performance computing model where a virtual parallel distributed computing model is integrated with a mechanism that supports low communication costs. The proposed algorithm is implemented as a distributed service embedded in a cooperative micro-services team that uses an advanced message queuing protocol (AMQP) based asynchronous communication mechanism for achieving high degree scalability. Table 9 presents a summary of the K-means algorithm implementation variants. Table 10 provides the details of some datasets used to test the performance of the K-means algorithm variant implementation.

4. Discussion

This study presents an extensive literature review on the various improvements of the K-means algorithm, mainly from 2010 to date. The review has surveyed the variety of modifications available of the standard K-means algorithm design and implementation which are intended to enhance its clustering performance and speed. The current study found that the improvements span all the major aspects of algorithm design, including the algorithm input, processes, output, and concept modification. In the area of implementation, new techniques such as parallel machine implementation, quantum machine implementation, the MapReduce framework, amongst others, have been adopted by researchers to enhance the performance of the standard K-means algorithm.

Table 9
Summary of Algorithm Implementation Variants.

Algorithm Implementation Variants	References	Methods
Parallel Machine implementation	Kijsipongse and Suriya [112]	Parallel implementation
	Fatta et al. [64]	Implemented using a distributed framework
	He, Vialle, and Baboulin [84]	Parallel optimization technique on central processing unit (CPU) and graphical processing unit (GPU)
Quantum Machine Implementation	Casper et al. [35]	Quantum model
Map-Reduce Framework Implementation	Boukhdhir et al. [30]	Map-reduce programming model
	Van-Hieu and Meesad [208]	Fast map-reduce model
	Wang et al. [213]	Map-reduce model on the Hadoop platform
	Mao et al. [138]	Parallel clustering based on grid density and a local sensitive hash function (MR-PGDLSH)
Other implementation paradigms	Ragunthar et al. [178]	Reinforcement parallel implementation with message passing interface
	Estlick et al. [58]	Algorithmic level transformation implemented on FPGA
	Benchara and Youssfi [23]	Distributed service clustering method using virtual parallel distributed computing model
	Kavitha and Kaulgud [110]	Quantum circuit for the distance calculation.

Table 10
Details of standard datasets used for the validation of various K-means algorithm variant implementation.

	Datasets	Number of Objects	Number of Attributes	References		Datasets	Number of Objects	Number of Attributes	References
1	3rd Road Network	434,874	3	Capó, Pérez, and Lozano [34]	31	MNIST	60,000	780	Chen et al. [38]
2	Abalone	4177	7	Bache and Lichman [17]	32	Multiple features	2000	649	Bache and Lichman [17]
3	Air Quality Monitoring	720	9	Chen, Chen, and Lu [41]	33	Mushroom	8124	22	Bache and Lichman [17]
4	Breast cancer	286	9	[17].	34	Norm25	2048	15	Arthur and Vassilvitskii [16]
5	CIFAR-10	60,000	1024	Krizhevsky and Hinton [115]	35	Parabolic	500	2	Chen et al. [38]
6	Cloud	1024	10	[17].	36	PenDigits	10,992	16	Chen et al. (2024)
7	Cloud Cover	10,000	54	Bache and Lichman [17]	37	Post patient	90	8	Bache and Lichman [17]
8	Color Quantization	16,000	16	Bache and Lichman [17]	38	Ring	500	16	Chen et al. [38]
9	Corel Image Features	68,037	17	Capó, Pérez, and Lozano [34]	39	RNA (RNASEQUENCES)	488,565	8	Bachem et al. [18]
10	Credit approval	690	15	Bache and Lichman. [17]	40	Satimage	6435	36	Chen et al. (2025)
11	CSN(EARTHQUAKES)	80,000	17	Bachem et al. [18]	41	Shuttle	43,500	9	Chen et al. (2026)
12	Cylinder bands	540	39	[17].	42	SONG (MUSIC SONGS)	515,345	90	Bachem et al. [18]
13	Dermatology	366	34	Bache and Lichman [17]	43	Soybean	307	35	Bache and Lichman [17]
14	Dow Jones Index	180	15	Chen, Chen, and Lu, [41]	44	Spam base	4601	58	Bache and Lichman [17]
15	E. coli	336	7	Bache and Lichman [17]	45	Spiral	312	2	Biswas et al. [27]
16	Flame	240	2	Biswas et al. 2019)	46	Sponge	76	45	Bache and Lichman [17]
17	Ford	79,332	4	Chen, Chen, and Lu [41]	47	SUSY	5,000,000	19	Capó, Pérez, and Lozano [34]
18	Gas Sensor	4,208,259	19	Capó, Pérez, and Lozano [34]	48	SUSY (SuperSym Particles	5,000,000	18	Bachem et al. [18]
19	Gene Expression Data	4029	96	Alizadeh, et al. [14]	49	Synthetic	10,000	3	Arthur and Vassilvitskii [16]
20	Glass	214	9	Bache and Lichman [17]	50	Tunisians Stock exchange daily trading	1,020,000	6	Boukhdhir et al. [30]
21	Heart disease	303	13	Bache and Lichman [17]	51	Two moons	10,000	2	Biswas et al. [27]
22	Hepatitis	155	19	Bache and Lichman [17]	52	Village	11,904	9	Chen, Chen, and Lu [41]
23	Horse colic	299	304	Bache and Lichman [17]	53	Voting	435	16	Bache and Lichman [17]
24	Household Power Consumption	2,049,259	7	Capó, Pérez, and Lozano [34]	54	Web Users Yahoo!	45,811,883	5	Capó, Pérez, and Lozano [34]
25	Image Segmentation	2310	19	Bache and Lichman [17]	55	WEB (WEB USERS)	45,811,883	5	Bachem et al. [18]
26	Internet Advertisement	3279	1558	Bache and Lichman [17]	56	Wind	6574	15	Bache and Lichman [17]
27	Iris	150	4	Bache and Lichman [17]	57	Wine	178	13	Bache and Lichman [17]
28	KDD (PROTEINHOMOLOGY)	145,751	74	Bachem et al. [18]	58	Year Prediction MSD	515,345	91	Bertin-Mahieux et al. [25]
29	Letters	20,000	16	Bache and Lichman [17]	59	Yeast	384	17	Bache and Lichman [17]
30	Leuk72_3k	72	39	Bache and Lichman [17]	60	Zigzag	500	8	Chen et al. (2023)

Regarding the first review question: "*What research has been conducted to improve the standard K-means clustering algorithm?*", insights have been provided on the various modifications to the standard K-means clustering algorithm that have been reported in the literature to enhance its performance. Findings from the reviewed literature show that the main focus has been on improving the standard process of centroid selection to improve the accuracy of the resulting clusters in algorithm input modification. The efficiency of this step depends on the specified input parameter k (number of clusters to be generated). In some cases, dataset preprocessing is performed to enhance optimum centroid selection. To reduce processing time, some researchers concentrated on reducing the number of iterations performed in the distance calculation required between the data objects and the selected centroid. Some improvements were also made in assigning data objects to the various clusters.

In some of the variants, attention was focused more on improving the standard K-means clustering algorithm to detect clusters of other shapes other than the usual spherical or ball-shaped clusters, while in a few cases, the basic concept of the algorithm design was changed to improve its performance. The standard K-means algorithm was implemented differently from the regular model for the algorithm implementation modifications. The primary aim of adapting this new implementation technique is to improve the processing time required by the standard algorithm. In some of the studies, parallel machine implementation was adopted, while quantum machine implementation and the MapReduce framework were adopted in other studies.

The second and third review questions reveal the various methods adopted by the different researchers in improving the standard K-means clustering algorithm and the resulting performances of the adopted methods compared with the standard algorithm. Adopted methods vary with the standard clustering algorithm's identified area of limitations. This is summarized in Tables 4, 5, 6, 7, 8, and 9. Researchers adopted methods in line with the area of the standard algorithm they set out to improve upon in addressing some of the limitations of the standard algorithm. In most of the research works, one or two limitations were addressed, while others left unaddressed still pose the existing corresponding challenges in the reported variants. However, there are marked improvements in the new variants' performances in all the reported research compared with the standard algorithm. In Tables 4, 5, 6, 7, 8, and 9, the summary of methods applied in reported variants of the standard K-means algorithm is presented. The details of some of the datasets used in implementing the K-means variants are shown in Table 10.

All the aspects of the improvements to the K-means algorithm continued to re-emerge as a research focus throughout the years reported for further refinement to enhance their performances. It can be observed that amendment to the K-means algorithm started with improving the clustering output in 1984 and continued till 2019 with various approaches. The need to apply K-means for clustering non-numerical or mixed datasets introduced the corresponding variants in 1998. Research in this area is still ongoing as one of the current research areas in 2021. In the year 2000, three different areas of modification to the classical K-means were reported: automatic specification of k, general modification of the K-means concept, and reduction in the number of iterations performed in the classical algorithm. With the introduction of the metaheuristic algorithm for clustering, hybridizing these algorithms with K-means is currently ongoing to automatically find the number of clusters in any given dataset and to generate optimized cluster centers for the K-means. Not much research was available on general concept modification of the K-means algorithm from 2000 until 2016, when research in this area was re-visited and has continued with active engagement through to 2021.

In the case of approaches for reducing the number of iterations, research reports were gathered between 2000 and 2021 with research breaks at three different periods, with the most extended break recorded between 2012 and 2020. Implementation of K-means using other paradigms and resolution of the convergence of K-means to local optimal took the research stage in 2001, while attention to improving the initial cluster centroids selection came on board in 2003. PCA was introduced to generate initial cluster centroids in 2004, and hybridization of K-means with other algorithms was considered in 2007. Research on detecting outliers and variants addressing large-scale clustering was reported in the literature in 2009. In 2011, research attention was paid to clustering datasets with missing values or incomplete records, multiview data, and parallel machine implementation, while quantum machine implementation was introduced in 2012.

New approaches to improving initial cluster centroid selection and resolution of optimal local convergence were also reported in 2012. In 2013, variants involving modification to data objects assignment to clusters were reported with new approaches for other existing variants. Domain-based data preprocessing variant was reported in 2014, and implementation using the MapReduce framework was reported in 2015. In 2016, much attention was paid to improving the procedure for initial centroid selection with further investigation on large-scale clustering and the MapReduce framework implementation. Six areas of improvement, including outlier detection, domain-based data preprocessing large-scale clustering, mixed dataset clustering, and avoidance of optimal local convergence, were further investigated and reported in 2017. The hybridization of K-means took center stage in 2018, along with other areas. Privacy preservation consideration was also investigated during the period. In recent years, from 2019 to date, research involving each of the existing variants has been ongoing to meet new challenges posed by various domains to cater to the clustering needs within the domain, with attention being paid to automatic clustering with reduced computational time for robust and improved clustering output.

The last review question: "*What are the current research progressions involving the K-means clustering algorithm?*" is a major part of this review work. Evolving application areas such as big data with a massive dataset characterized by high dimensionality is a major aspect of current research involving K-means. Adapting the traditional K-means to cluster such datasets successfully is the current trend in K-means variants. The explosive size and dynamic nature of real-life data make it difficult to specify the number of clusters apriori required in traditional K-means. New research directions in K-means variants

regarding this aspect seek to make K-means clustering automatic without specifying the number of clusters as an initial parameter. Other research regarding K-means variants and improved clustering of big data involve computational complexity reduction, handling high data dimensionality, addressing the convergence into local optimum, and data object representation for effective clustering. Sections 5 and 6 detailed the current research progression in K-means variants.

5. Trending application areas of the K-Means algorithm

The K-means clustering algorithm and variants have been applied widely in many research areas, including: image recognition [12], image processing [151], market analysis [70], data processing [152], medical image segmentation [153,151], risk evaluation [248], medical diagnosis [200,245], medical services [215,100], etc. In the health sector, some parts of the human body have been tested and examined for tumor detection (tumor in the brain, cancer), and cluster analysis in medical science is a major issue in digital (medical) image processing. A human brain tumor as one of its most challenging issues facing medical science. The template-based K-means clustering algorithm has been combined with an improved fuzzy c-means algorithm to automatically detect tumors in the human brain from magnetic resonance images [12]. Nanda et al. [151] hybridized the galactic swarm optimization (GSO) with the K-means clustering algorithm for detecting and capturing the shape, location, and size of various brain tumors on brain MRI images. Improved ensemble learning has been combined with K-means clustering by Singh et al. [200] as an intelligent hybrid diagnosis system for hepatitis disease. Zhu, Idemudia, and Feng [245] combined a logistic regression classifier with PCA enhanced K-means clustering algorithm for diabetes diagnosis and prediction at the early stage.

Nasir, Mashor, and Mohamed [153] used a variant of the standard algorithm for the malaria slide image segmentation tool in medical image segmentation. Garg and Jindal [69] used the K-means clustering algorithm in conjunction with an optimized firefly algorithm for skin lesion segmentation. Segmentation of mammography images using fuzzy c-means and K-means clustering algorithm was proposed by Kamil and Salih [100]. In medical services, deciding on the best location to build a new private clinic is critical for its development and survival on a long-term basis. A two-dimensional uncertain language variables (2DULVs) integrated with a technique for order preference by similarity to ideal solution (TOPSIS), and the Dempster-Shafer conjunctive combination rule (DSCCR) model, were combined with K-means clustering algorithm by Wang et al. [215] to develop a feasible scheme for citing new private clinics. Li et al. [125] used the K-means clustering algorithm with the help of quantile transformation (Song and Yang, 2019) for effectively clustering the population of overweight and obese invididuals from a metabolic point of view, even though there is usually considerable variations in the obesity attribute values, including skewed distribution of the importance and presence of outliers. Kamil and Salih [100] proposed a novel K-means clustering algorithm for automated differential blood smear cell detection and counting for clinical examination.

In the financial sector, the K-means algorithm has been used for early financial risk warning by identifying and analyzing existing factors of financial risk in a financial dataset. Zhu and Liu [248] constructed a K-means clustering algorithm-based financial risk early-warning model that avoids the subjective negative impact of artificial division thresholds, redistributing the target dataset to obtain an optimized solution. The K-means clustering algorithm was used with Markov chains to determine inflection points at which companies' credit risk moves from minimal to high [70].

Customer relationship management is crucial in determining which consumers are most profitable for a company's future growth. Nandapala and Jayasena [152] utilized the K-means algorithm for customer segmentation based on a customer's annual income and spending score. Lingxian, Jiaqing, and Shihuai [130] combined K-means clustering with the standard unsupervised learning of data features with long-term advanced memory artificial intelligence models as an integrated framework for effective allocation of resources and reduction in the cost of sale network. Furthermore, the K-means clustering algorithm has been used in borrowers' credit quality scoring models [166], identifying sustainable manufacturing layouts [207], and predicting exchange-traded fund (ETF) performance [228].

In aviation and automotive systems, Lim and Hwang [128] used the K-means algorithm to cluster the commuting data of high population density areas in Seoul to show variations in the number of vertiports' traveling time to determine appropriate locations for new vertiports for personal air vehicles. Pusadan, Buliali, and Ginardi [175] used K-means in conjunction with support vector machine (SVM) and K-nearest neighbor (kNN) to detect flight route anomalies using similarity and grouping. The K-means clustering algorithm combined with the minimum bounding rectangle model has been used for helicopter maritime search area planning [223]. Sanwale and Singh [186] used the K-means clustering algorithm to train the radial basis function neural network (RBF NN) to find the centers of the RBF for the computation of stability and control derivatives from recorded flight data.

K-means clustering algorithm has been used in privacy protection [247], data security protection [32], smart building electrical equipment identification applications [235], and determining the engagement level of students in an e-learning environment [147].

6. Open issues and challenges

The main aim of the K-means algorithm and its variants is to group any given dataset into k clusters such that the data objects within clusters are similar but different from the ones in other clusters. Open issues and challenges in the K-means

algorithm and its variants include the challenges common to the generality of clustering techniques as well as those peculiar to it.

Initialization Problem: - The initialization problem in K-means is twofold: defining the accurate cluster numbers to be generated from the given dataset, and solving the problem of locating the position of initial centroids. The result obtained from K-means depends on the specified initial cluster number (k) [2,66,90]. As a result, the quest for improvement in initialization is an aspect of the K-means algorithm that has witnessed much research, as shown in the reported variants. An accurate number of clusters must be stated so that the algorithm iteration will not get stuck in local optimal, producing wrong clusters [174]. Fränti and Sieranoja [66] presented essential factors that negatively affect the K-means algorithm's performance and how much the negative impact of these factors can be overcome using two major concepts as proposed by them. The authors observed that despite their proposed approaches to alleviating the identified problems of K-means, K-means performance on any given dataset is still unarguably dependent on the number of clusters specified.

Accurate determination of initial centroids from the given dataset is another aspect of the K-means initialization problem. According to Fränti and Sieranoja [66]; solving the problem of locating initial centroids is very difficult. Several techniques have been proposed in the literature addressing the issue of the K-means algorithm's initial centroid selection. These include random selection, furthest point heuristic, sorting heuristic, density-based, projection-based, and splitting techniques. Several comparisons have been carried out and reported in the literature [169,83,204,36,107] in order to determine which of them is consistently better than others in terms of comparable computational requirements. Fränti and Sieranoja [66] reported that a clear state-of-the-art is missing. Therefore, there is a need for an initialization technique for K-means that is simple to implement with low time complexity and no additional parameters.

Computational Complexity: The standard algorithm is burdened with the problem of needing to perform a large number of calculations of each data point's distance from the centroid at each iteration [109]. This requires a time proportional to the product of the cluster number and number of data objects. The standard K-means algorithm is computationally expensive when clustering big data due to the number of iterations involved [15,30]. This becomes relevant because big data characterizes the current era. There is a dire need for a reduction in the number of iterations to be executed when clustering large datasets to scale up the performance of the standard algorithm for efficient handling.

High Data Dimensionality: K-means algorithm cannot efficiently perform maximum-margin and information-theoretic clustering on high dimensional data [61]. Big data analysis is a complicated task due to new challenges raised with big data. As mentioned earlier, the high data dimensionality challenges are reflected in data distribution, size, noise, and heterogeneity (Xie, Liu, and We, 2020). K-means algorithm and its current variants usually encounter problems dealing with these challenges within a reasonable running time. Hadoop MapReduce technique is now commonly applied to cluster big data using standard K-means in a distributed environment [15]; Agnivesh, [165]. Enhancement of existing variants to handle higher dimensional data has been referred to in much of the current research as a viable future research direction [21].

Global Optimum Convergence: Random initialization of centroids often leads to unexpected convergence [10]. According to Fränti and Sieranoja [66], K-means rarely find the global optimum for the centroid locations. This is because the movement of centroids between clusters is impossible if there is a significant distance between clusters or if stable clusters are situated between two such clusters. The use of metaheuristics algorithms for finding the global optimum for the clustering problem is a new and thriving research direction now being explored [59,99]. Deploying metaheuristic algorithms for clustering further solves the problem of automatic clustering where it is not required to specify cluster numbers beforehand [174,30]. Some of these metaheuristics algorithms have been combined with K-means algorithm to enhance its performance. Hybridizing these algorithms with K-means is a viable future research direction for the purpose of optimizing the standard algorithm's performance [59,99,185].

Data Object Representation: The standard algorithm with most variants still cannot handle various data types [10]. Different application areas have different methods for their domain data representation. Most of the existing variants of the K-means algorithm use numerical data for distance calculation, making them unsuitable for clustering datasets with mixed attributes. There is a need to develop a flexible mapping for data representation and a distance calculation scheme that supports mixed data types. This will enable the K-means algorithm to quickly handle datasets with mixed data types. Ahmed et al. [10], in their paper, suggested combining overlap measures (for categorical attributes) with Euclidean distance (for numerical attributes) as a mixed distance measure to enhance the performance of the K-means algorithm.

Irregular Cluster Shape: The K-means algorithm performs well with compact and hyper-spherical clusters. However, the need for apriori specification results in different cluster shapes and outlier effects [10]. Aside from this, using the simple squared distances in the clustering process can only model ball-shaped clusters [224] which means that the K-means algorithm strubbles to discover clusters of other shapes.

General Effectiveness and Scalability: Most of the proposed variants of K-means algorithms for handling some of the known disadvantages of the standard algorithm are domain-specific and do not generalize very well [10]. A particular variant may handle categorical datasets very well but exhibit poor performance due to the adopted initialization. According to these authors there is no universal solution for the limitations of the standard algorithm. Existing variants are either application-specific or data-specific. There is a need for a robust generalized K-means algorithm that can be used in various domains with considerable effectiveness and scalability as the dataset increases in size. Pérez-Ortega, Almanza-Ortega, and Romero [170] have suggested implementing K-means algorithms and variants using parallel and distributed computing paradigms to enhance the algorithm performance processing speed.

Sensitivity to outliers: A dataset with many outliers produces unstable clusters with several K-means clustering algorithm runs. The presence of outliers in the dataset also increases the sum of square error within clusters, thus affecting the accuracy of the final clustering result [193]. There is a need to incorporate an outlier detection and removal scheme into the K-means algorithm to improve its performance further.

7. Conclusion

The K-means clustering algorithm is known for its simplicity and is applied in clustering datasets from different domains. Despite this advantage, its performance is greatly hampered due to some of the problems inherent in its implementation. As a result, much research has been conducted to improve the algorithm's general performance. This review work has been able to identify the various limitations of the standard algorithm and the numerous variants developed to solve the identified problems up to the time of this review work. This paper will benefit researchers working on extending the existing variants to achieve a more robust and scalable K-means-based clustering technique, as well as practitioners interested in using the state-of-the-art variants of the standard algorithm to meet the data clustering needs in their domain. Practitioners having a problem with the existing K-means-based algorithm can easily identify which variant will adequately serve their application need, or identify the method that can be adopted to improve their existing algorithm.

The findings of this study reveal that much focus has been placed on solving the initialization problems of K-means algorithms with little focus on addressing the problem of mixed data type. New technology such as MapReduce, parallel implementation, and Kernel-based implementation is being researched to address big data clustering using the standard algorithm. The hybridization of the standard algorithm with a metaheuristic algorithm for automatic clustering is a new and upcoming area with little work done so far. Only a few existing metaheuristic algorithms have been reportedly combined with the standard algorithm to solve the problem of convergence into local optimal. Future research can investigate automatic clustering algorithms that hybridize the standard or variants with other swarm intelligence metaheuristic algorithms. Researchers and practitioners seeking to design improved automatic clustering based on the standard algorithm or its variants will find this survey very useful.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Abernathy, M.E. Celebi, The incremental online k-means clustering algorithm and its application to color quantization, Expert Syst. Appl. 207 (2022) 117927.
[2] K. Abhishekkumar, C. Sadhana, Survey report on K-means clustering algorithm, Int. J. Mod. Trends Eng. Res 4 (2017) 218–221.
[3] L. Abualigah, A. Diabat, Z.W. Geem, A comprehensive survey of the harmony search algorithm in clustering applications, Appl. Sci. 10 (11) (2020) 3827.
[4] L.M.Q. Abualigah, Feature selection and enhanced krill herd algorithm for text document clustering, Springer, Berlin, 2019.
[5] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[6] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A.H. Gandomi, A novel hybridization strategy for krill herd algorithm applied to clustering techniques, Appl. Soft Comput. 60 (2017) 423–435.
[7] M.B. Agbaje, A.E. Ezugwu, R. Els, Automatic data clustering using hybrid firefly particle swarm optimization algorithm, IEEE Access 7 (2019) 184963–184984.
[8] A. Ahmad, L. Dey, A K-mean clustering algorithm for mixed numeric and categorical data, Data Knowl. Eng. 63 (2007) 503–527.
[9] A. Ahmad, S.S. Khan, Survey of state-of-the-art mixed data clustering algorithms, IEEE Access 7 (2019) 31883–31902.
[10] Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The K-means algorithm: A comprehensive survey and performance evaluation. *Electronics (Switzerland), 9 (8),* 1-12. (1295). https://doi.org/10.3390/electronics9081295.
[11] Ailon, N., Jaiswal, R., & Monteleoni, C. (2009). Streaming K-means approximation. NIPS'09: Proceedings of the 22nd International Conference on Neural Information Processing Systems *Advances in Neural Information Processing Systems, 22*, 10-18.
[12] M.S. Alam, M.M. Rahman, M.A. Hossain, M.K. Islam, K.M. Ahmed, K.T. Ahmed, B.C. Singh, M.S. Miah, Automatic human brain tumor detection in MRI image using template-based K means and improved fuzzy C means clustering algorithm, Big Data Cognit. Comput. 3 (2) (2019) 27.
[13] R.M. Alguliyev, R.M. Aliguliyev, L.V. Sukhostat, Parallel batch K-means for big data clustering, Comput. Ind. Eng. 152 (2021) 107023.
[14] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, Distinct types of diffuse large B-celllymphoma identified by gene expression profiling, Nature 403 (2000) 503–511.
[15] K. Alsabti, S. Ranka, V. Singh, An efficient k-means clustering algorithm. *Electrical Engineering and Computer*, Science 43 (1997).
[16] Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *Proceedings of the Annuual ACM-SIAM Symposium on Discrete Algorithms, 8,* 1027-1035. 10.1145/1283383.1283494.
[17] K. Bache, M. Lichman, UCI machine learning repository, University of California, School of Information and Computer Science, Irvine, CA, USA, 2013 [Accessed Online: 02/07/2021]. Available:.
[18] Bachem, O., Lucic, M., Hassani, H., & Krause, A. (2016). Fast and provably good seedings for k-means. *Advances in Neural Information Processing Systems*, *29*.
[19] L. Bai, J. Liang, F. Cao, A multiple K-means clustering ensemble algorithm to find nonlinearly separable clusters, Inform. Fusion 61 (2020) 36–47.

[20] A. Belhadi, Y. Djenouri, K. Nørvåg, H. Ramampiaro, F. Masseglia, J.C.W. Lin, Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities, Eng. Appl. Artif. Intel. 95 (2020) 103857.
[21] S.B. Belhaouari, S. Ahmed, S. Mansour, Optimized k-means algorithm, Math. Probl. Eng. 2014 (2014), https://doi.org/10.1155/2014/506480.
[22] B. Ben Gouissem, R. Gantassi, S. Hasnaoui, Energy efficient grid-based k-means clustering algorithm for large scale wireless sensor networks, Int. J. Commun Syst e5255 (2022).
[23] F.Z. Benchara, M. Youssfi, A new scalable distributed K-means algorithm based on Cloud micro-services for high-performance computing, Parallel Comput. 101 (2021) 102736.
[24] N.S. Berry, R. Maitra, TiK-means: Transformation-infused K-means clustering for skewed groups, Stat. Anal. Data Mining: ASA Data Sci. J. 12 (3) (2019) 223–233.
[25] T. Bertin-Mahieux, D.P. Ellis, B. Whitman, P. Lamere, The Million-Song Dataset, Academic Commons, Columbia University, 2011, pp. 591–596.
[26] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.
[27] Biswas, C., Ganguly, D., Roy, D., & Bhattacharya, U. (2019). Privacy preserving approximate K-means clustering. *In Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1321-1330.
[28] J. Blömer, C. Lammersen, M. Schmidt, C. Sohler, Theoretical analysis of the K-means algorithm–a survey, in: Algorithm Engineering, Springer, Cham, 2016, pp. 81–116.
[29] I.D. Borlea, R.E. Precup, A.B. Borlea, D. Iercan, A unified form of fuzzy C-means and K-means algorithms and its partitional implementation, Knowl.-Based Syst. 214 (2021) 106731.
[30] A. Boukhdhir, O. Lachiheb, M.S. Gouider, An improved MapReduce design of Kmeans for clustering very large datasets, in: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), IEEE, 2015, pp. 1–6, https://doi.org/10.1109/AICCSA.2015.7507226.
[31] P.S. Bradley, O.L. Mangasarian, W.N. Street, Clustering via concave minimization, Adv. Neural Inf. Proces. Syst. 9 (1997) 368–374.
[32] Cai, J., Liao, D., Chen, J., Chen, X., Liu, T., & Xi, J. (2020). Research on data security protection method based on improved K-means clustering algorithm. In *2020 the 4th International Conference on Big Data Research (ICBDR'20)*, 7-11.
[33] D. Calandriello, L. Rosasco, Statistical and computational trade-offs in kernel k-means, Adv. Neural Inf. Proces. Syst. 31 (2018).
[34] M. Capó, A. Pérez, J.A. Lozano, An efficient K-means clustering algorithm for tall data, Data Min. Knowl. Disc. (2020) 1–36.
[35] Casper, E., Hung, C. C., Jung, E., & Yang, M. (2012). A quantum-modeled K-means clustering algorithm for multi-band image segmentation. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, 158-163.
[36] M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the K-means clustering algorithm, Expert Syst. Appl. 40 (1) (2013) 200–210.
[37] Chawla, S., & Gionis, A. (2013). K-means–: A unified approach to clustering and outlier detection. In *Proceedings of the 2013 Society for Industrial and Applied Mathematics (SIAM) international conference on data mining*, 189-197.
[38] L. Chen, S. Zhou, J. Ma, M. Xu, Fast kernel K-means clustering using incomplete Cholesky factorization, Appl. Math Comput. 402 (2021) 126037.
[39] X. Chen, Y. Yang, Diffusion K-means clustering on manifolds: Provable exact recovery via semidefinite relaxations, Appl. Comput. Harmon. Anal. 52 (2021) 303–347.
[40] X. Chen, X. Xu, J.Z. Huang, Y. Ye, TW-K-means: Automated two-level variable weighting clustering algorithm for multiview data, IEEE Trans. Knowl. Data Eng. 25 (4) (2011) 932–944.
[41] Y.C. Chen, Y.L. Chen, J.Y. Lu, MK-means: Detecting evolutionary communities in dynamic networks, Expert Syst. Appl. 176 (2021) 114807.
[42] Y.M. Cheung, K∗-Means: A new generalized *K-means* clustering algorithm, Pattern Recogn. Lett. 24 (15) (2003) 2883–2893.
[43] I. Chien, C. Pan, O. Milenkovic, Query k-means clustering and the double dixie cup problem, Adv. Neural Inf. Proces. Syst. 31 (2018).
[44] A. Chokniwal, M. Singh, Faster Mahalanobis K-means clustering for Gaussian distributions, in: 2016 International Conference on Advances in Computing, Communications, and Informatics, (ICACCI), IEEE, 2016, pp. 947–952.
[45] G. Cleuziou, A generalization of K-means for overlapping clustering, Rapport Technique 54 (2007).
[46] Cleuziou, G. (2008, December). An extended version of the K-means method for overlapping clustering. In *2008 19th International Conference on Pattern Recognition*, *IEEE*, 1-4.
[47] Couto, J. (2005, September). Kernel K-means for categorical data. In *International Symposium on Intelligent Data Analysis Springer, Berlin, Heidelberg*, 46-56.
[48] X. Cui, P. Zhu, X. Yang, K. Li, C. Ji, Optimized big data K-means clustering using MapReduce, J. Supercomput. 70 (3) (2014) 1249–1259.
[49] Z. Dafir, Y. Lamari, S.C. Slaoui, A survey on parallel clustering algorithms for big data, Artif. Intell. Rev. 54 (4) (2021) 2411–2443.
[50] Dai, W., Jiao, C., & He, T. (2007). Research of K-means clustering method based on parallel genetic algorithm. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, IEEE, *2*,158-161.
[51] H.T. Dashti, T. Simas, R.A. Ribeiro, A. Assadi, A. Moitinho, MK-means-modified K-means clustering algorithm, in: The 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1–6.
[52] C. Ding, X. He, K-means clustering via principal component analysis, in: Proceedings of the twenty-first international conference on machine learning, 2004, p. 29.
[53] D.T. Dinh, V.N. Huynh, S. Sriboonchitta, Clustering mixed numerical and categorical data with missing values, Inf. Sci. 571 (2021) 418–442.
[54] A.V. Doumas, V.G. Papanicolaou, The coupon collector's problem revisited: generalizing the double dixie cup problem of newman and shepp, ESAIM: Probab. Stat. 20 (2016) 367–399.
[55] R.O. Duda, P.E. Hart, Pattern classification and scene analysis Vol. 3 (1973) 731–739.
[56] P. Drineas, A.M. Frieze, R. Kannan, S.S. Vempala, V. Vinay, Clustering in large graphs and matrices, SODA 99 (1999) 291–299.
[57] Elkan, C. (2003). Using the triangle inequality to accelerate K-means. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), Washington, DC, USA*,147-153.
[58] M. Estlick, M. Leeser, J. Theiler, J.J. Szymanski, Algorithmic transformations in the implementation of k-means clustering on reconfigurable hardware, in: Proceedings of the 2001 ACM/SIGDA ninth international symposium on Field programmable gate arrays, 2001, pp. 103–110.
[59] A.E. Ezugwu, M.B. Agbaje, N. Aljojo, R. Els, H. Chiroma, M. Abd Elaziz, A comparative performance study of hybrid firefly algorithms for automatic data clustering, IEEE Access 8 (2020) 121089–121118.
[60] A.E. Ezugwu, A.M. Ikotun, O.O. Oyelade, L. Abualigah, J.O. Agushaka, C.I. Eke, A.A. Akinyelu, A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intel. 110 (2022) 104743.
[61] A.E. Ezugwu, A.K. Shukla, M.B. Agbaje, O.N. Oyelade, A. José-García, J.O. Agushaka, Automatic clustering algorithms: A systematic review and bibliometric analysis of relevant literature, Neural Comput. & Applic. 33 (11) (2021) 6247–6306.
[62] A.M. Fahim, A.M. Salem, F.A. Torkey, M. Ramadan, An efficient enhanced K-means clustering algorithm, Journal of Zhejiang University-Science A 7 (10) (2006) 1626–1633.
[63] A. Farcomeni, Snipping for robust K-means clustering under component-wise contamination, Stat. Comput. 24 (6) (2014) 907–919.
[64] Fatta G. D., Blasa, F., Cafiero, S., & Fortino, G. (2011). Epidemic K-means clustering. *2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, 151(158), 11-11 Dec. 2011.
[65] J. Feng, Z. Lu, P. Yang, X. Xu, A K-means clustering algorithm based on the maximum triangle rule, in: 2012 IEEE International Conference on Mechatronics and Automation, 2012, pp. 1146–1456.
[66] P. Fränti, S. Sieranoja, How much can K-means be improved by using better initialization and repeats?, Pattern Recogn 93 (2019) 95–112.
[67] K. Fukunaga, Introduction to statistical pattern recognition, Elsevier (2013), https://doi.org/10.1016/C2009-0-27872-X.
[68] G. Gan, C. Ma, J. Wu, Data clustering: theory, algorithms, and applications, Biometrics 64 (2020) 651–662.
[69] S. Garg, B. Jindal, Skin lesion segmentation using K-means and optimized firefly algorithm, Multimed. Tools Appl. 80 (5) (2021) 7397–7410.

[70] N. Gavira-Durón, O. Gutierrez-Vargas, S. Cruz-Aké, Markov chain K-means cluster models and their use for companies' credit quality and default probability estimation, Mathematics 9 (8) (2021) 879.
[71] X. Geng, Y. Mu, S. Mao, J. Ye, L. Zhu, An improved K-means algorithm based on fuzzy metrics, IEEE Access 8 (2020) 217416–217424.
[72] Georgogiannis, A. (2016). Robust K-means: a theoretical revisit. *Advances in Neural Information Processing Systems,* 29.
[73] F. Gocer, N. Sener, Spherical fuzzy extension of AHP-ARAS methods integrated with modified k-means clustering for logistics hub location problem, Expert. Syst. 39 (2) (2022) e12886.
[74] M. Gönen, A.A. Margolin, Localized data fusion for kernel K-means clustering with application to cancer biology, Adv. Neural Inf. Proces. Syst. 27 (2014).
[75] M. Goyal, S. Aggarwal, A review on K-mode clustering algorithm, Int. J. Adv. Res. Comput. Sci. 8 (7) (2017).
[76] L. Gu, A novel locality sensitive K-means clustering algorithm based on subtractive clustering, in: In 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016, pp. 836–839.
[77] H. Guan, Y. Ding, X. Shen, H. Krim, Reuse-centric K-means configuration, Inf. Syst. 100 (2018) 101787.
[78] F. Guo, D. Han, C. Han, K-intervals: A new extension of the K-means algorithm, in: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, 2014, pp. 251–258.
[79] S. Gupta, R. Kumar, K. Lu, B. Moseley, S. Vassilvitskii, Local search methods for K-means with outliers, Proceedings of the VLDB Endowment 10 (7) (2017) 757–768.
[80] G. Hamerly, C. Elkan, Learning the k in K-means, Adv. Neural Inf. Proces. Syst. 16 (2003).
[81] B.O.C.K. Hans-Hermann, Origins and extensions of the K-means algorithm in cluster analysis, Journal Electronique d'Histoire des Probabilités et de la Statistique Electron. J. History Prob. Stat. 4 (2) (2008).
[82] H. Harb, A. Makhoul, R. Couturier, An enhanced K-means and ANOVA-based clustering approach for similarity aggregation in underwater wireless sensor networks, IEEE Sens. J. 15 (10) (2015) 5483–5493.
[83] J. He, M. Lan, C.-L. Tan, S.-Y. Sung, H.-B. Low, Initialization of cluster refinement algorithms: A review and comparative study, IEEE International Joint Conference Neural Networks, 2004.
[84] G. He, S. Vialle, M. Baboulin, Parallel and accurate k-means algorithm on CPU-GPU architectures for spectral clustering, Concurr. Comput. Pract. Exp. 34 (14) (2022) e6621.
[85] K. Honda, R. Nonoguchi, A. Notsu, H. Ichihashi, PCA-guided K-means clustering with incomplete data, in: 2011 IEEE International Conference on Fuzzy Systems (FUZZ), 2011, pp. 1710–1714.
[86] K. Honda, A. Notsu, H. Ichihashi, Fuzzy PCA-guided robust K-means clustering, IEEE Trans. Fuzzy Syst. 18 (1) (2009) 67–79.
[87] J. Hu, C. Wang, C. Liu, Z. Ye, Improved K-means algorithm based on hybrid fruit fly optimization and differential evolution, in: 2017 12th International Conference on Computer Science and Education (ICCSE), 2017, pp. 464–467.
[88] S. Huang, Z. Kang, Z. Xu, Q. Liu, Robust deep K-means: An effective and simple method for data clustering, Pattern Recogn. 117 (2021) 107996.
[89] Z. Huang, Extensions to the K-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Disc. 2 (3) (1998) 283–304.
[90] S.F. Hussain, M. Haris, A K-means based co-clustering (kCC) algorithm for sparse, high dimensional data, Expert Syst. Appl. 118 (2019) 20–34.
[91] K. Ichikawa, S. Morishita, A simple but powerful heuristic method for accelerating K-Means clustering of large-scale data in life science, IEEE/ACM Trans. Comput. Biol. Bioinf. 11 (4) (2014) 681–692.
[92] A.M. Ikotun, M.S. Almutari, A.E. Ezugwu, K-Means-based nature-inspired metaheuristic algorithms for automatic data clustering problems: Recent advances and future directions, Appl. Sci. 11 (23) (2021) 11246.
[93] M.Z. Islam, V. Estivill-Castro, M.A. Rahman, T. Bossomaier, Combining K-means and a genetic algorithm through a novel arrangement of genetic operators for high quality clustering, Expert Syst. Appl. 91 (2018) 402–417.
[94] H. Ismkhan, IK-means−+: An iterative clustering algorithm based on an enhanced version of the K-means, Pattern Recogn. 79 (2018) 402–413.
[95] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recogn. Lett. 31 (8) (2010) 651–666.
[96] A.K. Jain, R.C. Dubes, Algorithms for clustering data, Prentice-Hall, Hoboken, NJ, 1988.
[97] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: A review, ACM Comput. Surveys (CSUR) 31 (3) (1999) 264–323.
[98] R.C. Jancey, Multidimensional group analysis, Aust. J. Bot. 14 (1966) 127–130.
[99] A. José-García, W. Gómez-Flores, Automatic clustering using nature-inspired metaheuristics: A survey, Appl. Soft Comput. 41 (2016) 192–213.
[100] M.Y. Kamil, A.M. Salih, Mammography images segmentation via fuzzy C-mean and K-means, Internat. J. Intell. Eng. Syst. 12 (1) (2019) 22–29.
[101] S. Kant, I.A. Ansari, An improved K-means clustering with Atkinson index to classify liver patient dataset, Internat. J. Syst. Assurance Eng. Manage. 7 (1) (2016) 222–228.
[102] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient K-means clustering algorithm: Analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 881–892.
[103] Kao, Y., & Lee, S. Y. (2009). Combining K-means and particle swarm optimization for dynamic data clustering problems. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, IEEE,*1, 757-761.
[104] S. Kapil, M. Chawla, M.D. Ansari, On K-means data clustering algorithm with genetic algorithm, in: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, 2016, pp. 202–206.
[105] A. Kapoor, A. Singhal, A comparative study of K-means, K-means++ and Fuzzy C-Means clustering algorithms, in: 2017 3rd international conference on computational intelligence and communication technology (CICT), 2017, pp. 1–6.
[106] L. Kaufman, P.J. Rousseeuw, Clustering by means of medoids, in: Y. Dodge (Ed.), Statistical Data Analysis Based on the L1 - Norm and Related Methods, Elsivier, Amsterdam, 1987, pp. 405–416.
[107] L. Kaufman, P.J. Rousseeuw, Finding groups in data: An introduction to cluster analysis, John Wiley, New York, NY, 2009.
[108] L. Kaufman, P.J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, John Wiley, New York, NY, 2008.
[109] K. Kaur, D.S. Dhaliwal, R.K. Vohra, Statistically refining the initial points for K-means clustering algorithm, in: International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), 2013, p. 2.
[110] S.S. Kavitha, N. Kaulgud, Quantum K-means clustering method for detecting heart disease using quantum circuit approach, Soft. Comput. (2022) 1–14.
[111] M.F. Khan, K.L.A. Yau, R.M. Noor, M.A. Imran, Survey and taxonomy of clustering algorithms in 5G, J. Netw. Comput. Appl. 154 (2020) 102539.
[112] E. Kijsipongse, U. Suriya, Dynamic load balancing on GPU clusters for large-scale K-means clustering, in: 2012 IEEE International Joint Conference on Computer Science and Software Engineering (JCSSE), 2012, pp. 346–350.
[113] B. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering–a systematic literature review, Inf. Softw. Technol. 51 (1) (2009) 7–15.
[114] S. Krey, U. Ligges, F. Leisch, Music and timbre segmentation by recursive constrained K-means clustering, Comput. Stat. 29 (1) (2014) 37–50.
[115] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Computer Science University of Toronto, Canada, 2009.
[116] Kumar, P., & Wasan, S. K. (2010). Analysis of X-means and global K-means using tumor classification. In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), IEEE*, *5*, 832-835.
[117] Kuo, R. J., Suryani, E., & Yasid, A. (2013). Automatic clustering combining differential evolution algorithm and K-means algorithm. In *Proceedings of the Institute of Industrial Engineers Asian Conference Springer, Singapore,* 1207-1215.
[118] J.Z. Lai, T.J. Huang, Y.C. Liaw, A fast K-means clustering algorithm using cluster center displacement, Pattern Recogn. 42 (11) (2009) 2551–2556.
[119] Y.K. Lam, P.W.M. Tsang, C.S. Leung, PSO-based K-means clustering with enhanced cluster matching for gene expression data, Neural Comput. Appl. 22 (7) (2013) 1349–1355.

[120] Lange, T., Law, M. H., Jain, A. K., & Buhmann, J. M. (2005). Learning with constrained and unlabelled data. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 731-738), doi: 10.1109/CVPR.2005.210.
[121] S.S. Lee, J.C. Lin, An accelerated K-means clustering algorithm using selection and erasure rules, J. Zhejiang Univ. Sci. C 13 (10) (2012) 761–768.
[122] S.S. Lee, J.C. Lin, Fast K-means clustering using deletion by center displacement and norms product (CDNP), Pattern Recognit Image Anal. 23 (2) (2013) 199–206.
[123] J. Lei, T. Jiang, K. Wu, H. Du, G. Zhu, Z. Wang, Robust K-means algorithm with automatically splitting and merging clusters and its applications for surveillance data, Multimed. Tools Appl. 75 (19) (2016) 12043–12059.
[124] J. Lever, M. Krzywinski, N. Altman, Points of significance: Principal component analysis, Nat. Methods 14 (7) (2017) 641–643.
[125] L. Li, Q. Song, X. Yang, K-means clustering of overweight and obese population using quantile-transformed metabolic data, Diabetes Metab. Syndrome Obes. Targets Ther. 12 (2019) 1573.
[126] Y. Li, Z. Qin, An improved algorithm of K-means, J. Beijing Inst. Graph. Commun. 2 (2007) 63–65.
[127] A. Likas, N. Vlassis, J.J. Verbeek, The global K-means clustering algorithm, Pattern Recogn. 36 (2) (2003) 451–461.
[128] E. Lim, H. Hwang, The selection of vertiport location for on-demand mobility and its application to Seoul metro area, Int. J. Aeronaut. Space Sci. 20 (1) (2019) 260–272.
[129] P. Lingras, C. West, Interval set clustering of web users with rough K-means, J. Intell. Inf. Syst. 23 (1) (2004) 5–16.
[130] Y. Lingxian, K. Jiaqing, W. Shihuai, Online retail sales prediction with integrated framework of K-means and neural network, in: Proceedings of the 2019 10th International Conference on E-business, Management and Economics, 2019, pp. 115–118.
[131] A. Lithio, R. Maitra, An efficient K-means-type algorithm for clustering datasets with incomplete records, Stat. Anal. Data Mining: ASA Data Sci. J. 11 (6) (2018) 296–311.
[132] S. Lloyd, Least squares quantization in PCM, in: Bell Telephone Labs Memorandum, Murray Hill NJ. Reprinted In: IEEE Trans. Information Theory IT-28 2, 1982, pp. 129–137.
[133] H. Lu, Q. Gao, X. Zhang, W. Xia, A multi-view clustering framework via integrating K-means and graph-cut, Neurocomputing 501 (2022) 609–617.
[134] Z. Lv, Y. Hu, H. Zhong, J. Wu, B. Li, H. Zhao, Parallel K-means clustering of remote sensing images based on MapReduce, in: International Conference on Web Information Systems and Mining, Springer, Berlin, Heidelberg, 2010, pp. 162–170.
[135] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Fifth Berkeley Symposium on Mathematics. Statistics and Probability, University of California Press, Berkeley, CA, 1967, pp. 281–297.
[136] M.S. Mahmud, M.M. Rahman, M.N. Akhtar, Improvement of K-means clustering algorithm with better initial centroids based on weighted average, in: 2012 7th International Conference on Electrical and Computer Engineering, 2012, pp. 647–650.
[137] K. Makarychev, A. Reddy, L. Shan, Improved guarantees for K-means++ and K-means++ Parallel, Adv. Neural Inf. Proces. Syst. 33 (2020) 16142–16152.
[138] Y. Mao, D. Gan, D.S. Mwakapesa, Y.A. Nanehkaran, T. Tao, X. Huang, A MapReduce-based K-means clustering algorithm, J. Supercomput. 78 (4) (2022) 5181–5202.
[139] Y. Marom, D. Feldman, K-means clustering of lines for big data, Adv. Neural Inf. Proces. Syst. 32 (2019).
[140] R. Mendes, J.P. Vilela, Privacy-preserving data mining: methods, metrics, and applications, IEEE Access 5 (2017) 10562–10582.
[141] Min, W., & Siqing, Y. (2010). Improved K-means clustering based on genetic algorithm. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010),* IEEE, 6, 636.
[142] H.L. Minh, T. Sang-To, M.A. Wahab, T. Cuong-Le, A new metaheuristic optimization based on K-means clustering algorithm and its application for structural damage identification in a complex 3D concrete structure, Knowl.-Based Syst. 251 (2022) 109189.
[143] Mirkin, B. (2005), Clustering for data mining: A data recovery approach. Boca Raton FL: Chapman and Hall/CRC. https://doi.org/10.1201/9781420034912.
[144] B.K. Mishra, A. Rath, N.R. Nayak, S. Swain, Far efficient K-means clustering algorithm, in: Proceedings of the International Conference on Advances in Computing, Communications, and Informatics, 2012, pp. 106–110.
[145] D.S. Modha, W.S. Spangler, Feature weighting in K-means clustering, Mach. Learn. 52 (3) (2003) 217–237.
[146] F. Moodi, H. Saadatfar, An improved K-means algorithm for big data, IET Softw. 16 (1) (2021) 48–59.
[147] A. Moubayed, M. Injadat, A. Shami, H. Lutfiyya, Student engagement level in an e-learning environment: Clustering using K-means, Am. J. Dist. Educ. 34 (2) (2020) 137–156.
[148] J.P. Mouton, M. Ferreira, A.S. Helberg, A comparison of clustering algorithms for automatic modulation classification, Expert Syst. Appl. 151 (2020) 113317.
[149] D. Mustafi, G. Sahoo, A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the K-means algorithm with applications in text clustering, Soft. Comput. 23 (15) (2019) 6361–6378.
[150] S. Na, L. Xumin, G. Yong, Research on K-means clustering algorithm: An improved K-means clustering algorithm, in: 2010 Third International Symposium on intelligent information technology and security informatics, 2010, pp. 63–67.
[151] S.J. Nanda, I. Gulati, R. Chauhan, R. Modi, U. Dhaked, A K-means-galactic swarm optimization-based clustering algorithm with Otsu's entropy for brain tumor detection, Appl. Artif. Intell. 33 (2) (2019) 152–170.
[152] Nandapala, E. Y. L., & Jayasena, K. P. N. (2020). The practical approach in customers segmentation by using the K-means algorithm. *In 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS),* IEEE, 344-349.
[153] A.S.A. Nasir, M.Y. Mashor, Z. Mohamed, Enhanced K-means clustering algorithm for malaria image segmentation, J. Adv. Res. Fluid Mech. Thermal Sci. 42 (1) (2018) 1–15.
[154] Nazeer, K. A., & Sebastian, M. P. (2009). Improving the accuracy and efficiency of the k-means clustering algorithm. In *Proceedings of the world congress on engineering, 1*, 1-3. London, UK: Association of Engineers.
[155] K.A. Nazeer, S.M. Kumar, M.P. Sebastian, Enhancing the k-means clustering algorithm by using a O (n logn) heuristic method for finding better initial centroids, 2011 Second International Conference on Emerging Applications of Information Technology 261–264, IEEE, 2011.
[156] Newling, J., & Fleuret, F. (2016). Nested mini-batch K-means. *Advances in Neural Information Processing Systems, 29*.
[157] J. Newling, F. Fleuret, K-medoids for K-means seeding, Adv. Neural Inf. Proces. Syst. 30 (2017).
[158] D.J. Newman, The double dixie cup problem, Am. Math. Mon. 67 (1) (1960) 58–61.
[159] R.T. Ng, J. Han, CLARANS: A method for clustering objects for spatial data mining, IEEE Trans. Knowl. Data Eng. 14 (5) (2002) 1003–1016.
[160] F. Nie, Z. Li, R. Wang, X. Li, An effective and efficient algorithm for K-means clustering with new formulation, IEEE Trans. Knowl. Data Eng. 14 (8) (2022) 1–11.
[161] Niu, K., Gao, Z., Jiao, H., & Deng, N. (2016). K-means+: A developed clustering algorithm for big data. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE*, 141-144.
[162] Olukanmi, P. O., & Twala, B. (2017). K-means-sharp: modified centroid update for outlier-robust K-means clustering. *In 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), IEEE*, 14-19.
[163] P. Olukanmi, F. Nelwamondo, T. Marwala, B. Twala, Automatic detection of outliers and the number of clusters in k-means clustering via Chebyshev-type inequalities, Neural Comput. & Applic. 34 (8) (2022) 5939–5958.
[164] M.K. Pakhira, A linear time-complexity k-means algorithm using cluster shifting, in: 2014 international conference on computational intelligence and communication networks, IEEE, 2014, pp. 1047–1051.
[165] R. Panday, A. Singh, Improved K-means map reduce algorithm for big data cluster analysis, Internat. J. Innov. Technol. Explor. Eng. (IJITEE) 8 (8) (2019).
[166] S. Pang, X. Hou, L. Xia, Borrowers' credit quality scoring model and applications, with default discriminant analysis based on the extreme learning machine, Technol. Forecast. Soc. Chang. 165 (2021) 120462.

[167] H.S. Park, C.H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Syst. Appl. 36 (2) (2009) 3336–3341.
[168] Pelleg, D., & Moore, A. W. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning, June 2000, San Francisco*, 727-734.
[169] J.M. Peña, J.A. Lozano, P. Larrañaga, An empirical comparison of four initialization methods for the K-means algorithm, Pattern Recogn. Lett. 20 (1999) 1027–1040.
[170] J. Pérez-Ortega, N.N. Almanza-Ortega, D. Romero, Balancing effort and benefit of K-means clustering algorithms in big data realms, PLoS One 13 (9) (2018) 0201874.
[171] J. Pérez-Ortega, N.N. Almanza-Ortega, A. Vega-Villalobos, R. Pazos-Rangel, C. Zavala-Díaz, A. Martínez-Rebollar, The K-means algorithm evolution, Introduction to Data Science and Machine Learning, IntechOpen, 2019.
[172] B.A. Pimentel, A.C. de Carvalho, A meta-learning approach for recommending the number of clusters for clustering algorithms, Knowl.-Based Syst. 195 (2020) 105682.
[173] Q. Pu, J. Gan, L. Qiu, J. Duan, H. Wang, An efficient hybrid approach based on PSO, ABC and k-means for cluster analysis, Multimed. Tools Appl. 81 (14) (2022) 19321–19339.
[174] A. Pugazhenthi, L.S. Kumar, Selection of optimal number of clusters and centroids for K-means and fuzzy C-means clustering: A review, in: 2020 5th International Conference on Computing, Communication and Security (ICCCS), IEEE, 2020, pp. 1–4.
[175] M.Y. Pusadan, J.L. Buliali, R.V.H. Ginardi, Anomaly detection on flight route using similarity and grouping approach based-on automatic dependent surveillance-broadcast, Internat. J. Adv. Intell. Inform. 5 (3) (2019) 285–296.
[176] Qi, J., Yu, Y., Wang, L., & Liu, J. (2016). K*-means: An effective and efficient K-means clustering algorithm. In *2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom)(BDCloud-SocialCom-SustainCom), IEEE* 242-249.
[177] Z. Qi, W. Chen, X. Sun, W. Sun, H. Yang, KText: Arbitrary shape text detection using modified K-means, IET Comput. Vis. 16 (1) (2021) 38–49.
[178] T. Ragunthar, P. Ashok, N. Gopinath, M. Subashini, A strong reinforcement parallel implementation of K-means algorithm using message passing interface, Mater. Today: Proc. (2021).
[179] M.A. Rahman, M.Z. Islam, A hybrid clustering technique combining a novel genetic algorithm with K-means, Knowl.-Based Syst. 71 (2014) 345–365.
[180] Rajah, V., & Ezugwu, A. E. (2020). Hybrid symbiotic organism search algorithms for automatic data clustering. *In 2020 Conference on Information Communications Technology and Society (ICTAS), IEEE,* 1-9.
[181] Rathore, P., & Shukla, D. (2015). Analysis and performance improvement of K-means clustering in big data environment. In *2015 International Conference on Communication Networks (ICCN), IEEE*, 43-46.
[182] Ren, S., & Fan, A. (2011). K-means clustering algorithm based on coefficient of variation. In *2011 4th International Congress on Image and Signal Processing IEEE,* 4, 2076-2079.
[183] M.J. Rezaee, M. Eshkevari, M. Saberi, O. Hussain, GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game, Knowl.-Based Syst. 213 (2021) 106672.
[184] J. Saha, J. Mukherjee, CNAK: Cluster number assisted K-means, Pattern Recogn. 110 (2021) 107625.
[185] G. Saini, H. Kaur, K-mean Clustering and PSO: A review, Internat. J. Eng. Adv. Technol. (IJEAT) ISSN 3 (5) (2014) 2249–8958.
[186] J. Sanwale, D.J. Singh, Aerodynamic parameters estimation using radial basis function neural partial differentiation method, Def. Sci. J. 68 (3) (2018).
[187] Sarma, T. H. Viswanath, P., & Reddy, B. E. (2011). A fast approximate kernel K-means clustering method for large data sets. *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, *545*(550), 22-24.
[188] Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web,* 1177-1178.
[189] R. Sharma, V. Vashisht, U. Singh, Performance analysis of evolutionary technique based partitional clustering algorithms for wireless sensor networks, in: Soft Computing: Theories and Applications, Springer, Singapore, 2020, pp. 171–180.
[190] X. Shen, W. Liu, I. Tsang, F. Shen, Q.S. Sun, Compressed K-means for large-scale clustering, Thirty-first AAAI Conference on Artificial Intelligence, 2017.
[191] T. Shibayama, A PCA-like method for multivariate data with missing values, Jpn. J. Educ. Psychol. 40 (1992) 257–265, in Japanese.
[192] Shindler, M., Wong, A., & Meyerson, A. (2011). Fast and accurate K-means for large datasets. *Advances in Neural Information Processing Systems, 24*.
[193] K. Shiudkar, P.S. Takmare, Review of existing methods in K-means clustering algorithm, Internat. Res. J. Eng. Technol. 4 (2) (2017) 1213–1216.
[194] N.H. Shrifan, M.F. Akbar, N.A.M. Isa, An adaptive outlier removal aided K-means clustering algorithm, J. King Saud Univ.-Comput. Inform. Sci. 34 (8) (2021) 6365–6376.
[195] S. Sieranoja, P. Fränti, Adapting k-means for graph clustering, Knowl. Inf. Syst. 64 (1) (2022) 115–142.
[196] J. Silva, O.B.P. Lezama, N. Varela, J.G. Guiliany, E.S. Sanabria, M.S. Otero, V.Á. Rojas, U-control chart based differential evolution clustering for determining the number of clusters in K-means, in: International Conference on Green, Pervasive, and Cloud Computing, Springer, Cham, 2019, pp. 31–41.
[197] K.P. Sinaga, M.S. Yang, Unsupervised K-means clustering algorithm. *IEEE*, Access 8 (2020) 80716–80727.
[198] K.P. Sinaga, I. Hussain, M.S. Yang, Entropy K-means clustering with feature reduction under unknown number of clusters, IEEE Access 9 (2021) 67736–67751.
[199] S. Singh, S.S. Gill, Analysis and study of K-means clustering algorithm, Internat. J. Eng. Res. Technol. 2 (2013).
[200] A. Singh, J.C. Mehta, D. Anand, P. Nath, B. Pandey, A. Khamparia, An intelligent hybrid approach for hepatitis disease diagnosis: Combining enhanced K-means clustering and improved ensemble learning, Expert. Syst. 38 (1) (2021) e12526.
[201] A. Sinha, P.K. Jana, A hybrid MapReduce-based K-means clustering using genetic algorithm for distributed datasets, J. Supercomput. 74 (4) (2018) 1562–1579.
[202] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques KDD workshop on text mining, Department of Computer Science/Army HPC Research Center, 2000.
[203] H. Steinhaus, Sur la division des corps matériels en parties, Bulletin de l'Académie Polonaise des Sciences. Classe 3 (12) (1956) 801–804.
[204] D. Steinley, M.J. Brusco, Initializing K-means batch clustering: A critical evaluation of several techniques, J. Classif. 24 (2007) 99–121.
[205] D. Steinley, K-means clustering: a half-century synthesis, Br. J. Math. Stat. Psychol. 59 (1) (2006) 1–34.
[206] U. Stemmer, H. Kaplan, Differentially private K-means with constant multiplicative error, Adv. Neural Inf. Proces. Syst. 31 (2018).
[207] A. Tayal, A. Solanki, S.P. Singh, Integrated framework for identifying sustainable manufacturing layouts based on big data, machine learning, meta-heuristic, and data envelopment analysis, Sustain. Cities Soc. 62 (2020) 102383.
[208] D. Van-Hieu, P. Meesad, Fast K-means clustering for very large datasets based on mapreduce combined with a new cutting method, in: Knowledge and Systems Engineering, Springer, Cham, 2015, pp. 287–298.
[209] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin, Y. Theodoridis, State-of-the-art in privacy preserving data mining, ACM SIGMOD Rec. 33 (1) (2004) 50–57.
[210] Vij, R., & Kumar, S. (2012). Improved K-means clustering algorithm for two-dimensional data. In *Proceedings of the Second International Conference on Computational Science, Engineering, and Information Technology*, 665-670.
[211] A. Vijayaraghavan, A. Dutta, A. Wang, Clustering stable instances of Euclidean K-means, Adv. Neural Inf. Proces. Syst. 30 (2017).
[212] J. Wang, J. Wang, Q. Ke, G. Zeng, S. Li, Fast approximate K-means via cluster closures, in: IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3037–3044, https://doi.org/10.1109/CVPR.2012.6248034.
[213] Wang, B., Lv, Z., Zhao, J., Wang, X., & Zhang, T. (2016). An adaptively disperse centroids K-means algorithm based on mapreduce model. In *2016 12th International Conference on Computational Intelligence and Security (CIS), IEEE*, 142-146.
[214] Wang, J., & Su, X. (2011). An improved K-means clustering algorithm. In *2011 IEEE 3rd international conference on communication software and networks, IEEE*, 44-46.

[215] X. Wang, C. Shao, S. Xu, S. Zhang, W. Xu, Y. Guan, Study on the location of private clinics based on K-means clustering method and an integrated evaluation model, IEEE Access 8 (2020) 23069–23081.
[216] Wei, D. (2016). A constant-factor bi-criteria approximation guarantee for K-means++. *Advances in Neural Information Processing Systems*, 29.
[217] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, S.Y. Philip, Z.H. Zhou, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1–37.
[218] F. Wu, C. Zhu, J. Xu, M.W. Bhatt, A. Sharma, Research on image text recognition based on canny edge detection algorithm and k-means algorithm, Internat. J. Syst. Assur. Eng. Manage. 13 (1) (2022) 72–80.
[219] Z. Wu, Z. Wu, An enhanced regularized K-means type clustering algorithm with adaptive weights, IEEE Access 8 (2020) 31171–31179.
[220] J. Xiao, Y. Yan, J. Zhang, Y. Tang, A quantum-inspired genetic algorithm for K-means clustering, Expert Syst. Appl. 37 (7) (2010) 4966–4973.
[221] T. Xie, R. Liu, Z. Wei, Improvement of the fast-clustering algorithm improved by K-means in the big data, Appl. Math. Nonlinear Sci. 5 (1) (2020) 1–10.
[222] C. Xiong, Z. Hua, K. Lv, X. Li, An improved K-means text clustering algorithm by optimizing initial cluster centers, in: 2016 7th International Conference on Cloud Computing and Big Data (CCBD), IEEE, 2016, pp. 265–268.
[223] P. Xiong, L.I.U. Hu, T.I.A.N. Yongliang, C.H.E.N. Zikun, W.A.N.G. Bin, Y.A.N.G. Hao, Helicopter maritime search area planning based on a minimum bounding rectangle and K-means clustering, Chin. J. Aeronaut. 34 (2) (2021) 554–562.
[224] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Ann. Data Sci. 2 (2) (2015) 165–193.
[225] L. Xu, A. Krzyzak, E. Oja, Rival penalized competitive learning for clustering analysis, RBF net, and curve detection, IEEE Trans. Neural Netw. 4 (4) (1993) 636–649.
[226] Q. Xu, C. Ding, J. Liu, B. Luo, PCA-guided search for K-means, Pattern Recogn. Lett. 54 (2015) 50–55.
[227] J. Yang, J. Wang, Tag clustering algorithm LMMSK: Improved K-means algorithm based on latent semantic analysis, J. Syst. Eng. Electron. 28 (2) (2017) 374–384.
[228] Yang, K. C., & Chao, W. P. (2020). Applying K-means technique and decision tree analysis to predict Taiwan ETF performance. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE*, 635-639.
[229] M.S. Yang, K.P. Sinaga, A feature-reduction multi-view K-means clustering algorithm, IEEE Access 7 (2019) 114472–114486.
[230] M. Yang, I. Tjuawinata, K.Y. Lam, K-means clustering with local d χ-privacy for privacy-preserving data analysis, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2524–2537.
[231] Ye, J., Zhao, Z., & Wu, M. (2007). Discriminative K-means for clustering. *Advances in Neural Information Processing Systems, 20*.
[232] Yuan, F., Meng, Z. H., Zhang, H. X., & Dong, C. R. (2004). A new algorithm to get the initial centroids. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826),* 2, 1191-1193.
[233] D.Q. Zeebaree, H. Haron, A.M. Abdulazeez, S.R. Zeebaree, Combination of K-means clustering with genetic algorithm: A review, Int. J. Appl. Eng. Res. 12 (24) (2017) 14238–14245.
[234] Zha, H., He, X., Ding, C., Gu, M., & Simon, H. (2001). Spectral relaxation for K-means clustering. *Advances in Neural Information Processing Systems, 14*.
[235] G. Zhang, Y. Li, X. Deng, K-Means clustering-based electrical equipment identification for smart building application, Information 11 (1) (2020) 27.
[236] Zhang, H., & Zhou, X. (2018). A novel clustering algorithm combining niche genetic algorithm with canopy and K-means. *In 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD),* IEEE, 26-32.
[237] P. Zhang, T. Huang, X. Sun, W. Zhao, H. Liu, S. Lai, J.K. Liu, Privacy-Preserving and Outsourced multi-party K-means clustering based on multi-key fully homomorphic encryption, IEEE Trans. Dependable Secure Comput. (2022).
[238] T. Zhang, F. Ma, Improved rough K-means clustering algorithm based on weighted distance measure with Gaussian function, Int. J. Comput. Math. 94 (4) (2017) 663–675.
[239] Zhang, Y., Zhang, D., & Shi, H. (2012). K-means clustering based on self-adaptive weight. In *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology, IEEE*, 1540-1544.
[240] Z. Zhang, K. Lange, J. Xu, Simple and scalable sparse K-means clustering via feature ranking, Adv. Neural Inf. Proces. Syst. 33 (2020) 10148–10160.
[241] W. Zhao, H. Ma, Q. He, Parallel K-means clustering based on mapreduce, in: IEEE international conference on cloud computing, Springer, Berlin, 2009, pp. 674–679.
[242] X. Zhao, F. Nie, R. Wang, X. Li, Improving projected fuzzy K-means clustering via robust learning, Neurocomputing 491 (2022) 34–43.
[243] Y. Zhou, H. Wu, Q. Luo, M. Abdel-Baset, Automatic data clustering using nature-inspired symbiotic organism search algorithm, Knowl.-Based Syst. 163 (2019) 546–557.
[244] X. Zhou, J. Gu, S. Shen, H. Ma, F. Miao, H. Zhang, H. Gong, An automatic K-means clustering algorithm of GPS data combining a novel niche genetic algorithm with noise and density, ISPRS Int. J. Geo Inf. 6 (12) (2017) 392.
[245] C. Zhu, C.U. Idemudia, W. Feng, Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques, Inf. Med. Unlocked 17 (2019) 100179.
[246] E. Zhu, R. Ma, An effective partitional clustering algorithm based on new clustering validity index, Appl. Soft Comput. 71 (2018) 608–621.
[247] Y. Zhu, X. Li, Privacy-preserving K-means clustering with local synchronization in peer-to-peer networks, Peer-to-Peer Networking and Applications 13 (6) (2020) 2272–2284.
[248] Zhu, Z., & Liu, N. (2021). Early warning of financial risk based on K-means clustering algorithm. *Complexity*, *2021*.
[249] Zhuang, Y., Mao, Y., & Chen, X. (2016). A limited-iteration bisecting K-means for fast clustering large datasets. In *2016 IEEE Trustcom /BigDataSE /ISPA*, 2257-2262.
[250] M. Zubair, M.D. Iqbal, A. Shil, M.J.M. Chowdhury, M.A. Moni, I.H. Sarker, An improved K-means clustering algorithm towards an efficient data-drivenmModeling, Annals of Data Science 2022 (2022), https://doi.org/10.1007/s40745-022-00428-2.