Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5521119, 12 pages
https://doi.org/10.1155/2021/5521119

*Research Article*

# Novel Automated K-means++ Algorithm for Financial Data Sets

**Guoyu Du**,[1] **Xuehua Li**,[1] **Lanjie Zhang**,[1] **Libo Liu**,[2] **and Chaohua Zhao**[2]

[1]*Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100083, China*
[2]*Beijing Sino-Bridge Technology LTD, Beijing 100089, China*

Correspondence should be addressed to Xuehua Li; lixuehua@bistu.edu.cn

Received 27 January 2021; Revised 2 April 2021; Accepted 26 April 2021; Published 5 May 2021

Academic Editor: Weifeng Pan



The K-means algorithm has been extensively investigated in the field of text clustering because of its linear time complexity and adaptation to sparse matrix data. However, it has two main problems, namely, the determination of the number of clusters and the location of the initial cluster centres. In this study, we propose an improved K-means++ algorithm based on the Davies-Bouldin index (DBI) and the largest sum of distance called the SDK-means++ algorithm. Firstly, we use the term frequency-inverse document frequency to represent the data set. Secondly, we measure the distance between objects by cosine similarity. Thirdly, the initial cluster centres are selected by comparing the distance to existing initial cluster centres and the maximum density. Fourthly, clustering results are obtained using the K-means++ method. Lastly, DBI is used to obtain optimal clustering results automatically. Experimental results on real bank transaction volume data sets show that the SDK-means++ algorithm is more effective and efficient than two other algorithms in organising large financial text data sets. The F-measure value of the proposed algorithm is 0.97. The running time of the SDK-means++ algorithm is reduced by 42.9% and 22.4% compared with that for K-means and K-means++ algorithms, respectively.

## 1. Introduction

Clustering is the process of dividing a data set into clusters (subsets) so that the objects in the same cluster are similar to each other and the objects in different clusters are dissimilar. Its main aim is to discover the natural grouping law in a set.

As an unsupervised data mining technique, text clustering does not require pretraining models or manual text preannotation [1]. Therefore, compared with other natural language processing algorithms, the clustering algorithm is more efficient and does not require human intervention [2]. Many clustering algorithms have been proposed. They are mainly divided into three categories, namely, overlapping/nonexclusive, partitional, and hierarchical [3]. Amongst them, partition-based algorithms are widely used in various fields because of their easy implementation [4]. The most typical partitional method is K-means [2]. The K-means algorithm can adapt to sparse matrix data sets, and it is efficient in organising large data sets. However, the number of clusters and the selection of initial centres have a huge impact on the clustering results of the K-means algorithm. Setting an inappropriate initial value can easily cause the algorithm to fall into a local optimum.

Several algorithms propose improved similarity measurement methods to adapt to different data types. Wang et al. [4] utilised knowledge graphs to optimise the calculation of the similarity of text data types, which effectively improves the accuracy of text clustering. Huan et al. [5] proposed using KL divergence to calculate the similarity between cluster centres and text data objects, thereby making the K-means algorithm increasingly efficient and effective. To analyse the load profiles of smart meters, Xiang et al. [6] proposed measuring the shape characteristics of such load profiles by using the segmented slope of the load planes. Cheng et al. [7] presented a novel distance based on the common neighbourhood of dense cores and used geodesic distance to calculate the similarity between density cores. Experimental results showed that the algorithm has

good performance in clustering data sets that contain much noise. To organise clusters with complex structures, Cheng et al. [8] combined shared neighbours and local density peaks to define a new distance for describing the dissimilarity of manifold data.

To discover global optimal clustering centres, several researchers proposed combining nature-inspired optimisation algorithms to optimise the given objective function [9]. Gao et al. [10] proposed using particle swarm optimisation based on the Gaussian estimation of distribution method to update population information. Experimental results showed that the algorithm has high effectiveness and robustness. Meena and Singh [11] used the genetic algorithm and discrete difference evolution to search for the location of the global optima solution whilst reducing the number of algorithm iterations. Abualigah et al. [3] comprehensively reviewed the application of optimisation algorithms based on metaheuristics in the field of text clustering. However, the abovementioned algorithms optimise the clustering centres successively through iteration, and algorithm efficiency is not ideal.

To reduce the number of iterations and avoid falling into the local optimum, many scholars proposed optimising the selection methods of initial clustering centres directly. Wang et al. [4] used concept distance to optimise the selection of initial clustering centres and thus enhance algorithm stability. Guo et al. [12] proposed a partition-based clustering algorithm to analyse biological sequences. The algorithm eliminates noise interference by deterministically initialising cluster centres. Cheng et al. [13] optimised the decision graph of the density peak (DP) algorithm through natural neighbourhood density and graph distance. The newly defined decision graph can help the DP algorithm avoid noise interference when selecting the initial cluster centres. Other improved methods, such as the semisupervised clustering algorithm based on pairwise constraints, can enhance clustering performance. It advocates the use of prior knowledge as pairwise constraints to enable the clustering algorithm to obtain abundant heuristic information and reduce blindness in the search process [14, 15]. However, this type of algorithm has two main problems: it is unsure whether a solution that satisfies all constraints exists, and it relies too much on prior knowledge.

A possible method to reduce the number of iterations and improve the clustering quality of an algorithm is to optimise the selection method of the initial clustering centre during the initialisation phase. For example, the K-means++ algorithm selects the initial cluster centres on the basis of the farthest distance criterion. Inspired by this feature, we propose a clustering method based on the sum of the farthest distance criteria to select initial clustering centres; the proposed method is called the SDK-means++ algorithm. It can effectively describe the difference between initial cluster centres and generate the initial cluster centres in different clusters. Moreover, we use the Davies–Bouldin index (DBI) to evaluate the clustering results and obtain the optimal number of clusters. It is efficient and improves clustering accuracy when organising many data sets. In the SDK-means++ algorithm, we represent financial text data based on term frequency-inverse document frequency (TF-IDF). The similarity between data objects is calculated using cosine similarity. After that, the initial cluster centres are generated based on the maximum density and the newly proposed maximum distance sum criterion. Then, with the K-means method, the clustering result is obtained through the movement of the centre points and the change in the objects in the clusters. Finally, we automatically obtain the best results through DBI. The SDK-means++ algorithm is compared with classic K-means and K-means++ algorithms to verify the effectiveness of the proposed algorithm proposed. The experimental results on real bank data sets show that the SDK-means++ algorithm is more effective and efficient than the two other algorithms.

The main parts of this paper are organised as follows. Section 2 introduces related work on data preprocessing and classic clustering algorithms. Section 3 presents the specific steps and theoretical advantages of the proposed SDK-means++ algorithm. Section 4 introduces several methods to evaluate the results of text clustering. Section 5 presents the experiments and discussions, and Section 6 provides the conclusions of the experiments and future development directions.

## 2. Related Work

*2.1. Text Preprocessing.* The purpose of the data preprocessing stage is to represent text data information quantitatively. This stage is crucial to text clustering [16]. Many scholars have proposed extracting the themes and features of text data by using optimisation algorithms or statistics and achieved good text clustering effects [3, 16]. Generally, the preprocessing steps of text data clustering are as follows: (i) tokenisation, (ii) stop words, and (iii) feature vector space.

*2.1.1. Tokenisation.* Tokenisation is the process of converting successive text data sets into words [2, 16]. English text is composed of words and separated by special characters and spaces. However, Chinese text is based on Chinese characters. Words or phrases are formed by a variable number of Chinese characters, and the words are continuous. These two factors make it difficult to use English text tokenisation methods to process Chinese text information. Therefore, China developed various Chinese text tokenisation technologies independently. For example, the Institute of Computing Technology and Chinese Lexical Analysis System is the world's best Chinese lexical analyser developed by the China Institute of Computer Science.

*2.1.2. Stop Words.* The main purpose of removing stop words is to save storage space and improve the effectiveness of clustering algorithms [3]. Stop words have two main types, namely, function and high-frequency words. Function words often appear in documents but have no practical meaning. Typical function words, such as prepositions (e.g., "to," "for," and "in"), need to be deleted directly in the program by default. High-frequency words appear in most

documents. Because these words contain a tiny amount of information, they are difficult to distinguish in different documents. High-frequency words can be automatically removed through the calculation of inverse document frequency.

*2.1.3. Feature Vector Space.* Feature vector space was proposed based on the idea of partial matching [17]. It gives each independent item in the text a weighted performance to characterise text data sets [3]. Many weighting techniques have been proposed in the past few decades. TF-IDF is one of the most commonly used methods [4].

TF-IDF is a statistical method that is often used for information retrieval and mining. It comprehensively considers the weighting of words from local and overall aspects. The normalised calculation formula of TF-IDF is as follows:

$$w_{i,j}^{*} = \frac{tf_{i,j} * \mathrm{id}f_{i,j}}{\sqrt{\sum_{j=1}^{N} \left[tf_{i,j} * \log\left(N/n_i\right)\right]^2}}, \quad (1)$$

where $w_{i,j}^{*}$ is the normalised weight of term $i$ in document $j$. $N$ is the total number of training texts. $n_i$ is the number of texts containing feature term $i$. $tf_{i,j}$ is the ratio of the number of occurrences of term $i$ in document $j$ to the total terms in document $j$. Each TF-IDF value is stored in a two-dimensional array to form the feature vector space of the data set $S$.

*2.2. Similarity Measurement Method.* Similarity or distance is used to determine the affiliation of data objects. To date, no unified method has been proposed to measure the similarity of all data types. In accordance with the characteristics of data types, researchers use different measurement methods. Generally, text clustering algorithms adopt cosine similarity to evaluate the similarity between texts [4]. The formula of cosine similarity is written as

$$\cos\,\theta = \frac{\sum_{i=1}^{n}\left(A_i \times B_i\right)}{\sqrt{\sum_{i=1}^{n}\left(A_i\right)^2} \times \sqrt{\sum_{i=1}^{n}\left(B_i\right)^2}} = \frac{A \cdot B}{\|A\|\|B\|}, \quad (2)$$

where $A_i$ is the $i$-th eigenvector of data object $A$. According to formula (2), the value of cosine similarity is within $[-1, 1]$. However, because TF-IDF is used to characterise text data, the actual range of cosine similarity is $[0, 1]$.

*2.3. Classic Clustering Algorithms Based on Partition*

*2.3.1. K-Means Test Clustering Algorithm.* K-means is an efficient clustering algorithm based on the partition method [18]. Its first step is to set the number of clusters ($K$). The second step is to randomly generate K initial cluster centres, that is, $C = \{c_1, c_2, \ldots, c_k\}$. The third step is to assign each data object to the cluster with the highest similarity. The fourth step is to search for reasonable cluster centres through iteration. Lastly, spherical clusters are formed around the cluster centres [19]. The K-means algorithm has the advantages of easy understanding, simple implementation, high convergence speed, and adaptability to sparse matrix data [20].

Random selection of initial cluster centres results in multiple initial cluster centres in the same class, especially when the data are complex. Moreover, finding the globally optimal cluster centre through limited iterations is difficult. Therefore, the K-means algorithm easily converges to the local optimum, leading to unsatisfactory clustering results.

*2.3.2. K-Means++ Test Clustering Algorithm.* The K-means++ algorithm generates initial cluster centres based on the idea of making the distance between initial cluster centres as large as possible. Its optimisation strategy for initial cluster centres is simple. The first step is to select a data object randomly from the data set as the first cluster centre c1. The second step is to select the initial cluster centres according to the probability formula $D(x)^2/\sum_{x\in\chi} D(x)^2$ until $K$ initial cluster centres are obtained. The subsequent steps are similar to those of the K-means algorithm. Compared with the original algorithm, the K-means++ algorithm improves clustering accuracy and reduces running time. However, K-means++ is random when selecting the first initial cluster centre. Besides, it also has a certain degree of randomness to select the initial cluster centres according to the probability formula. Therefore, the clustering result is still not ideal. The flow chart of the algorithm is shown as Algorithm 1.

## 3. SDK-Means++ Algorithm

Given that classic partition-based clustering algorithms use unreasonable methods to select the initial centre; they waste much time on iterative calculation and easily fall into the local optimum. This section proposes a novel K-means++ algorithm called SDK-means++ based on the largest sum of the distance. Initially, the proposed algorithm generates a feature vector space based on TF-IDF and uses cosine similarity to calculate the vector distance. Then, the algorithm selects the initial cluster centres based on the largest sum of the distance to all existing initial cluster centres. Then, it iterates the cluster centres and obtains the clustering results with the K-means method. Afterwards, DBI is used to obtain optimal clustering results automatically. The main process of the SDK-means++ algorithm is shown in Figure 1.

*3.1. Selection of the Initial Cluster Centres.* The selected initial clustering centres have a huge impact on the clustering results. However, the farthest distance criterion proposed by the K-means++ algorithm cannot clearly reflect the overall dissimilarity of the initial cluster centres. We propose a new method to describe the dissimilarity between the initial cluster centres. The first step is to select the first initial cluster centre based on the maximum density. The second step is to select the remaining initial cluster centres on the basis of the largest sum of the distance. The calculation of the density value draws on the concept of local density in the density peak. To avoid setting parameters, we set the cut-off distance to half the mean value of the distance between data objects.

The largest number of data objects in the neighbourhood is set as the first initial cluster centre. Afterwards, the existing initial cluster centre is regarded as a whole, and the next initial cluster centre is selected in accordance with the distance from the data object to the whole. The calculation of the newly defined selection method is shown in Figure 2.

After that, the data object with the largest $w(s_j)$, $j \in [1, m]$, is selected as the next initial cluster centre.

$$c_{h+1} = \max\{w(s_1), w(s_2), \ldots, w(s_m)\}. \quad (3)$$

The new selection method links each new initial cluster centre with all existing initial cluster centres. It reflects the overall difference between the initial cluster centres, and the initial centres are generated in different classes. The difference between the proposed selection method and the classic clustering algorithm is shown in Figure 3.

According to Figure 3, the newly proposed selection method has the best effect. The K-means algorithm has three selection situations. (1) The initial cluster centres are generated in different classes (with a probability of 26.47%). (2) The two initial cluster centres are generated in the same class (with a probability of 66.18%). (3) The initial cluster centres are generated in the same class (with a probability of 7.35%). As the number of data objects and the number of clusters increase, the probability of selecting the optimal initial cluster centres decreases.

Although the K-means++ algorithm avoids all the initial cluster centres that appear in the same class, two initial cluster centres could still be in the same class. Two main factors cause the initial cluster centres to be selected each time to be inconsistent. The first reason is that the first initial cluster centre point is randomly selected. The second reason is that the selection method is based on the probability formula.

The proposed selection method based on the largest sum of the distance ensures that the initial cluster centres are generated in different clusters. The first clear initial centre point and the selection method can ensure that the same clustering result can be obtained every time.

*3.2. Obtaining the Optimal Number of Clusters.* Generally, the optimal number of clusters is unknown. However, validity indexes can be used to evaluate the results of a clustering algorithm to obtain the optimal number of clusters [21]. Validity indexes can be divided into internal and external categories [22]. Although the use of external validity indexes to evaluate clustering results is accurate, these indexes need to be combined with prior knowledge (labels) [23, 24]. Internal validity indexes can be used to evaluate clustering effects on the basis of internal information only [25].

Researchers have proposed many internal cluster validity indexes, such as between-class distance, within-class distance, DBI [26], local cores-based cluster validity (LCCV) index [27], and silhouette index [28]. Between-class distance and within-class distance are the methods used in this study to evaluate the clustering results. They are introduced in Section 4. DBI comprehensively considers the independence and cohesion of clusters. This method does not depend on the number of clusters or data partitioning methods, and it can be widely used to guide clustering algorithms. The smaller the DBI value is, the better the clustering effect is.

To avoid the problem of fuzzy inflexion points in the traditional elbow method, we search for the minimum value of DBI to obtain the optimal number of clusters. The first step is to set the number of clusters [2, $n$], where $n$ is the number of sample points. The second step is to use DBI to evaluate the clustering results in this range. To improve algorithm efficiency, we set the following condition: if the value of three consecutive DBI after DBI ($i$) is greater than DBI ($i$), then $i$ will be regarded as the optimal number of clusters. The calculation formula of DBI is written as

$$\mathrm{DBI} = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\left(\frac{\mathrm{avg}(\mathrm{Cluster}_i) + \mathrm{avg}(\mathrm{Cluster}_j)}{d_{\mathrm{cen}}(u_i, u_j)}\right), \quad (4)$$

where $\mathrm{avg}(\mathrm{Cluster}_i)$ represents the average distance between data objects and the cluster centre in the $i$-th cluster. $d_{\mathrm{cen}}(u_i, u_j)$ represents the distance between the cluster centre in the $i$-th cluster and the cluster centre in the $j$-th cluster. The SDK-means++ algorithm is shown as Algorithm 2.

## 4. Validation Techniques

The clustering evaluation method is crucial for the analysis of clustering results. It is divided into internal and external validity indexes. The main difference is whether the data are labelled. This section introduces and analyses several classic validity indexes.

*4.1. Internal Validity Indexes.* Many internal cluster validity indexes have been proposed in the past few decades. They have different characteristics in accordance with the definition of varying clustering concepts. Amongst them, between-class distance and within-class distance are the most frequently used because of their simple and clear principles. Between-class distance describes the dissimilarity between clusters. Within-class distance can measure the cohesion of data objects in the cluster. They are used to evaluate the clustering results of the experiments in this study. The calculation formulas of between-class and within-class distance are, respectively, written as follows:

$$L = \sum_{i=1}^{K} |x_i - \overline{x}|, \quad (5)$$

where $L$ is the between-class distance, $x_i$ is the mean of all data objects in the $i$-th class, and $\overline{x}$ is the mean value of all data objects.

$$\mathrm{SE} = \sum_{i=1}^{K}\sum_{p\in N_j}[1 - \cos(p, m_i)] = \sum_{i=1}^{K}\sum_{p\in N_j} d(p, m_i), \quad (6)$$

where $m_i$ is the cluster centre of the data object $p$, $\cos(p, m_i)$ is the cosine similarity between $p$ and $m_i$, and $d(p, m_i)$ is the cosine distance.

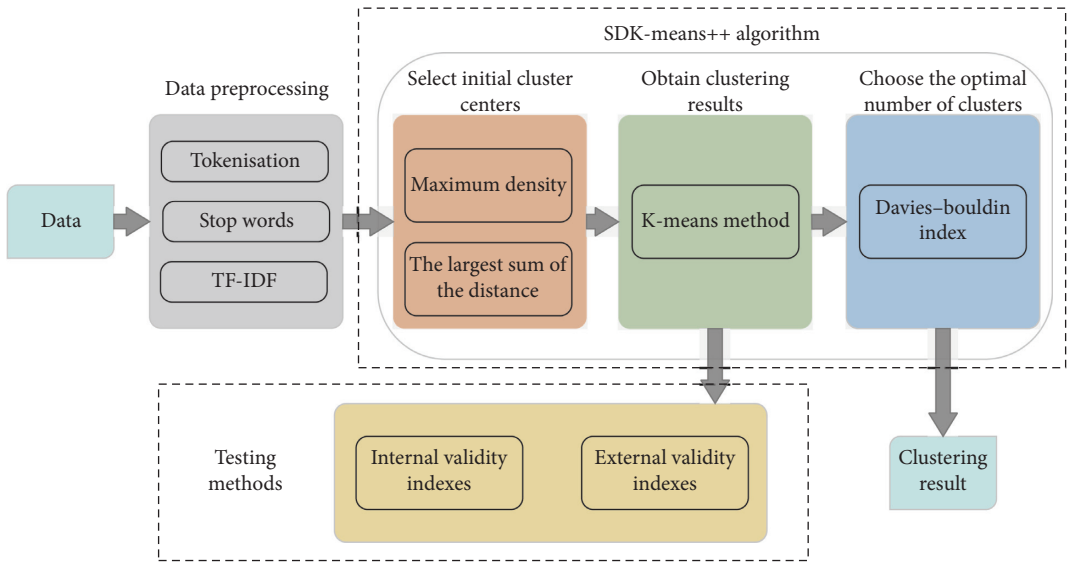


Figure 1: Brief flow chart of SDK-means++ algorithm.

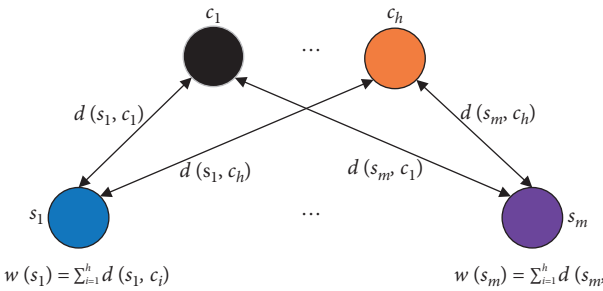


Figure 2: The sum of the distances from the data object $s_j$ to the cluster centre $c_i$ is used as the basis for $s_j$ to become the initial cluster centre.

*4.2. External Validity Indexes.* External validity indexes can accurately evaluate clustering results when the data are labelled. External validity indexes can be divided into different types, such as matching-based approach, entropy, and pair-counting measure (Rand and Jaccard indexes) [29, 30].

*4.2.1. Matching-Based Measures*

(i) Purity: it entails evaluating the purity of a cluster in accordance with the data objects that have an advantage in quantity [31]:

$$\text{purity} = \frac{1}{M}\sum_{i=1}^{K} w_i, \tag{7}$$

where $M$ is the number of all terms and $w_i$ is the number of data objects with the quantitative advantage in each cluster

(ii) Recall: it entails evaluating the clustering results from the perspective of the original data set

(iii) Precision: it entails evaluating the clustering results from the perspective of the clustering result

(iv) F-measure: it combines recall and precision to reflect the quality of clustering

Recall, precision, and F-measure are all based on confusion matrix. The confusion matrix is shown in Table 1.

TP and TN on the diagonal of the confounding matrix represent the correct clustering results, and FP and TN on the off-diagonal are misjudged.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{9}$$

$$F - \text{measure} = 2 \times \frac{\text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}. \tag{10}$$

*4.2.2. Entropy-Based Measures*

(i) Entropy: entropy is essentially a mathematical measure of uncertainty [32].

$$E = \sum_{i=1}^{K} \frac{M_i}{M} E_j = \sum_{i=1}^{K} \frac{M_i}{M} \left( -\sum_{j=1}^{L} \frac{M_{ij}}{M_i} \log\left(\frac{M_{ij}}{M_i}\right)\right), \tag{11}$$

where $M$ is the number of all data objects involved in the entire clusters, $M_i$ represents the number of all

data objects in the $i$-th cluster, and $M_{ij}$ represents the number of data objects belonging to the $j$-th class in the $i$-th cluster.

*4.2.3. Pairwise and Counting Measures*

(i) Rand index: it represents the proportion of documents in the data set which are correctly clustered.

(ii) Jaccard index: it handles asymmetric binary variables [33].

$$\text{Rand} - \text{index} = \frac{a+d}{N(N-1)/2} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \quad (12)$$

$$\text{Jaccard} - \text{index} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}. \quad (13)$$

Detailed information on internal and external validity indexes is shown in Table 2.

## 5. Experimental Results and Discussion

We conduct experiments on five bank data sets with different data volumes to evaluate the performance of the proposed SDK-means++ algorithm. The experimental data belong to the transaction volume data sets in the Business Performance Centre (BPC). The data set is mainly divided into mobile banking, online banking, WeChat, and financial products. These divisions can continue to be classified; for example, financial products include stocks, bonds, funds, and other wealth management products. We compare SDK-means++ with typical partition-based clustering algorithms, namely, K-means and K-means++.

In the experiment, two internal validity indicators and seven external validity indicators are used to evaluate the clustering results. We conduct experiments on a notebook computer with an Intel Core i7-9750H processor at 2.60 GHz, 16 GB of RAM, Windows 10 OS, and JAVA 1.8.0_231.

*5.1. The BPC Data Set Preprocessing.* The BPC data set needs to be preprocessed to characterise text data. Firstly, tokenisation is used to convert text information into terms. Secondly, the function and high-frequency words in the terms are removed. Lastly, the terms are given weights through TF-IDF calculation. The detailed information on the BPC data set is shown in Table 3.

As shown in Table 3, the amount of data gradually increases from S_1 to S_5. Classes denote the optimal number of clusters. Documents refer to the number of documents in the data set. Terms refer to the words in each data set after tokenization. Unique terms are the keywords, that is, the dimensions of the vector space model. The feature vector space constructed by data set S_1 is shown in Table 4.

The number of elements with a value of 0 in the matrix is larger than the number of nonzero elements, and the distribution of nonzero elements is irregular. Therefore, the feature vector space we constitute is a sparse matrix. The adaptability of the K-means algorithm to sparse matrix data is one of the main reasons we select it.

*5.2. Analysis of Algorithm Performance*

*5.2.1. Obtaining the Optimal Number of Clusters.* This experiment uses DBI to obtain the best clustering results and avoid the problem of fuzzy inflexion points in the elbow method. It indirectly obtains the optimal number of clusters through the evaluation of the clustering results. The experiment uses data set S_1, and the theoretical optimal number of clusters is 7.

In data set S_1, classes 4 and 6 are similar. Although most of the data formats in their text messages are the same, the crucial gateway and alarm messages are different. Given the fact that K-means and K-means++ cannot distinguish between classes 4 and 6, a more reasonable method is to select 6 as the number of clusters, as reflected in Figure 4. The proposed SDK-means++ can distinguish 7 clusters perfectly, so the best clustering result is obtained when the number of clusters is 7. The experiment shows that using any partition-based clustering method can approximate the optimal number of clusters through DBI.

*5.2.2. Verification and Analysis of Algorithm Effectiveness.* To verify the effectiveness of the algorithm, we compare and analyse the experimental results of SDK-means++, K-means, and K-means++. The experiment is divided into two parts, namely, evaluations of internal and external validity indexes. The first set of experiments uses internal validity indexes to evaluate the clustering results of data set S_1 with different clustering numbers. The clustering numbers of the three algorithms are set to the maximum range during the experiment [2, 11].

As shown in Figure 5, when clustering number $K$ is close to the optimal clustering number 7, SDK-means++ has a large between-class distance value and a small within-class distance value. The experimental results show that the proposed SDK-means++ exhibits good performance in dissimilarity between classes and cohesion within classes.

The second set of experiments uses external validity indexes to evaluate the clustering results of all data sets in Table 3. The number of clusters of the three algorithms is set to the optimal number of clusters 7 to determine the best clustering performance.

The experimental results in Table 5 show that the proposed SDK-means++ performs effectively in all the data sets. Data set S_5 has the largest amount of data, which represents a complex data situation. SDK-means++ in data set S_5 has the best performance in the evaluation of seven external validity indexes. The purity value is 0.89, recall is 0.32, precision is 0.88, F-measure is 0.47, the Rand index is 0.82, the Jaccard index is 0.31, and entropy is 0.43. Figure 5 and Table 5 show that the clustering accuracy of SDK-means++ is better than that of K-means++, and K-means++ is better than K-means. However, the stability of the proposed method is not good. The main reason is that the partition-based clustering methods are difficult to distinguish classes

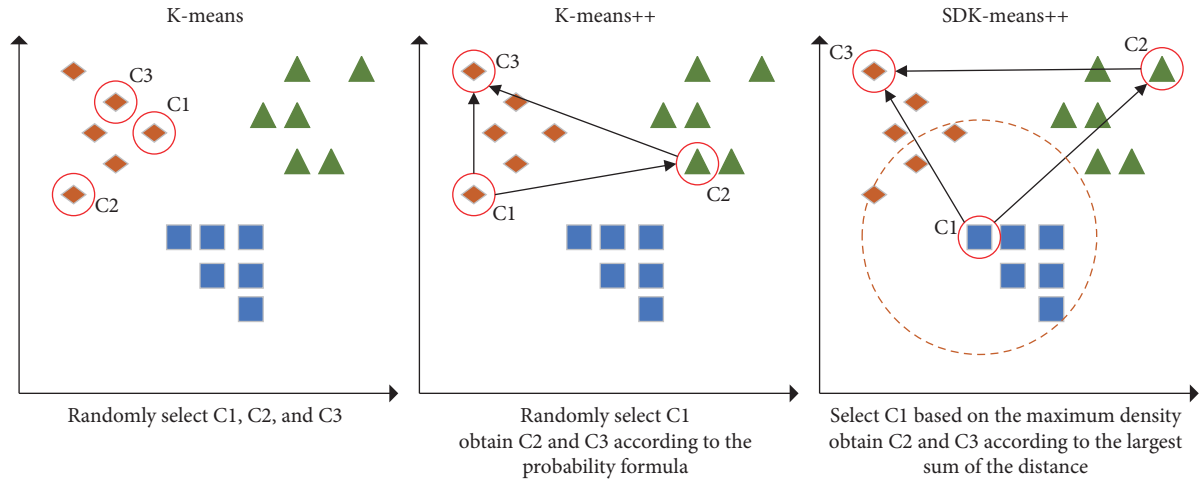


Figure 3: Schematic diagram of the difference between the three algorithms to select the initial clustering centres.

```
Input: k, dataset
Output: center, texts
  function CLUSTERING
    center = null
    texts = null
    m = the number of data objects in the dataset
    for i = 0 ⟶ k do
      if i == 0 then
        temp = random(m)
        center[i] = dataset[temp]
      else
        for j = 0 ⟶ m do
          for h = 0 ⟶ i do
            distance[j][h] = cosine distance of the dataset[j] and center[h]
          end for
        end for
        Min[j] = min(distance[j])
        Sum + = Min[j]
        Random_number = random(Sum)
        for j = 0 ⟶ m do
          Random_number = Random_number - Min[j]
          if Random_number < 0 then
            cluster[i] = dataset[j]
          end if
        end for
        2–4. Proceed as with the standard K-means algorithm
      end if
    end for
    return center, texts
  end function
```

Algorithm 1: K-means++.

with complex structures and high-dimensional data sets. In the future, we will continue to improve on this shortcoming.

When data sets S_3, S_4, and S_5 are used, the evaluation results of the external validity indexes are inferior to those when data sets S_1 and S_2 are used for three main reasons. Firstly, as the amount of data increases, the differences between clusters are reduced. Secondly, the K-means algorithms find it difficult to distinguish classes in complex data sets. Thirdly, the experimental data sets use gateway information as labels instead of real category information.

```
Input: k, dataset
Output: center, texts
  function CLUSTERING
    center = null
    texts = null
    m = the number of data objects in the dataset
    max = 0
    for i = 0 ⟶ k do
      if i == 0 then
        temp = maximum density(m)
        center[i] = dataset[temp]
      else
        for j = 0 ⟶ m do
          for h = 0 ⟶ i do
            distance[j][h] = cosine distance of the dataset[j] and center[h]
          end for
        end for
        for j = 0 ⟶ m do
          for h = 0 ⟶ i do
            Sum_distance[j] + = distcance[j][h]
          end for
        end for
        for j0 ⟶ m do
          if Sum_distance[j] > max then cluster[i] = dataset[j]
          end if
        end for
        2–4. Proceed as with the standard K-means algorthm
      end if
    end for
    return center, texts
  end function
```

Algorithm 2: SDK-means++.

Table 1: Confusion matrix for clustering.

| | Same cluster | Different cluster |
|---|---|---|
| Similar documents | True positive (TP) | False negative (FN) |
| Different documents | False positive (FP) | True negative (TN) |

Table 2: Detailed information about validation techniques.

| Validation methods | Internal validity indexes | | External validity indexes | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Between-class distance | Within-class distance | Purity | Recall | Precision | F-measure | Rand index | Jaccard index | Entropy |
| Range | [0,+∞] | [0,+∞] | [0,1] | [0,1] | [0,1] | [0,1] | [0,1] | [0,1] | [0,1] |
| Perfect clustering results | ⟶ + ∞ | ⟶0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Table 3: Details of the BPC data set.

| Version | Classes | Documents | Terms | Unique terms |
|---|---|---|---|---|
| S_1 | 7 | 100 | 2180 | 442 |
| S_2 | 9 | 150 | 3006 | 607 |
| S_3 | 12 | 1800 | 41700 | 4373 |
| S_4 | 15 | 3763 | 191627 | 9566 |
| S_5 | 14 | 9172 | 193838 | 10364 |

Table 4: Vector space mode.

| Document | Keyword | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | ⋯ | 441 | 442 |
| 1 | 0.0382 | 0.0096 | 0.0191 | 0.0191 | ⋯ | 0.0000 | 0.0000 |
| 2 | 0.0000 | 0.0101 | 0.0200 | 0.0200 | ⋯ | 0.0000 | 0.0000 |
| 3 | 0.0000 | 0.0102 | 0.0203 | 0.0203 | ⋯ | 0.0000 | 0.0000 |
| 4 | 0.0000 | 0.0100 | 0.0199 | 0.0199 | ⋯ | 0.0000 | 0.0000 |
| ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ |
| 99 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | ⋯ | 0.1285 | 0.0000 |
| 100 | 0.0000 | 0.0153 | 0.0000 | 0.0000 | ⋯ | 0.0000 | 0.1221 |

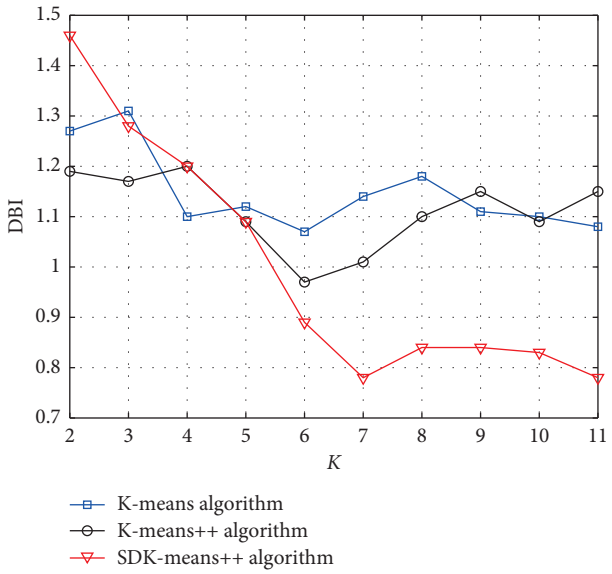


Figure 4: Davies–Bouldin index (DBI) in internal validity indexes.

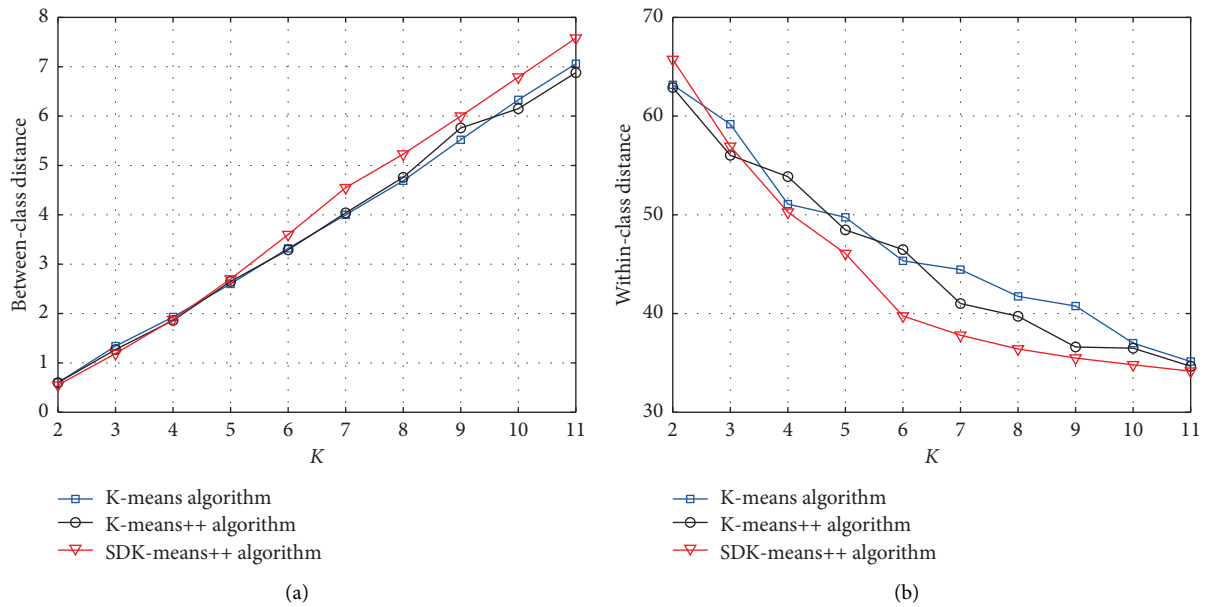


Figure 5: Evaluate the quality of clustering according to internal validity indexes.

Table 5: Evaluate the quality of clustering according to external validity indexes.

| Dataset and algorithm | Set matching measures | | | | Pair-counting measures | | Entropy |
|---|---|---|---|---|---|---|---|
| | Purity | Recall | Precision | F-measure | Rand-index | Jaccard-index | |
| S_1 K-means | 0.87 | 0.68 | 0.62 | 0.65 | 0.84 | 0.48 | 0.62 |
| S_1 K-means++ | 0.88 | 0.78 | 0.78 | 0.78 | 0.89 | 0.60 | 0.46 |
| S_1 SDK-means++ | **0.98** | **0.96** | **0.98** | **0.97** | **0.92** | **0.95** | **0.06** |
| S_2 K-means | 0.82 | 0.80 | 0.70 | 0.75 | 0.92 | 0.59 | 0.57 |
| S_2 K-means++ | 0.86 | 0.83 | 0.84 | 0.83 | 0.92 | 0.71 | 0.31 |
| S_2 SDK-means++ | **0.97** | **0.97** | **0.98** | **0.97** | **0.99** | **0.96** | **0.08** |
| S_3 K-means | 0.87 | 0.26 | 0.83 | 0.40 | 0.65 | 0.25 | 0.49 |
| S_3 K-means++ | 0.88 | 0.28 | 0.86 | 0.42 | 0.66 | 0.26 | 0.47 |
| S_3 SDK-means++ | **0.90** | **0.33** | **0.88** | **0.48** | **0.69** | **0.32** | **0.38** |
| S_4 K-means | 0.86 | 0.38 | 0.83 | 0.52 | 0.83 | 0.35 | 0.56 |
| S_4 K-means++ | 0.86 | 0.36 | 0.83 | 0.50 | 0.83 | 0.34 | 0.52 |
| S_4 SDK-means++ | **0.90** | **0.49** | **0.91** | **0.64** | **0.87** | **0.47** | **0.41** |
| S_5 K-means | 0.85 | 0.29 | 0.77 | 0.42 | 0.80 | 0.27 | 0.53 |
| S_5 K-means++ | 0.88 | 0.29 | 0.80 | 0.43 | 0.80 | 0.27 | 0.49 |
| S_5 SDK-means++ | **0.89** | **0.32** | **0.88** | **0.47** | **0.82** | **0.31** | **0.43** |
| RMSE & K-means | 0.018 | 0.217 | 0.081 | 0.134 | 0.088 | 0.129 | 0.043 |
| RMSE & K-means++ | 0.009 | 0.244 | 0.028 | 0.176 | 0.091 | 0.184 | 0.073 |
| RMSE & SDK-means++ | **0.038** | **0.292** | **0.042** | **0.223** | **0.101** | **0.293** | **0.165** |

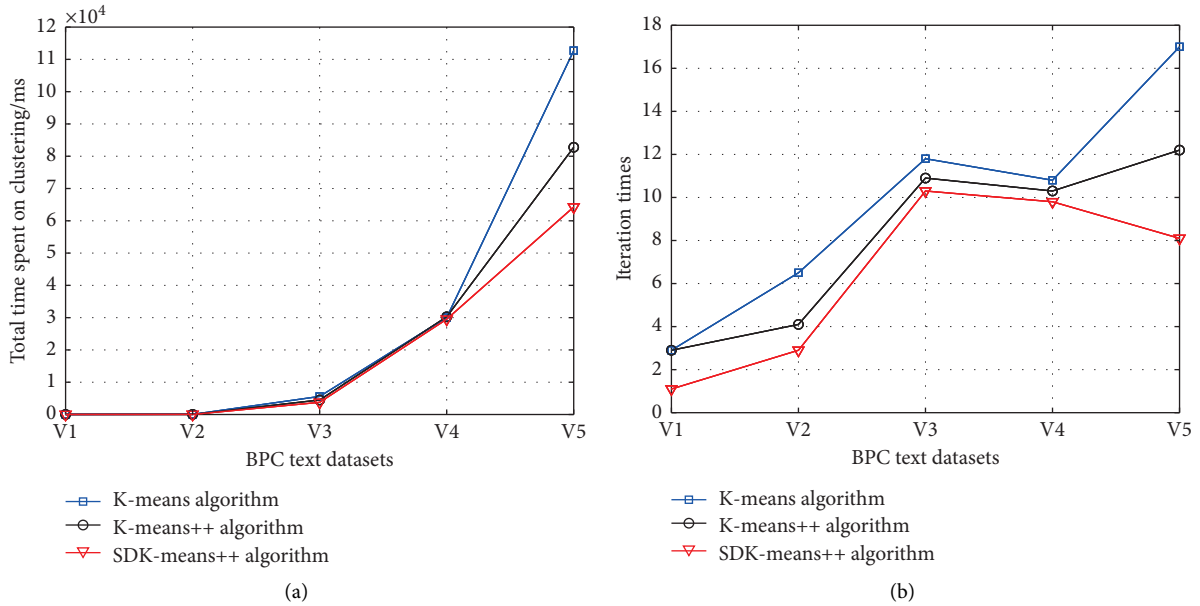


Figure 6: Evaluate the efficiency of the SDK-means algorithm according to the running time and the number of iterations.

*5.2.3. Verification and Analysis of Algorithm Efficiency.* In this section, the efficiency of the proposed algorithm is verified on all data sets in Table 3. Clustering number $K$ is set as the optimal number of clusters in accordance with Table 3. Afterwards, the running time and number of iterations of the three clustering algorithms are tested.

When dealing with massive amounts of data, the total time spent by the SDK-means++ algorithm on clustering is lower than that spent by the two other clustering algorithms, as shown in Figure 6. The results prove that the SDK-means++ algorithm has a significant improvement in time complexity. The iteration times experiment shows that, in most data sets, the SDK-means++ algorithm has fewer iterations compared with the other two algorithms. When dealing with data set S_5, the number of iterations of SDK-means++ is reduced by 47% and 26% compared with those for K-means and K-means++, respectively. Figure 6 proves the efficiency of SDK-means++ in organising massive data.

## 6. Conclusion

Classic K-means and K-means++ algorithms have randomness when selecting the initial clustering centres, resulting in unstable clustering results, easy detainment into

local optima, and large number of iterations. Moreover, the number of clusters needs to be set manually. This study proposes a new K-means++ algorithm called SDK-means++ based on the largest sum of the distance and DBI to solve these shortcomings. The algorithm selects the first initial cluster centre based on the maximum density value and selects the remaining initial cluster centres based on the largest sum of the distance. This selection method makes the result of each initialisation the same. Then, the K-means method is used to obtain the clustering results. Afterwards, the best clustering result is automatically obtained through DBI. The experimental results show that the proposed SDK-means++ algorithm outperforms the classic partition-based method in terms of effectiveness and efficiency.

SDK-means++ has two limitations. Firstly, many invalid features are extracted, which increases the amount of calculation and affects clustering accuracy. Secondly, partition-based clustering methods find it difficult to identify classes with complex structure data sets. Therefore, our future research will mainly include mining representative feature words in text data sets through heuristic optimisation algorithms [34–36]. In addition, density peaks will be combined to find cluster centres quickly in complex high-dimensional data sets [13].

## Data Availability

The original data set involved in this study cannot be shared because the bank information is confidential.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the enterprise entrusted project under Grant no. K2020004 and in part by the Foundation of Beijing Information Science and Technology University under Grant no. 2025020.

## References

[1] M. Alhawarat and M. Hegazi, "Revisiting K-means and topic modeling, a comparison study to cluster Arabic documents," *IEEE Access*, vol. 6, pp. 42740–42749, 2018.

[2] Y. Fan, L. Gongshen, M. Kui, and S. Zhaoying, "Neural feedback text clustering with BilSTM-CNN-Kmeans," *IEEE Access*, vol. 6, pp. 57460–57469, 2018.

[3] L. Abualigah, A. H. Gandomi, and M. A. Elaziz, "Advances in meta-heuristic optimization algorithms in big data text clustering," *Electronics*, vol. 10, no. 2, p. 101, 2021.

[4] X. Wang, Y. Li, M. Wang, Z. Yang, and H. Dong, "An improved K_means algorithm for document clustering based on knowledge graphs," in *Proceedings of the 2018 11th International Congress on Image and Signal Processing*, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, October 2018.

[5] Z. Huan, Z. Pengzhou, and G. Zeyang, "K-means text dynamic clustering algorithm based on KL divergence," in *Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Singapore, June 2018.

[6] Y. Xiang, J. Hong, Z. Yang et al., "Slope-based shape cluster method for smart metering load profiles," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1809–1811, 2020.

[7] D. Cheng, J. Huang, S. Zhang, X. Zhang, and X. Luo, "A novel approximate spectral clustering algorithm with dense cores and density peaks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 40, pp. 1–13, 2021.

[8] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "Clustering with local density peaks-based minimum spanning tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 374–387, 2021.

[9] L. Abualigah, A. H. Gandomi, M. A. Elaziz et al., "Nature-inspired optimization algorithms for text document clustering-a comprehensive analysis," *Algorithms*, vol. 13, no. 12, p. 345, 2020.

[10] H. Gao, Y. Li, P. Kabalyants, H. Xu, and R. Martínez-Béjar, "A novel hybrid PSO-K-means clustering algorithm using Gaussian estimation of distribution method and lévy flight," *IEEE Access*, vol. 8, pp. 122848–122863, 2020.

[11] K. Y. Meena and P. Singh, "Text documents clustering using genetic algorithm and discrete differential evolution," *International Journal of Computer Applications*, vol. 43, no. 1, pp. 975–8887, 2012.

[12] G. Guo, L. Chen, Y. Ye, and Q. Jiang, "Cluster validation method for determining the number of clusters in categorical sequences," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 2936–2948, 2017.

[13] D. Cheng, S. Zhang, and J. Huang, "Dense members of local cores-based density peaks clustering algorithm," *Knowledge-Based Systems*, vol. 193, Article ID 105454, 2020.

[14] D. W. Chen and Y. H. Jin, "An active learning algorithm based on shannon entropy for constraint-based clustering," *IEEE Access*, vol. 8, pp. 171447–171456, 2020.

[15] Z. Yu, P. Luo, J. Liu et al., "Semi-supervised ensemble clustering based on selected constraint projection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2394–2407, 2018.

[16] J. Rashid, S. M. Adnan Shah, A. Irtaza et al., "Topic modeling technique for text mining over biomedical text corpora through hybrid inverse documents frequency and fuzzy K-means clustering," *IEEE Access*, vol. 7, pp. 146070–146080, 2019.

[17] G. Salton, "Automatic text processing: the transformation, analysis, and retrieval of information by computer," *Addison-Wesley*, vol. 169, 1989.

[18] X. Wu, V. Kumar, J. Ross Quinlan et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281–297, 1967.

[20] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1/2, pp. 143–175, 2001.

[21] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[22] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *International Journal of Computers and Communications*, vol. 5, no. 1, pp. 27–34, 2011.

[23] Y. Lei, J. C. Bezdek, S. Romano, N. X. Vinh, J. Chan, and J. Bailey, "Ground truth bias in external cluster validity indices," *Pattern Recognition*, vol. 65, pp. 58–70, 2017.

[24] J. Wu, J. Chen, H. Xiong, and M. Xie, "External validation measures for K-means clustering: a data distribution perspective," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6050–6061, 2009.

[25] L. J. Deborah, R. Baskaran, and A. Kannan, "A survey on internal validity measure for cluster validation," *International Journal of Computer Science & Engineering Survey*, vol. 1, no. 2, pp. 85–102, 2010.

[26] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[27] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "A novel cluster validity index based on local cores," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 985–999, 2019.

[28] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[29] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[30] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

[31] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, UK, 2008.

[32] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[33] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[34] L. Abualigah, A. Diabat, and S. Mirjalili, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.

[35] L. Abualigah, "Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications," *Neural Computing and Applications*, vol. 33, no. 7, pp. 2949–2972, 2020.

[36] L. Abualigah and A. Diabat, "Advances in sine cosine algorithm: a comprehensive survey," *Artificial Intelligence Review*, vol. 54, no. 4, pp. 2567–2608, 2021.