Hindawi
Scientific Programming
Volume 2018, Article ID 9253295, 23 pages
https://doi.org/10.1155/2018/9253295

Research Article

Novel Two-Dimensional Visualization Approaches for Multivariate Centroids of Clustering Algorithms

Yunus Doğan,[1] **Feriştah Dalkılıç**,[1] **Derya Birant**,[1] **Recep Alp Kut**,[1] **and Reyat Yılmaz**[2]

[1]*Department of Computer Engineering, Faculty of Engineering, Dokuz Eylul University, Izmir, Turkey*
[2]*Department of Electrical and Electronics Engineering, Faculty of Engineering, Dokuz Eylul University, Izmir, Turkey*

Correspondence should be addressed to Yunus Doğan; yunus@cs.deu.edu.tr

Received 8 January 2018; Revised 23 June 2018; Accepted 9 July 2018; Published 8 August 2018

Academic Editor: José E. Labra

The dimensionality reduction and visualization problems associated with multivariate centroids obtained by clustering algorithms are addressed in this paper. Two approaches are used in the literature for the solution of such problems, specifically, the self-organizing map (SOM) approach and mapping selected two features manually (MS2Fs). In addition, principle component analysis (PCA) was evaluated as a component for solving this problem on supervised datasets. Each of these traditional approaches has drawbacks: if SOM runs with a small map size, all centroids are located contiguously rather than at their original distances according to the high-dimensional structure; MS2Fs is not an efficient method because it does not take features outside of the method into account, and lastly, PCA is a supervised method and loses the most valuable feature. In this study, five novel hybrid approaches were proposed to eliminate these drawbacks by using the quantum genetic algorithm (QGA) method and four feature selection methods, Pearson's correlation, gain ratio, information gain, and relief methods. Experimental results demonstrate that, for 14 datasets of different sizes, the prediction accuracy of the proposed weighted clustering approaches is higher than the traditional K-means++ clustering approach. Furthermore, the proposed approach combined with K-means++ and QGA shows the most efficient placements of the centroids on a two-dimensional map for all the test datasets.

1. Introduction

Human visual perception can be insufficient for the interpretation of a pattern within a multivariate (or high dimensional) structure, causing errors at the decision-making stage. In knowledge discovery processes, the same drawback is encountered in multivariate datasets because of not being able to print the dataset to a visual interface as a two-dimensional (2D) structure. Furthermore, inefficient features in a multivariate dataset negatively impact the accuracy and running performance of data analysis tasks. Therefore, the notion of dimensional reduction is of particular relevance in the preprocessing phase of data analysis. Many algorithms and methods have been proposed and developed for dimensional reduction [1], of which principal component analysis (PCA) is one of the most popular methods [2]. Regardless of popularity, neither PCA nor the other available dimensional reduction methods are suitably efficient to independently visualize all instances in the dataset because at the data preparation stage before the usage of the data-mining algorithm, PCA performs some feature selection tests separately for different dimensions. Thereafter, the optimal feature number and the optimal model are determined with respect to the variance values of these proposed models. The dataset is presented as a 2D map using PCA results in a low variance value. Another drawback of PCA is that the most valuable feature is transformed to new values [2].

As a result, PCA is not typically successful at mapping the dataset in 2D, except in image representation and facial recognition studies [3, 4]. Another difficulty is that even if a dimensional reduction is applied, the visualization of all instances in a large dataset causes storage and performance problems. To address this problem, clustering algorithms can be used for data summarization and the visualization methods applied afterwards. K-Means is the most-used clustering algorithm; however, K-means submits only the high-dimensional centroids, without any visualization and without any dimensional reduction operation.

Literature records three approaches to the visualization of K-means. The first approach involves mapping for two selected features from a multivariate dataset (MS2Fs). Curman et al. have clustered the coordinate data in Vancouver using K-means and presented them on a map [5]; Cicoria et al. have clustered "Titanic Passenger Data" using K-means and printed the survival information that was the goal of the study to different 2D maps according to features [6]; an obtained binary image was clustered onto a 2D interface using K-means for face detection in a study by Hadi and Sumedang [7]; and lastly, Fang et al. implemented software to visualize a dataset according to two selected features [8]. The second approach is visualization that takes place after K-means clustering of the dataset is dimensionally reduced by PCA. Nitsche has used this type of visualization for document clustering [9], and Wagh et al. have used it for a molecular biological dataset [10]. The third approach is the visualization of the dataset clustered by K-means in conjunction with an approach like neighborhood method location in the self-organizing map (SOM) technique, used by Raabe et al. to implement a new approach to show K-means clustering with a novel technique locating the centroids on a 2D interface [11]. SOM is a successful mapping and clustering algorithm; nevertheless, it relies on the map-size parameter as the cluster number, and if it runs with a small map-size value, all centroids of the clusters are located contiguously, not at their original distances according to the high-dimensional structure.

In our study, new approaches are proposed to visualize the centroids of the clusters on a 2D map, preserving the original distances in the high-dimensional structure. These different approaches are implemented as hybrid algorithms using K-means++ (an improved version of K-means), SOM++ (an improved version of SOM), and the quantum genetic algorithm (QGA). QGA was selected for use in our hybrid solutions because dimensionality reduction and visualization problems for multivariate centroids (DRV-P-MC) are also an optimization problem.

The contributions of this study are threefold: first, clustering by K-means++; then, mapping the centroids onto a 2D interface using QGA; and evaluating the success of this method. A heuristic approach is proposed for DRV-P-MC, and the aim is to avoid the drawback of locating the clusters contiguously as in the traditional SOM++. Mapping the centroids onto a 2D interface using SOM++ and evaluating the success of this approach are performed to enable comparison. Second, the usage of four major feature selection methods was mentioned in the paper by De Silva and Leong [12], specifically relief, information gain, gain ratio, and correlation. The aim is to preserve the most valuable feature and to evaluate it as the X axis. Additionally, the Y axis is obtained by a weighted calculation using the coefficient values returned from these feature selection methods. This provides an alternative to PCA for generating a 2D dataset that avoids the PCA drawback of losing the most valuable feature. Then, clustering the datasets by K-means++ and mapping the centroids by these novel approaches separately onto a 2D interface are performed. Moreover, mapping the centroids onto a 2D interface using PCA and evaluating the success of this approach for comparative purposes are performed. Third, a versatile tool is implemented with the capability to select the desired file, algorithm, normalization type, distance metric, and size of the 2D map to calculate successes across six different metrics: "sum of the square error," "precision," "recall," "f-measure," "accuracy," and "difference between multivariate and 2D structures." These metrics are formulated in detail in Problem Definition.

In literature, generally, MS2Fs has been the preferred method to manually determine the relations between features. Our tool is not only built on MS2Fs but also contains another algorithm that maps the centroids by using information gain, for comparison with our novel approaches. It assumes that the X axis is the most valuable feature, followed by the Y axis, according to information gain scores.

This paper details our study in seven sections. DRV-P-MC is defined in detail in Section 2; related works providing guidance on DRV-P-MC, including traditional algorithms, hybrid approaches, and the notion of dimensionality reduction, are submitted in Section 3; in Sections 4 and 5, the reorganized traditional approaches and the proposed algorithms are formulated and presented in detail; the experimental studies performed by using 14 datasets with different characteristics and their accuracy results are given in Section 6; and finally, Section 7 presents conclusions about the proposed methods.

2. Problem Definition

Considering that K-means++ is the major clustering algorithm used in this sort of problem, the significance of pattern visualization in decision-making is clear. However, the visualization requirement for the centroids of K-means++ uncovers the need for DRV-P-MC, since the centroids returned and the elements in them are presented by the traditional K-means++ irrespective of any relation among the centroids. The aim in solving this problem is to map the centroids onto a 2D interface by attaining the minimum difference among the multivariate centroids and their obtained 2D projections.

DRV-P-MC can be summarized as obtaining E from C. To detect whether the solution E of this problem is optimal or not, ℓ must be measured as zero or a value close to zero.

Two theorems and their proofs are described in this section with an illustration for the measurement of ℓ.

2.1. Theorem 1. In measuring the distance between two matrices like S and T, performing the divisions of the values in S and T by the minimum distances in the matrices separately avoids the revealed numeric difference owing to the dimensional difference between E and C, and thus, a proportional balance between S and T is supplied as in (1) and (2):

$$S \longleftarrow \sum_{i=1}^{k} \sum_{j=i+1}^{k} S_{i,j} = \sqrt{\sum_{a=0}^{f} \left(C_{i,a} - C_{j,a}\right)^2},$$
$$S \longleftarrow \sum_{i=0}^{k} \sum_{j=i+1}^{k} S[i,j] = \frac{S[i,j]}{S[mi,mj]}, \tag{1}$$

where $S_{mi,mj}$ is the minimum distance value in S.

$$T \longleftarrow \sum_{i=1}^{k} \sum_{j=i+1}^{k} T_{i,j} = \sqrt{\sum_{a=0}^{2} \left(E_{i,a} - E_{j,a}\right)^2},$$
$$T \longleftarrow \sum_{i=0}^{k} \sum_{j=i+1}^{k} T[i,j] = \frac{T[i,j]}{T[mi,mj]}, \tag{2}$$

where $T_{mi,mj}$ is the minimum distance value in T.

2.2. Proof 1. Focusing on the optimal M, it can be observed that the distance between the most similar centroids must be located in the closest cells in M, and the smallest values must be obtained for both S and T. Moreover, in this optimal M, the other values in S and T must be obtained proportionally to their smallest values.

2.2.1. Illustration. After clustering and placement operations for $k = 4$, $f = 3$, $r = 6$, and $c = 6$, assume that $C = \{\{0.2, 0.3, 0.1\}, \{0.3, 0.4, 0.2\}, \{0.5, 0.6, 0.4\}, \{0.9, 0.9, 0.8\}\}$ and $E = \{\{0, 0\}, \{0, 3\}, \{2, 2\}, \{5, 5\}\}$, and M is obtained as in the following equation:

$$M = \begin{bmatrix} - & - & - & - & - & C_3 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & C_2 & - & - & - \\ - & - & - & - & - & - \\ C_0 & - & - & C_1 & - & - \end{bmatrix}. \tag{3}$$

For M, S and T matrices are obtained as the following equations:

$$S = \begin{bmatrix} 0 & 0.1 & 0.9 & 1.1 \\ 0 & 0 & 0.3 & 0.9 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$T = \begin{bmatrix} 0 & 1.7 & 2.8 & 11.2 \\ 0 & 0 & 2.2 & 5.3 \\ 0 & 0 & 0 & 4.2 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$S = \begin{bmatrix} 0 & x & 9x & 11x \\ 0 & 0 & 3x & 9x \\ 0 & 0 & 0 & 5x \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$T = \begin{bmatrix} 0 & y & 1.6y & 6.5y \\ 0 & 0 & 1.2y & 3.1y \\ 0 & 0 & 0 & 2.4y \\ 0 & 0 & 0 & 0 \end{bmatrix} \Longrightarrow \tag{4}$$
$$S = \begin{bmatrix} 0 & 1 & 9 & 11 \\ 0 & 0 & 3 & 9 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$T = \begin{bmatrix} 0 & 1 & 1.6 & 6.5 \\ 0 & 0 & 1.2 & 3.1 \\ 0 & 0 & 0 & 2.4 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where x and $y \in R^+$, x is the minimum number in S, and y is the minimum number in T.

2.3. Theorem 2. To calculate the distance between two matrices containing values that are balanced with each other, traditional matrix subtraction can be used. The subtraction operations must be performed with the absolute values, like the approach in Manhattan distance, to obtain a distance value greater than or equal to zero. After the subtraction operations, a Z matrix is obtained, and the sum of all values gives the difference between these matrices as follows:

$$Z \longleftarrow \sum_{i=1}^{k} \sum_{j=i+1}^{k} Z_{i,j} = \left|S_{i,j} - T_{i,j}\right|,$$
$$\ell = \sum_{i=1}^{k} \sum_{j=i+1}^{k} Z_{i,j}. \tag{5}$$

2.4. Proof 2. To compare the two instances containing numeric values using machine learning, normalization operations must be performed to supply numeric balance among the features before the usage of any distance metric like Euclidean distance or Manhattan distance. The first theorem, essentially, claims a normalization operation for S and T matrices. The second theorem claims that, with normalized values, the distance calculation can be performed by using the traditional subtraction operation.

To illustrate, both S and T have the smallest value as 1 and normalized values. The closer the E is to the optimal solution, the smaller the values in the subtraction matrix Z. Thus, ℓ can be obtained as a value close to zero. In this example, the value of ℓ is $7.4 + 5.5 + 1.8 + 5.9 + 2.6 = 23.2$, as in the following equation:

$$\begin{bmatrix} 0 & 1 & 9 & 11 \\ 0 & 0 & 3 & 9 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1.6 & 6.5 \\ 0 & 0 & 1.2 & 3.1 \\ 0 & 0 & 0 & 2.4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Longrightarrow Z = \begin{bmatrix} 0 & 0 & 7.4 & 5.5 \\ 0 & 0 & 1.8 & 5.9 \\ 0 & 0 & 0 & 2.6 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

3. Related Works

In our study, some traditional machine learning algorithms were utilized to implement hybrid approaches: K-means++, SOM++, PCA, and QGA. In this section, these algorithms, hybrid logic, and dimensionality reduction are described, and reasons for preferences in our study are explained along with supporting literature.

The K-means++ algorithm is a successful clustering algorithm, inspired by K-means, that has been used in studies across broadly different domains. For example, Zhang and Hepner have clustered a geographic area in a phenology study [13]; Sharma et al. have clustered satellite images in an astronomy study [14]; Dzikrullah et al. have clustered a passenger dataset in a study of public transportation systems [15]; Nirmala and Veni have used K-means++ to obtain an efficient hybrid method in an optimization study [16]; and lastly, Wang et al. have clustered a microcomputed tomography dataset in a medical study [17].

K-means++ is consistent in that it returns the same pattern at each run. In addition, the K-means++ algorithm submits related clusters for all instances and offers the advantage of starting the cluster analysis with a good initial set of centers [18]. Conversely, it suffers the drawback of poor running performance in determining this initial set of centers, and as to find good initial centroids, it must perform k passes over the data. Therefore, it was necessary to improve K-means++ for use of large datasets, leading to the development of a more efficient parallel version of K-means++ [19]. Another study [20] addressed the problem by using a sorted dataset, which is claimed to decrease the running time. The literature also includes many studies on enhancing the accuracy or the performance of K-means++, but none on visualizing a 2D map for the clusters of this successful algorithm. Therefore, in this study, a novel approach to visualizing K-means++ clusters on a 2D map is detailed.

SOM is both a clustering and a mapping algorithm, used as a visualization tool for exploratory data in different domains owing to its mapping ability [21]. Each cluster in SOM is illustrated as a neuron, and after the training process in an artificial neural network (ANN) structure, each neuron has X and Y values as a position on a map. In addition, all clusters in a SOM map are neighboring [22]. Nanda et al. used SOM for hydrological analysis [23]; Chu et al. used SOM for their climate study [24]; Voutilainen et al. clustered a gerontological medical dataset using SOM [25]; Kanzaki et al. used SOM in their radiation study to analyze the liver damage from radon, X-rays, or alcohol treatments in mice [26]; and Tsai et al. have clustered a dataset about water and fish species in an ecohydrological environment study [27].

Although SOM produces a map where each neuron represents a cluster and all clusters are neighboring [21, 22], the map does not position the returned centroids adjacent to each other as 1-unit distances; in fact, these clusters must be located on farther cells of the map. In this study, a novel approach to visualize SOM mappings in 2D retaining practically original distances among clusters is detailed, using the fast version of SOM, SOM++. In SOM++, the initialization step of K-means++ is used to find the initial centroids of SOM [28].

PCA is another popular machine learning method for feature selection and dimensionality reduction. Owing to its versatility, this algorithm is also used across different domains: Viani et al. used PCA to analyze channel state information (CSI) for wireless detection of passive targets [29]; Tiwari et al. analyzed solar-based organic ranking cycles for optimization using PCA [30]; Hamill et al. used PCA to sort multivariate chemistry datasets [31]; Ishiyama et al. analyzed a cytomegalovirus dataset in a medical study [32]; and Halai et al. used PCA to analyze a neuropsychological model in a psychology study [33]. PCA is used to eliminate some features for multivariate datasets before machine learning analysis, as the dataset may be large and contain some features that would make analysis efficient [34]. These unnecessary features may be identified by PCA for subsequent removal, resulting in a new dataset with new values and fewer features [2]. Essentially, this algorithm is not a clustering algorithm; however, PCA is relevant owing to its dimensionality reduction capacity; this is utilized in our study to obtain a new 2D dataset, which is then clustered using the traditional K-means++ approach.

QGA is a heuristic optimization algorithm. It refers to the smallest unit storing information in a quantum computer as a quantum bit (qubit). A qubit may store a value in between the binary values of "1" or "0," significantly decreasing running time in the determination of an optimized result [35, 36]. This contemporary optimization algorithm is in wide-spread use. Silveira et al. used QGA to implement a novel approach for ordering optimization problems [37]; Chen et al. used QGA for a path planning problem [38]; Guan and Lin implemented a system to obtain a structural optimal design for ships using QGA [39]; Ning et al. used QGA to solve a "job shop scheduling problem" in their study [40]; and Konar et al. implemented a novel QGA as a hybrid quantum-inspired genetic algorithm to solve the problem of scheduling real-time tasks in multiprocessor systems [41]. DRV-P-MC, the focus of interest in our study, is an optimization problem as well, so

this algorithm's efficient run-time performance is employed to determine suitable cluster positions on a 2D map.

Hybrid approaches combine efficient parts of certain algorithms into new wholes, enhancing the accuracy and the efficiency of these algorithms, or producing novel algorithms. In literature, there are many successful hybrid data-mining approaches: Kumar et al. implemented a highly accurate optimization algorithm from the combination of a genetic algorithm with fuzzy logic and ANN [42]; Singhal and Ashraf implemented a high-performance classification algorithm from the combination of a decision tree and a genetic algorithm [43]; Hassan and Verma collected successful high-accuracy hybrid data-mining applications for the medical domain in their study [44]; Thamilselvan and Sathiaseelan reviewed hybrid data-mining algorithms for image classification [45]; Athiyaman et al. implemented a high-accuracy approach combination of association rule mining algorithms and clustering algorithms for meteorological datasets [46]; Sahay et al. proposed a high-performance hybrid data-mining approach combining apriori and K-means algorithms for cloud computing [47]; Yu et al. obtained a novel solution selection strategy using hybrid clustering algorithms [48]; Sitek and Wikarek implemented a hybrid framework for solving optimization problems and constraint satisfaction by using constraint logic programming, constraint programming, and mathematical programming [49]; Abdel-Maksoud et al. proposed a hybrid clustering technique combining K-means and fuzzy C-means algorithms to detect brain tumours with high accuracy and performance [50]; Zhu et al. implemented a novel high-performance hybrid approach containing hierarchical clustering algorithms for the structure of wireless networks [51]; Rahman and Islam combined K-means and a genetic algorithm to obtain a novel high-performance genetic algorithm [52]; and Jagtap proposed a high-accuracy technique to diagnose heart disease by combining Naïve Bayes, Multilayer Perceptron, C4.5 as a decision tree algorithm, and linear regression [53]. What we can infer from a detailed examination of these studies is that K-means and genetic algorithms, and their variants, can be adapted to other algorithms to implement a hybrid approach successfully. Moreover, the combination of K-means and genetic algorithms creates an extremely efficient and highly accurate algorithm.

In data analysis, unnecessary features cause two main problems in performance and accuracy. If a dataset is large or has insignificant features, a downscaling process should be performed by a dimensionality reduction operation to enable efficient use of the analysis algorithms. In literature, many techniques related to dimensionality reduction are presented. For example, Dash et al. claimed that using PCA for dimensionality reduction causes a drawback in understanding the dataset owing to the creation of new features with new values. Furthermore, they posit that the most effective attributes are damaged. Therefore, they presented a novel approach based on an entropy measure for dimensionality reduction [54]. Bingham and Mannila used a random projection (RP) method instead of PCA for the dimensionality reduction of image and text datasets, singular value decomposition (SVD), latent semantic indexing (LSI), and discrete cosine transform, claiming that RP offers simpler calculation than the other methods and has low error rates [55]. Goh and Vidal have used *k*-nearest neighbor and K-means to obtain a novel method for clustering and dimensionality reduction on Riemannian manifolds [56]; Napoleon and Pavalakodi implemented a new technique using PCA and K-means for dimensionality reduction on high-dimensional datasets [57]; Samudrala et al. implemented a parallel framework to reduce the dimensions of large-scale datasets by using PCA [58]; Cunningham and Byron used PCA, factor analysis (FA), Gaussian process factor analysis, latent linear dynamical systems, and latent nonlinear dynamical systems for the dimensional reduction of human neuronal data [59]; and Demarchi et al. reduced the dimensions of the APEX (airborne prism experiment) dataset using the auto-associative neural network approach and the BandClust algorithm [60]. Boutsidis et al. implemented two different dimensional reduction approaches for K-means clustering: the first based on RP and the second based on SVD [61]. Azar and Hassanien proposed a neurofuzzy classifier method based on ANN and fuzzy logic for the dimensional reduction of medical big datasets [62]; Cohen et al. implemented a method using RP and SVD for the dimensional reduction in K-means clustering [63]; Cunningham and Ghahramani discussed PCA, FA, linear regression, Fisher's linear discriminant analysis, linear multidimensional scaling, canonical correlations analysis, slow feature analysis, undercomplete independent component analysis, sufficient dimensionality reduction, distance metric learning, and maximum autocorrelation factors in their survey article and observed that, in particular, PCA was used and evaluated in many studies as a highly accurate analysis [1]; and Zhao et al. used 2D-PCA and 2D locality preserving projection for the 2D dimensionality reduction in their study [64]. Sharifzadeh et al. improved a PCA method as sparse supervised principal component analysis (SSPCA) to adapt PCA for dimensionality reduction of supervised datasets, claiming that the addition of the target attribute made the feature selection and dimensional reduction operations more successful [65].

These studies show PCA to be the most-used method for dimensionality reduction despite reported disadvantages including the creation of new features, which may hamper the understanding of the dataset, changing the values in the most important and efficient features, and complex calculation and low performance for big datasets.

In other sample clustering studies, Yu et al. proposed some distribution-based distance functions, used to measure the similarity between two sets of Gaussian distributions, in their study, and distribution-based cluster structure selection. Additionally, they implemented a framework to determine the unified cluster structure from multiple cluster structures in all data used in their study [66]. In another study by Yu and Wong, a quantization driven clustering approach was designed to obtain classes for many instances. Moreover, they proposed two different methods to improve the performance of their approach, the shrinking process, and the hierarchical structure process [67]. A study by Wang et al. proposed a local gravitation model and implemented two novel measures to discover more information among instances, a local gravitation clustering algorithm for clustering and evaluating the effectiveness

of the model, and communication with local agents to attain satisfactory clustering patterns using only one parameter [68]. Yu et al. designed a framework known as the double affinity propagation driven cluster for clustering on noisy instances and integrated multiple distance functions to avoid the noise involved with using a single distance function [69].

4. The Reorganized Traditional Approaches

This paper assumes two ways of K-means++ clustering, the traditional usage, and a weighted usage. After normalization techniques are used to balance the dataset features and normalize the dataset, traditional K-means++ clustering is performed. This has a preprocess step to discover the initial centroids for the standard K-means. After the initialization of the centroids, K-means clustering runs using the initial centroid values. K-means++ is expressed as Algorithm 1.

In this study, three reorganized traditional mapping approaches were implemented to visualize the centroids of K-means++ returned by Algorithm 1: mapping by SOM++, by PCA, and according to the best two features determined by information gain. These three algorithms are the approaches that were implemented previously in the literature. To compare these approaches with the other five proposed algorithms under the same conditions, the former three algorithms are appropriately reorganized and implemented in this study and formulated in the following manner.

4.1. K-Means++ and Mapping by SOM++ (KS). The best-known mapping algorithm is SOM, and in this study, SOM++ was implemented as the improved version of SOM, providing the centroids generated by K-means++ as the initial weights for SOM to solve DRV-P-MC. In this approach, the traditional K-means++ clustering runs first and then returns the centroids to be mapped. The SOM++ algorithm trains the weights of the neurons on its map by using the Gaussian function as the update function, as in (7). After a specific number of iterations, the weight values reach a trained-value state according to the centroids. To map the centroids, the winning neurons with the nearest distance for all centroids separately are calculated, and the weight values of these winning neurons are converted to the multivariate values of the related centroids, as in (8). Finally, M, containing the pure multivariate values of the centroids as the weight values, is returned.

Let us assume that C is obtained by Algorithm 1, θ is the closest cell in M for each element in C, d is the minimum distance between the current instances, and C_c and θ, and for each neighbor of θ, is computed, as in the following equation:

$$h = \exp\left(\frac{-d^2}{2\sigma^2}\right),$$
$$\omega \longleftarrow \sum_{i=1}^{f} \omega_i = \omega_i + h * \eta * \left(C_{c,i} - \omega_i\right), \tag{7}$$

where α is the neighborhood width parameter, η is the learning rate parameter, h is the neighbourhood function, and ω is the set of weight values in each cell in M.

$$\sum_{i=1}^{k}\sum_{j=1}^{f} \mathrm{WC}(i, j) = C(i, j), \tag{8}$$

where WC is the set of the winner cells for each element in C.

4.2. K-Means++ and Mapping by PCA (KP). The next approach implemented in this study to map the centroids of K-means++ incorporates evaluated PCA. In this approach, once again the traditional K-means++ clustering runs first and then returns the centroids to be mapped. The PCA algorithm is a supervised learning algorithm; therefore, the dataset cannot be used by PCA without a target attribute. In the next step, to enable the use of PCA, another dataset was created with all the features in the original version and an extra attribute to function as the target attribute. This new attribute is filled for each instance according to the obtained centroids, by calculating the distances among the instances and these centroids. Finally, PCA calculates two components by using this new dataset and considering the target attribute, and a 2D structure is attained to map the centroids on M. This approach is formulated in the following expressions.

Let us assume that C is obtained by Algorithm 1, Ω computed by C is the set of the clusters of the instances in I, and I' is a new set of the instances with a new attribute created as the target by filling it with Ω.

Let us assume that, for matrix L, L_x and L_y are the result matrices, as shown in the following equations, respectively:

$$\begin{aligned} L_x &= L * L^T, \\ L_y &= L_T * L. \end{aligned} \tag{9}$$

An alternative way of computing L_y is as in the following equation:

$$\begin{aligned} \mathrm{EV}_x &= L * \mathrm{EV}_y * \mathrm{EL}^{1/2}, \\ L_y &= \mathrm{EV}_y * \mathrm{EL} * \mathrm{EV}_y T, \\ L &\longleftarrow \sum_{i=1}^{n} (I_i' - \mu), \\ L_y &\longleftarrow \sum_{i=1}^{n} L^T * L, \end{aligned} \tag{10}$$

where EL is the same positive eigenvalue of L_x and L_y, EV_x is the eigenvectors of L_x, EV_y is the eigenvectors of L_y, and μ is the mean vector of the instances in I.

I_{PCA} is the 2D dataset computed by PCA as in the following equation:

$$I_{\mathrm{PCA}} = U_2^T * L, \tag{11}$$

where $U_2 = [u_1, u_2]$ is the set of the two first components for the 2D structure.

In the next expressions, the aim is to obtain 2D centroids equivalent to their multivariate structure. Therefore, firstly, the means of two new features of the instances in the same cluster are calculated for each cluster and each feature separately. Let us assume that Ω is valid for the 2D instances in I_{PCA}, and C_{PCA} is calculated as 2D centroids as in the following equation:

```
Input: Number of the centroids, k
Output: Set of the final centroids, C
Begin
v := Randomly selected element in I
V: Set of the initial centroids
Y: Set of the distances between v and the closest element to v in V
y: Length of Y
Add v into V
Add the distance between v and the closest element to v in V, into Y
Repeat
  For i = 1 : y
    U := U + Y_i^2
  u := A random real number between 0 and U
  p := 2
  Repeat
    U' := 0
    For i = 1 : p − 1
      U' := U' + Y_i^2
    p := p + 1
  Until U ≥ u > U'
  Add I_{p−1} into V
  Add the distance between I_{p−1} and the closest element to I_{p−1} in V, into Y
Until V has k centroids
Run the standard K-means with V
Return C returned from the standard K-means
End
```

Algorithm 1: K-means++.

$$C_{\mathrm{PCA}} \longleftarrow \sum_{i=1}^{n}\sum_{j=1}^{2} C_{\mathrm{PCA}}(\Omega i, j) = C_{\mathrm{PCA}}(\Omega i, j) + I_{\mathrm{PCA}}(i, j),$$

$$C_{\mathrm{PCA}} \longleftarrow \sum_{i=1}^{k}\sum_{j=1}^{2} C_{\mathrm{PCA}}(i, j) = \frac{C_{\mathrm{PCA}}(i, j)}{n}. \tag{12}$$

Moreover, the map has the number of its column between 0 and c and the number of its row between 0 and r, and PCA returns two features as decimal numbers. Therefore, secondly, an operation for normalization must be implemented to place these centroids on the map having $c \times r$ dimensions. Thus, these decimal numbers shift their integer equivalents by using the min-max normalization technique. The centroids can then be placed on the $c \times r$ map. Finally M, in which the centroids in E are mapped, is obtained as in the following equation:

$$E \longleftarrow \sum_{i=1}^{k}\sum_{j=1}^{2} E(i, j) = \frac{\left[C_{\mathrm{PCA}}(i, j) - \min_j\right] * c}{\left(\max_j - \min_j\right)}, \tag{13}$$

where NC_1 is the set of the first column values in C_{PCA}, NC_2 is the set of the second column values in C_{PCA}, $\min_{\mathrm{NC1}}$ is the minimum value in NC_1, $\max_{\mathrm{NC1}}$ is the maximum value in NC_1, $\min_{\mathrm{NC2}}$ is the minimum value in NC_2, and $\max_{\mathrm{NC2}}$ is the maximum value in NC_2.

4.3. K-Means++ and Mapping according to the Best Two Features Determined by Information Gain (B2FM). The third approach, mapping the centroids by information gain according to the best two features, is formulated in the following expressions. Firstly, K-means++ runs, and the centroids are obtained. The aim in this approach is to evaluate only the most valuable features for mapping; therefore, the information gain method is used, computed to determine the root feature by decision tree algorithms. Two features having the highest information gain scores are considered as the X axis and the Y axis of the map. The information gain method is a supervised learning algorithm, and it needs a target attribute like PCA. For this reason, in the second step of this approach, a new dataset with the target attribute is created by evaluation of the distances between all centroids and all instances. Thus, the information gain method can use this new dataset and obtain the scores for each feature. At the end of this approach, the values in these features in the centroids shift their integer equivalents between 0 and c and between 0 and r by using the min-max normalization technique. Finally, the centroids are placed on the c x r map.

Let us assume that C is obtained by Algorithm 1, Ω computed by C is the set of the clusters of the instances in I, I' is a new set of the instances using a new attribute as the target (by filling it with Ω), IG is obtained by information gain feature selection method with I', expressed in (23), fc is the highest ranked feature in IG, and FC is the set of the values in the fcth feature in C as in (14); sc is the second highest ranked feature in IG, and SC is the set of the values in the scth feature in C, as shown in the following equations:

$$\mathrm{FC} \longleftarrow \sum_{i=1}^{k} C(i, \mathrm{fc}), \tag{14}$$

$$\mathrm{SC} \longleftarrow \sum_{i=1}^{k} C(i, \mathrm{sc}). \tag{15}$$

Finally, M, where the centroids in E are mapped, is obtained as in the following equation:

$$E \longleftarrow \sum_{i=1}^{k} \sum_{j=1}^{2} E(i, j) = \frac{\left[C_{\mathrm{PCA}}(i, j) - \min_j\right] * c}{\left(\max_j - \min_j\right)}, \tag{16}$$

where $\min_{\mathrm{fc}}$ is the minimum value in FC, $\max_{\mathrm{fc}}$ is the maximum value in FC, $\min_{\mathrm{fc}}$ is the minimum value in FC, and $\max_{\mathrm{fc}}$ is the maximum value in FC.

5. The Proposed Approaches

The five approaches proposed in this study are K-means++ using QGA mapping and weighted K-Means++ using mapping by Pearson's correlation, gain ratio, information gain, and relief feature selection methods. These five approaches are implemented in the study and formulated in the following manner.

5.1. K-Means++ and Mapping by QGA (KQ). The first proposed approach in our study uses QGA to map the centroids of K-means++. Solving DRV-P-MC can be considered as a heuristic approach because k numbers of centroids are tried for placement on a $c - r$ dimensional map, meaning that there are $(c * r) * (c * r - 1) * (c * r - 2) * (c * r - 3) * \cdots * (c * r - k + 1)$ probabilities. Therefore, this problem resembles the well-known "*Travelling Salesmen*" optimization problem. Given the large number of probabilities, the fittest map pattern can be attained by an optimization algorithm. In this study, QGA, the improved version of genetic algorithms, is used. In QGA, the chromosomes represent the 2D map matrix, M, and the genes in the chromosomes represent the cells in M. Moreover, each gene in QGA has a substructure named "qubit" with 2 components α and β; α represents the probability of the existence of the evaluated centroid for the current cell, and β represents the probability of the absence of the evaluated centroid for the current cell. This multicontrol mechanism helps QGA achieve high performance for a large search space.

The first operation is initialization of each qubit (α, β) with $1/\sqrt{2}$ in genes identically. In the second step, the aim is to check the probabilities of placing the centroids into the cells separately and to obtain the map in which all centroids are placed, as a candidate map. These operations are performed by the makeOperator() method using a randomization technique containing three random numbers. Two of them specify the column and the row, and the third determines the cell of the current centroid [36]. The fitness function in this approach calculates the differences between E and C, which is described in detail in Problem Definition. Therefore, the value returned by the fitness function is ideally approximately zero. QGA has a generation number as a parameter, and if this generation number of iterations completes, the best individual with the lowest fitness function value is assumed as the fittest map. If not, QGA needs another assistant method called updateOperator().

Let us assume that rc is the random column number attained by random [0, c), rr is the random row number attained by random [0, r), rn is the random number attained by random [0, 1), and M' returned by makeOperator() is computed as in the following equation:

$$M' \longleftarrow \sum_{i=1}^{r} \sum_{j=}^{c} M'_{i,j} = 0,$$

$$M' \longleftarrow \sum_{i=1}^{k} M'_{\mathrm{rc,\ rr}} = \begin{cases} i, \ \mathrm{rn} > \left|\alpha_{\mathrm{rc,rr}}\right|^2, \\ i, \ \mathrm{it} = 0, \\ \text{obtain new rc and rr numbers, decrease it,} \\ \text{and for the current } i\text{, continue to control, otherwise.} \end{cases} \tag{17}$$

The updateOperator() method updates all qubits in the individuals in the population according to the best individual, to approximate its angularity using a trigonometric approach. Firstly, this method needs a subfunction called the sign function, sf, to obtain a score between 0 and 1 according to the distance between the fitness function values of the best individual and the individual being evaluated. After that, this sf value is evaluated in a lookup table containing a condition list to determine the angle [36].

Let us assume that ff is the value of the fitness function; BS is the best individual in the previous population according to ff(BS); CS is the current individual in the current population according to ff(CS); bv is a gene value obtained by makeOperator() for BS; cv is a gene value obtained by makeOperator() for CS; α' and β' are the previous values of α and β for CS; sf is the sign function, and its value is extracted according to the conditions list as in (18); $\Delta\partial$ is the orientation of rotation angle to update the qubit values as in (19); and α and β are updated for each gene in each chromosome in QM as in (20). Finally, QM with new qubit values is returned by updateOperator():

$$\mathrm{sf}(\alpha', \beta') = \begin{cases} +1, & \alpha' * \beta' * [\mathrm{ff}(\mathrm{BS}) - \mathrm{ff}(\mathrm{CS})] < 0, \\ -1, & \alpha' * \beta' * [\mathrm{ff}(\mathrm{BS}) - \mathrm{ff}(\mathrm{CS})] > 0, \\ \pm 1, & \alpha' = 0 \ \text{ and } \ (bv - cv) * [\mathrm{ff}(\mathrm{BS}) - \mathrm{ff}(\mathrm{CS})] < 0, \\ \pm 1, & \beta' = 0 \ \text{ and } \ (bv - cv) * [\mathrm{ff}(\mathrm{BS}) - \mathrm{ff}(\mathrm{CS})] > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

$$\Delta\partial = \begin{cases} 0.025\pi, & \mathrm{ff}(\mathrm{BS}) \geq \mathrm{ff}(\mathrm{CS}) \ \text{ and } \ \mathrm{cv} = 1 \ \text{ and } \ \alpha' = 0 \ \text{ and } \ \beta' = -1 \ \text{ and } \ \mathrm{sf}(\alpha', \beta') = \pm 1, \\ 0.005\pi, & \mathrm{ff}(\mathrm{BS}) < \mathrm{ff}(\mathrm{CS}) \ \text{ and } \ \mathrm{cv} = 1 \ \text{ and } \ \mathrm{bv} = 1 \ \text{ and } \ \alpha' = 0 \ \text{ and } \ \beta' = -1 \ \text{ and } \ \mathrm{sf}(\alpha', \beta') = +1, \\ 0.01\pi, & \mathrm{ff}(\mathrm{BS}) < \mathrm{ff}(\mathrm{CS}) \ \text{ and } \ \mathrm{cv} = 1 \ \text{ and } \ \mathrm{bv} = 0 \ \text{ and } \ \alpha' = \pm 1 \ \text{ and } \ \beta' = +1 \ \text{ and } \ \mathrm{sf}(\alpha', \beta') = +1, \\ 0.05\pi, & \mathrm{ff}(\mathrm{BS}) \geq \mathrm{ff}(\mathrm{CS}) \ \text{ and } \ \mathrm{cv} = 0 \ \text{ and } \ \mathrm{bv} = 1 \ \text{ and } \ \alpha' = \pm 1 \ \text{ and } \ \beta' = +1 \ \text{ and } \ \mathrm{sf}(\alpha', \beta') = -1, \\ 0, & \text{otherwise}, \end{cases} \tag{19}$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos(\Delta\partial) & -\sin(\Delta\partial) \\ \sin(\Delta\partial) & \cos(\Delta\partial) \end{bmatrix} * \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix}. \tag{20}$$

The calling of the makeOperator() and updateOperator() methods continues until the generation number is reached. After completing the iterations, the fittest map is returned by the algorithm as given in Algorithm 2.

5.2. Weighted K-Means++ and Mapping by Feature Selection Methods. Literature indicates that when the supervised analysis of a dataset involves many features, insignificant features can negatively affect the results of the analysis. Therefore, feature selection methods are used to eliminate some features according to their degrees of influence over the class attribute. In the next proposed approaches, all features can be considered as being determined by the X and Y axes without any elimination and with losing the most valuable feature, unlike in PCA, even if 2D mapping needs only two dimensions. Indeed, the aims of these approaches are to keep the most valuable feature and to assume it to be the X axis and in addition, to evaluate other all features in a weighted manner as the Y axis. As a result, these approaches use four feature selection methods to determine the efficiencies of the features separately: Pearson's correlation, gain ratio, information gain, and relief. In these proposed approaches, the centroids obtained by the traditional K-means++ clustering are not evaluated in their pure form but are converted to their weighted values according to the scores returned by the feature selection methods.

In Algorithm 3, the weighted K-means++ clustering is expressed. Firstly, the values of the features in all instances are multiplied by the weight values, and a new set of the instances is attained. Finally, these transformed instances are used by the traditional K-means++ clustering, and the more qualified centroids are returned by this method. The efficiency and the success of the weighted clustering approach are detailed in [70].

5.2.1. Weighted K-Means++ and Mapping by Pearson's Correlation (WMC). In all approaches in this study, unsupervised analyses are the focus; therefore, to use feature selection methods, which are supervised learning algorithms, firstly, the traditional K-means++ clustering is implemented, and a new dataset with the clusters returned is created. The correlation coefficient values are formulated in the following expressions, for all features can be calculated with this dataset; thus, these values are evaluated by weighted K-means++ clustering, and the centroids that are returned can be placed on the 2D map.

Let us assume that CC is the set of the correlation coefficient values between all features and the target attribute as in the following equation:

$$\mathrm{CC} \longleftarrow \sum_{i=1}^{f} \frac{n * \sum_{j=1}^{n} \left(I_{ij} * t_j\right) - \sum_{j=1}^{n} I_{ij} * \sum_{j=1}^{n} t_j}{\sqrt{\left[n * \sum_{j=1}^{n} I_{ij}^2 - \left(\sum_{j=1}^{n} I_{ij}\right)^2\right] * \left[n * \sum_{j=1}^{n} t_j^2 - \left(\sum_{j=1}^{n} t_j\right)^2\right]}},$$

$$\mathrm{CC} \longleftarrow \sum_{i=1}^{f} \mathrm{CC}_i = \begin{cases} \dfrac{\mathrm{CC}_i}{\min_{\mathrm{cc}}}, & \min_{\mathrm{cc}} > 0 > 0, \\ \dfrac{\max_{\mathrm{cc}}}{\mathrm{CC}_i} & \max_{\mathrm{cc}} < 0, \\ \dfrac{\mathrm{CC}_i - \min_{\mathrm{cc}}}{\max_{\mathrm{cc}} - \min_{\mathrm{cc}}} * 0.9 + 0.1, & \text{else}, \end{cases} \tag{21}$$

```
Input: Number of the centroids, k, iteration number (it)
Output: Map with C placed, M
Begin
QM: Population containing individual qubit matrices
PM: Population containing individual probability matrices
qm: Length of QM
pm: Length of PM
tr := 0
C: Set of the centroids returned by Algorithm 1
Repeat
For a = 1 : qm
   For i = 1 : c
      For j = 1 : r
         QM · α_{a,i,j} := 1/√2
         QM · β_{a,i,j} := 1/√2
For a = 1 : pm
   makeOperator(it, PMa)
QM = updateOperator(QM)
tr := tr + 1
Until tr = it
Return M having the fittest value
End
```

Algorithm 2: K-means++ and mapping by QGA.

```
Input: Set of the weight values for all features (W)
Output: Set of the final centroids, C
Begin
For i = 1 : n
   For j = 1 : f
      I_{i,j} = I_{i,j} * W_j
Call Algorithm 1 for I to cluster with K-means++
Return C returned by Algorithm 1
End
```

Algorithm 3: The weighted K-means++ clustering.

where $\min_{cc}$ is the minimum value in CC and $\max_{cc}$ is the maximum value in CC.

Algorithm 4, formulated to preserve the most valuable feature determined by Pearson's correlation, gives a detailed presentation of the weighted mapping of the centroids on the 2D map. In this method, the values in the most valuable feature and the weighted averaged values in the other features in the centroids shift their integer equivalents between 0 and c, and between 0 and r, by using the min-max normalization technique. Finally, M is returned.

5.2.2. Weighted K-Means++ and Mapping by Gain Ratio (WMG). In Algorithm 5, the weighted K-means++ using gain ratio feature selection and weighted mapping by preserving the most valuable feature are formulated. The same operations as in Algorithm 4 are implemented in this proposed approach, except the calculation of the weight values. The weight values are obtained by the gain ratio feature selection method in this approach. In addition, owing to the different weight values from the correlation coefficient values, centroids with different multivariate feature values are computed in this mapping method.

Let us assume that B_i represents the ith category in t, $P(B_i)$ is the probability of the ith category, G_j represents the jth category in a feature, $P(G_i)$ is the probability of the jth category, $P(B_i|G_j)$ is the conditional probability of the ith category given that term G_j appeared, and GR is the set of the gain ratio values between all features and the target attribute, as shown in the following equation:

Input: Number of the centroids, k
Output: Map with C placed, M
Begin
C': Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 1
Ω: Set of the clusters of the instances in I, computed by C'
I': Set of the instances, with a new attribute as the target by filling it with Ω,
CC: Set of the weight values obtained by Pearson's correlation feature selection method
C: Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 3
fc: The highest ranked feature in CC
FC: Set of the values in the fcth feature in C
wc: Sum of the scores in the features except the fcth feature
WC: Set of the average of the values in the other features
For $i = 1 : k$
  $\mathrm{FC}_i = C_{i,\mathrm{fc}}$
For $i = 1 : k$
  For $j = 1 : f$
    If j is not equal to fc
      $\mathrm{WC}_i = \mathrm{WC}_i + C_{i,j}$
For $i = 1 : f$
  If j is not equal to fc
    $\mathrm{wc} = \mathrm{wc} + \mathrm{CC}_i$
For $i = 1 : k$
  $\mathrm{WC}_i = \mathrm{WC}_i/\mathrm{wc}$
$\min_{\mathrm{fc}}$: The minimum value is in FC
$\max_{\mathrm{fc}}$: The maximum value is in FC
$\min_{\mathrm{wc}}$: The minimum value is in WC
$\max_{\mathrm{wc}}$: The maximum value is in WC
For $i = 1 : k$
  For $j = 1 : f$
    $E_{i,j} = [C_{i,j} - \min_j] * c/(\max_j - \min_j)$
Return M where the centroids in E are mapped
End

Algorithm 4: The weighted K-means++ and mapping by Pearson's correlation.

$$\mathrm{GR} \longleftarrow \frac{-\sum_{i=1}^{b} P(B_i) \log P(B_i) + \sum_{i=1}^{b} \sum_{j=1}^{g} P(G_i) \sum_{i=1}^{b} P\left(B_i \middle| G_j\right) \log P\left(B_i \middle| G_j\right)}{-\sum_{j=1}^{g} P(G_i) \log P(G_i)},$$

$$\mathrm{GR} \longleftarrow \sum_{i=1}^{f} \mathrm{GR}_i = \begin{cases} \dfrac{\mathrm{GR}_i}{\min_{\mathrm{gr}}}, & \min_{\mathrm{gr}} > 0, \\ \dfrac{\max_{\mathrm{gr}}}{\mathrm{GR}_i}, & \max_{\mathrm{gr}} < 0, \\ \dfrac{\mathrm{GR}_i - \min_{\mathrm{gr}}}{\max_{\mathrm{gr}} - \min_{\mathrm{gr}}} * 0.9 + 0.1, & \text{otherwise}, \end{cases} \tag{22}$$

where $\min_{\mathrm{gr}}$ is the minimum value in GR and $\max_{\mathrm{gr}}$ is the maximum value in GR.

5.2.3. Weighted K-Means++ and Mapping by Information Gain (WMI). In Algorithm 5, the weighted K-means++ using information gain feature selection and weighted mapping by preserving the most valuable feature is formulated. The same operations are implemented as in the previous approach, except the calculation of the weight values. The weight values are obtained by the information gain feature selection method in this approach. Furthermore, because of the different weight values from gain ratio values, centroids with different multivariate feature values are computed in this mapping method (Algorithm 6).

Let us assume that B_i represents the ith category in t, and $P(Bi)$ is the probability of the ith category; G_j represents the jth category in a feature, and (G_j) is the probability of the jth category; $P(B_i|G_j)$ is the conditional probability of the ith category given that term G_j appeared, and IG is the set of the information gain values between all features and the target attribute:

Input: Number of the centroids, k
Output: Map with C placed, M
Begin
C': Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 1
Ω: Set of the clusters of the instances in I, computed by C'
I': Set of the instances, with a new attribute as the target by filling it with Ω
GC: Set of the weight values obtained by gain ratio feature selection method
C: Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 3
fc: The highest ranked feature in GR
FC: Set of the values in the fcth feature in C
wc: Sum of the scores in the features except the fcth feature
WC: Set of the average of the values in the other features
For $i = 1 : k$
  $\mathrm{FC}_i = C_{i,\mathrm{fc}}$
For $i = 1 : k$
  For $j = 1 : f$
    If j is not equal to fc
      $\mathrm{WC}_i = \mathrm{WC}_i + C_{i,\ j}$
For $i = 1 : f$
  If j is not equal to fc
    $\mathrm{wc} = \mathrm{wc} + \mathrm{GR}_i$
For $i = 1 : k$
  $\mathrm{WC}_i = \mathrm{WC}_i/\mathrm{wc}$
$\min_{\mathrm{fc}}$: The minimum value is in FC
$\max_{\mathrm{fc}}$: The maximum value is in FC
$\min_{\mathrm{wc}}$: The minimum value is in WC
$\max_{\mathrm{wc}}$: The maximum value is in WC
For $i = 1 : k$
  For $j = 1 : f$
    $E_{i,j} = [C_{i,j} - \min_j] * c/(\max_j - \min_j)$
Return M where the centroids in E are mapped
End

Algorithm 5: The weighted K-means++ and mapping by gain ratio.

Input: Number of the centroids, k
Output: Map with C placed, M
Begin
C': Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 1
Ω: Set of the clusters of the instances in I, computed by C'
I': Set of the instances, with a new attribute as the target by filling it with Ω
IG: Set of the weight values obtained by information gain feature selection method
C: Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 3
fc: The highest ranked feature in IG
FC: Set of the values in the fcth feature in C
wc: Sum of the scores in the features except the fcth feature
WC: Set of the average of the values in the other features
For $i = 1 : k$
  $\mathrm{FC}_i = \mathrm{C}_{i,\mathrm{fc}}$
For $i = 1 : k$
  For $j = 1 : \mathrm{f}$
    If j is not equal to fc
      $\mathrm{WC}i = \mathrm{WC}i + \mathrm{C}_{i,j}$
For $i = 1 : f$
  If j is not equal to fc
    $\mathrm{wc} = \mathrm{wc} + \mathrm{IG}_i$
For $i = 1 : k$
  $\mathrm{WC}_i = \mathrm{WC}_i/\mathrm{wc}$
$\min_{\mathrm{fc}}$: The minimum value is in FC

Algorithm 6: Continued.

$\max_{fc}$: The maximum value is in FC
$\min_{wc}$: The minimum value is in WC
$\max_{wc}$: The maximum value is in WC
For $i = 1 : k$
  For $j = 1 : f$
    $E_{i,j} = [C_{i,j} - \min_j] * c/(\max_j - \min_j)$
Return M where the centroids in E are mapped
End

Algorithm 6: The weighted K-means++ and mapping by information gain.

$$\begin{aligned}
\mathrm{IG} \longleftarrow & -\sum_{i=1}^{b} P(B_i) \log P(B_i) \\
& + \sum_{i=1}^{b} \sum_{j=1}^{g} P(G_i) \sum_{i=1}^{b} P\left(B_i \middle| G_j\right) \log P\left(B_i \middle| G_j\right), \\
\mathrm{IG} \longleftarrow \sum_{i=1}^{f} \mathrm{IG}_i = & \begin{cases} \dfrac{\mathrm{IG}_i}{\min_{\mathrm{IG}}}, & \min_{\mathrm{IG}} > 0, \\ \dfrac{\max_{\mathrm{IG}}}{\mathrm{IG}_i}, & \max_{\mathrm{IG}} < 0, \\ \dfrac{\mathrm{IG}_i - \min_{\mathrm{IG}}}{\max_{\mathrm{IG}} - \min_{\mathrm{IG}}} * 0.9 + 0.1, & \text{else}, \end{cases}
\end{aligned} \tag{23}$$

where $\min_{IG}$ is the minimum value in IG and $\max_{IG}$ is the maximum value in IG.

5.2.4. Weighted K-Means++ and Mapping by Relief (WMR). In Algorithm 7, the weighted K-means++ using relief feature selection and weighted mapping by preserving the most valuable feature is formulated. The same operations as in the previous approach are implemented, except the calculation of the weight values. The weight values are obtained by the relief feature selection method in this approach. Moreover, because of the different weight values from information gain values centroids with different multivariate feature values are computed in this mapping method. Let us assume that R is a random instance from D, I_h is the closest instance to R in the same class where R exists, I_m is the closest instance to R in another class where R does not exist, and RF is the set of the relief feature values between all features and the target attribute in the following equation:

$$\begin{aligned}
\mathrm{RF} \longleftarrow & \sum_{i=1}^{f} \mathrm{RF}_i = 0, \\
\mathrm{RF} \longleftarrow & \sum_{i=1}^{n} \sum_{j=1}^{f} \mathrm{RF}_j = \mathrm{RF}_j - \left|I_{hj} - R_i\right| + \left|I_{mj} - R_i\right|, \\
\mathrm{RF} \longleftarrow \sum_{i=1}^{f} \mathrm{RF}_i = & \begin{cases} \dfrac{\mathrm{RF}}{\min_{\mathrm{rf}}}, & \min_{\mathrm{rf}} > 0, \\ \dfrac{\max_{\mathrm{rf}}}{\mathrm{RF}_i}, & \max_{\mathrm{rf}} < 0, \\ \dfrac{\mathrm{RF}_i - \min_{\mathrm{rf}}}{\max_{\mathrm{rf}} - \min_{\mathrm{rf}}} * 0.9 + 0.1, & \text{else}, \end{cases}
\end{aligned} \tag{24}$$

where $\min_{rf}$ is the minimum value in RF and $\max_{rf}$ is the maximum value in RF.

6. Experimental Results and Discussions

Accuracy, consistency, and compatibility tests are implemented in the study. The accuracy tests involve measurements of precision, recall, *f*-measure, and accuracy. Consistency tests involve calculation of the sum of squared error (SSE). Compatibility tests consider the difference between multivariate and 2D structures (DBM2). This section details the implementation of the tests on the proposed and traditional algorithms on 14 datasets of various sizes and comparison of the test results. For the measurement of the relevant metrics for these processes, a versatile tool, whose interface is shown in Figure 1, was developed as a desktop application on Visual Studio 2017 using the WEKA data-mining software package for background operations [71]. To implement this tool, programming aspects included the development of pseudocode (fragments were given in Section 4 and Section 5), with consideration of mainly object-oriented programming concepts, leading to the placement of each algorithm as a class structure within the application. Finally, C# was selected as a programming language, in which sorted lists were implemented as data structures.

The test tool was developed with the capability to select the desired file, algorithm, size of the 2D map, normalization type, and distance metric. Aside from the algorithms, the options are 2D map sizes from 2×2 to 20×20, Euclidean (25), Manhattan (26), or Chebyshev (27) distance metrics, and four normalization types, specifically min-max (28), *z*-score (29), decimal scaling (30), and division by maximum value methods (31):

$$d_{i,j} = \sqrt{\sum_{a=1}^{f} \left(I_{i,a} - I_{j,a}\right)^2}, \tag{25}$$

$$d_{i,j} = \sum_{a=1}^{f} \left|I_{i,a} - I_{j,a}\right|, \tag{26}$$

$$d_{i,j} = \max\left(\sum_{a=1}^{f} \left|I_{i,a} - I_{j,a}\right|\right), \tag{27}$$

where $d_{i,j}$ is the distance between the ith and jth instances.

Input: Number of the centroids, k
Output: Map with C placed, M
Begin
C' : Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 1
Ω : Set of the clusters of the instances in I, computed by C'
I' : Set of the instances, with a new attribute as the target by filling it with Ω
RF : Set of the weight values obtained by information gain feature selection method
C : Set of the centroids obtained by the traditional K-means++ clustering in Algorithm 3
fc : The highest ranked feature in RF
FC : Set of the values in the fcth feature in C
wc : Sum of the scores in the features except the fcth feature
WC : Set of the average of the values in the other features
For $i = 1 : k$
  $FC_i = C_{i,fc}$
For $i = 1 : k$
  For $j = 1 : f$
    If j is not equal to fc
      $WC_i = WC_i + C_{i,j}$
For $i = 1 : f$
  If j is not equal to fc
    $wc = wc + RF_i$
For $i = 1 : k$
  $WC_i = WC_i/wc$
min_{fc}: The minimum value is in FC
max_{fc}: The maximum value is in FC
min_{wc}: The minimum value is in WC
max_{wc}: The maximum value is in WC
For $i = 1 : k$
  For $j = 1 : f$
    $E_{i,j} = [C_{i,j} - min_j] * c/(max_j - min_j)$
Return M where the centroids in E are mapped
End

Algorithm 7: The weighted K-means++ and mapping by relief.

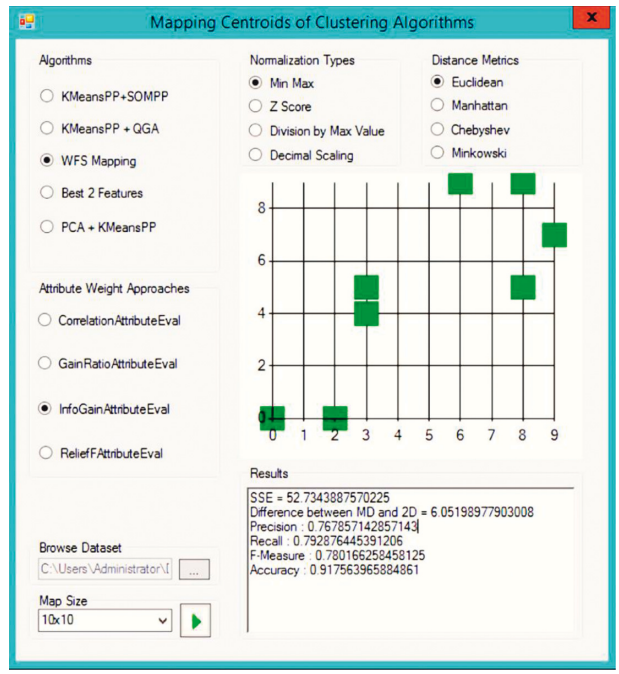

Figure 1: An interface of the tool after clustering.

$$\mathrm{NV} = \frac{\mathrm{FV} - \mathrm{FV}_{\min}}{\mathrm{FV}_{\max} - \mathrm{FV}_{\min}} * \left(\mathrm{NV}_{\max} - \mathrm{NV}_{\min}\right) + \mathrm{NV}_{\min}, \quad (28)$$

$$\mathrm{NV} = \frac{\mathrm{FV} - \mu}{\sigma}, \quad (29)$$

$$\mathrm{NV} = \frac{\mathrm{FV}}{10^{\tau}}, \quad (30)$$

$$\mathrm{NV} = \frac{\mathrm{FV}}{\mathrm{NV}_{\max}}, \quad (31)$$

where FV is the current value, NV is the normalized value of FV, $\mathrm{FV}_{\min}$ is the minimum value in the current feature (F), $\mathrm{FV}_{\max}$ is the maximum value in F, $\mathrm{NV}_{\max}$ is the new maximum value in F, $\mathrm{NV}_{\min}$ is the new minimum value in F, μ is the mean of the values in F, σ is the standard deviation value in F, and τ is lowest integer value such that max $(|\mathrm{NV}|) < 1$.

6.1. Dataset Description. The datasets used in our study are listed in Table 1 together with their characteristics. They are open-source datasets, from the UCI machine learning repository, and are the same datasets used in previous studies in the literature containing supervised analyses because of their class attributes.

6.2. Validation Metrics. To verify the accuracy of the algorithms, blind versions of these datasets were created in the experimental studies, and the results of the precision (P_r), recall (R_e), f-measure (F_m), and accuracy (A_c) tests were generated by means of matches between the desired and proposed classes. The P_r, R_e, F_m, and A_c test results in this section were calculated using min-max normalization between 0 and 1 and using the Euclidean distance metric owing to this being generally used for K-means++ clustering in literature. In addition, the clustering operations were performed according to the original class numbers of the datasets, in order to be able to compare the desired classes with the proposed clusters.

In Table 2, the A_c, F_m, R_e, and P_r values, which were computed by the standard K-means++ clustering, and the weighted K-means++ clustering with Pearson's correlation, gain ratio, information gain, and relief features selection methods are given. They are presented separately as percentages for 14 datasets in Table 2. The A_c, F_m, R_e, and P_r values can be computed as follows, respectively:

$$A_c = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}, \quad (32)$$

$$F_m = 2 * \frac{P_r * R_e}{P_r + R_e}, \quad (33)$$

$$R_e = \frac{T_p}{T_p + T_n}, \quad (34)$$

$$P_r = \frac{T_p}{T_p + F_p}, \quad (35)$$

where T_p is the true-positive value, T_n is the true-negative value, F_p is the false-positive value, and F_n is the false-negative value.

To verify the consistency of the centroids, the measurements for SSE analyses were implemented in this experimental study. The SSE values for 14 datasets using Euclidean, Manhattan, and Chebyshev distance metrics to see the variances for each metric were computed separately. SSE is computed as in the following equation:

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{j=1}^{f} \left(C_{i,j} - I''_{i,j}\right)^2, \quad (36)$$

where I'_i is the set of the closest instances in I to C_i.

Finally, for compatibility testing, the differences between multivariate and 2D structures of the centroids as ℓ were measured as detailed in Problem Definition. The ℓ value is expected to be zero or approximately zero. Resultant ℓ values were evaluated separately in terms of the traditional (KS, KP, and B2FM) and the proposed (KQ, WMC, WMG, WMI, and WMR) approaches. The results shown in Figure 2 were obtained on 10 datasets having more than 2 classes because it was observed that all approaches could map the 2 centroids of the other 4 datasets (Heart Statlog, Johns Hopkins University Ionosphere Database, Pima Indians Diabetes Database, and Wisconsin breast cancer) on the furthest edges on the map, and consequently, the ℓ values were computed as 0, that is, the most optimal value that can be obtained.

6.3. Experimental Results. The results of accuracy tests show generally that information gain is the most successful clustering approach as it has higher A_c values for 12 datasets than is standard in comparison with the other approaches. Gain ratio has higher accuracy values than standard for 9 datasets, correlation has higher accuracy for 7 datasets, and relief has higher accuracy for 5 datasets. Considering F_m values, information gain has higher values than standard for 10 datasets, gain ratio has higher values for 8 datasets, correlation has higher values for 6 datasets, and relief has higher values for 4 datasets. With respect to R_e, information gain has higher values then standard for 10 datasets, gain ratio has higher values for 8 datasets, correlation has higher values for 6 datasets, and relief has higher values 5 for datasets. Finally, regarding P_r values, info gain has higher percentage values for 9 datasets than standard; gain ratio has higher values for 8 datasets, correlation has higher values for 6 datasets, and relief has higher values for 3 datasets. These results support the conclusion that weighted K-means++ clustering with feature selection methods, particularly information gain, provides more accurate centroids than traditional K-means++ clustering.

The values in Table 3 were obtained by calculating the means of the A_c, F_m, R_e, and P_r values in Table 2 for 14 datasets. The results in this table demonstrate that the weighted clustering approaches help to increase the accuracy of the clustering algorithms. In addition, information gain is shown to be the most efficient feature selection method for clustering operations, as it has the highest values for the

Table 1: The datasets with their types, existence of missing values, and sizes.

Datasets	Type	Instances	Features	Classes	Missing values
Blocks Classification	Numeric	5473	10	5	No
Cardiac Arrhythmia Database	Numeric	452	278	16	Yes
Column 3C Weka	Numeric	310	6	3	No
Glass Identification Database	Numeric	214	9	7	No
Heart Statlog	Numeric	270	13	2	No
Iris Plants Database	Numeric	150	4	3	No
Johns Hopkins University Ionosphere Database	Numeric	351	34	2	No
Optical Recognition of Handwritten Digits	Numeric	5620	64	10	No
Pima Indians Diabetes Database	Numeric	768	8	2	No
Protein Localization Sites	Numeric	336	7	8	No
Vowel Context Data	Numeric	990	11	11	No
Waveform Database Generator	Numeric	5000	40	3	No
Wine Recognition Data	Numeric	178	13	3	No
Wisconsin Breast Cancer	Numeric	699	9	2	Yes

Table 2: The A_c, F_m, R_e, and P_r values (%) calculated separately for 14 datasets and the standard and the weighted K-means++ clustering approaches.

Datasets		Standard	Correlation	Gain ratio	Info gain	Relief
Blocks Classification	A_c	94.81	95.13*	94.93*	95.12*	95.13*
	F_m	95.37	95.59*	95.45*	95.58*	95.59*
	R_e	96.86	96.95*	96.89*	96.95*	96.95*
	P_r	92.69	92.03	94.44*	94.81*	94.45
Cardiac Arrhythmia Database	A_c	97.64	98.08*	96.91	97.66*	95.85
	F_m	94.35	95.97*	93.22	93.77	90.67
	R_e	96.07	96.60*	94.69	95.44	93.28
	P_r	92.69	95.35*	91.79	92.15	88.21
Column 3C Weka	A_c	85.77	84.73	84.29	86.81*	83.81
	F_m	79.79	81.03	80.46	83.58*	79.86
	R_e	80.24	81.42	80.93	83.95*	80.37
	P_r	79.35	80.64	80.00	83.22*	79.35
Glass Identification Database	A_c	81.05	75.39	77.62	81.55*	75.57
	F_m	56.85	51.55	50.49	58.48*	48.10
	R_e	79.85	56.54	66.47	79.99*	55.27
	P_r	44.13	47.38*	40.70	46.09*	42.57
Heart Statlog	A_c	69.20	73.61*	73.61*	73.61*	73.61*
	F_m	74.16	77.19*	77.19*	77.19*	77.19*
	R_e	75.81	78.11*	78.11*	78.11*	78.11*
	P_r	72.59	72.59	72.22	86.29*	80.74*
Iris Plants Database	A_c	91.97	95.98*	96.87*	96.87*	97.32*
	F_m	88.01	94.00*	95.33*	95.33*	97.32*
	R_e	88.03	94.01*	95.33*	95.33*	96.04*
	P_r	88.00	94.00*	95.33*	95.33*	96.00*
Johns Hopkins University Ionosphere Database	A_c	67.22	67.17	67.56*	68.21*	67.17
	F_m	73.60	73.55	73.85*	74.30*	73.55
	R_e	78.58	78.46	78.77*	79.05*	78.46
	P_r	69.23	69.23	69.51*	70.08*	69.23
Optical Recognition of Handwritten Digits	A_c	84.51	84.52*	86.17*	85.47*	84.50
	F_m	19.68	19.54	31.97*	26.03*	19.49
	R_e	25.15	24.56	41.63*	33.02*	24.59
	P_r	16.17	16.23*	25.95*	21.48*	16.14
Pima Indians Diabetes Database	A_c	77.77	77.77	77.51	78.01*	77.10
	F_m	80.98	80.98	80.81	81.10*	80.53
	R_e	83.31	83.31	83.26	83.27	83.10
	P_r	78.77	78.77	78.51	79.03*	78.12
Protein Localization Sites	A_c	92.38	92.05	95.30*	91.75	91.85
	F_m	79.44	78.44	88.61*	78.01	77.37
	R_e	81.63	80.66	90.06*	79.28	79.86
	P_r	77.38	76.33	87.20*	76.78	75.03

Table 2: Continued.

Datasets		Standard	Correlation	Gain ratio	Info gain	Relief
Vowel Context Data	A_c	88.48	88.81*	88.68*	88.77*	87.75
	F_m	42.26	43.49*	41.30	41.96	35.87
	R_e	53.06	54.95*	49.54	53.97*	43.59
	P_r	35.11	35.99*	35.41*	34.33	30.48
Waveform Database Generator	A_c	83.26	83.24	83.38*	83.17	83.09
	F_m	79.41	79.39	79.54*	79.32	79.23
	R_e	82.67	82.66	82.80*	82.65	82.54
	P_r	76.40	76.38	76.54*	76.26	76.18
Wine Recognition Data	A_c	96.39	94.77	95.94	97.48*	97.51*
	F_m	95.14	92.85	94.45	96.36*	96.40
	R_e	95.35	93.01	94.53	96.65*	96.75*
	P_r	94.94	92.69	94.38	96.06	96.06
Wisconsin Breast Cancer	A_c	71.69	72.43*	72.58*	72.58*	72.58*
	F_m	77.19	77.69*	77.79*	77.79*	77.79*
	R_e	82.48	82.71*	82.76*	82.76*	82.76*
	P_r	72.53	73.24*	73.39*	73.39*	73.39*

*The A_c, F_m, R_e, and P_r values (%) higher than the standard approach.

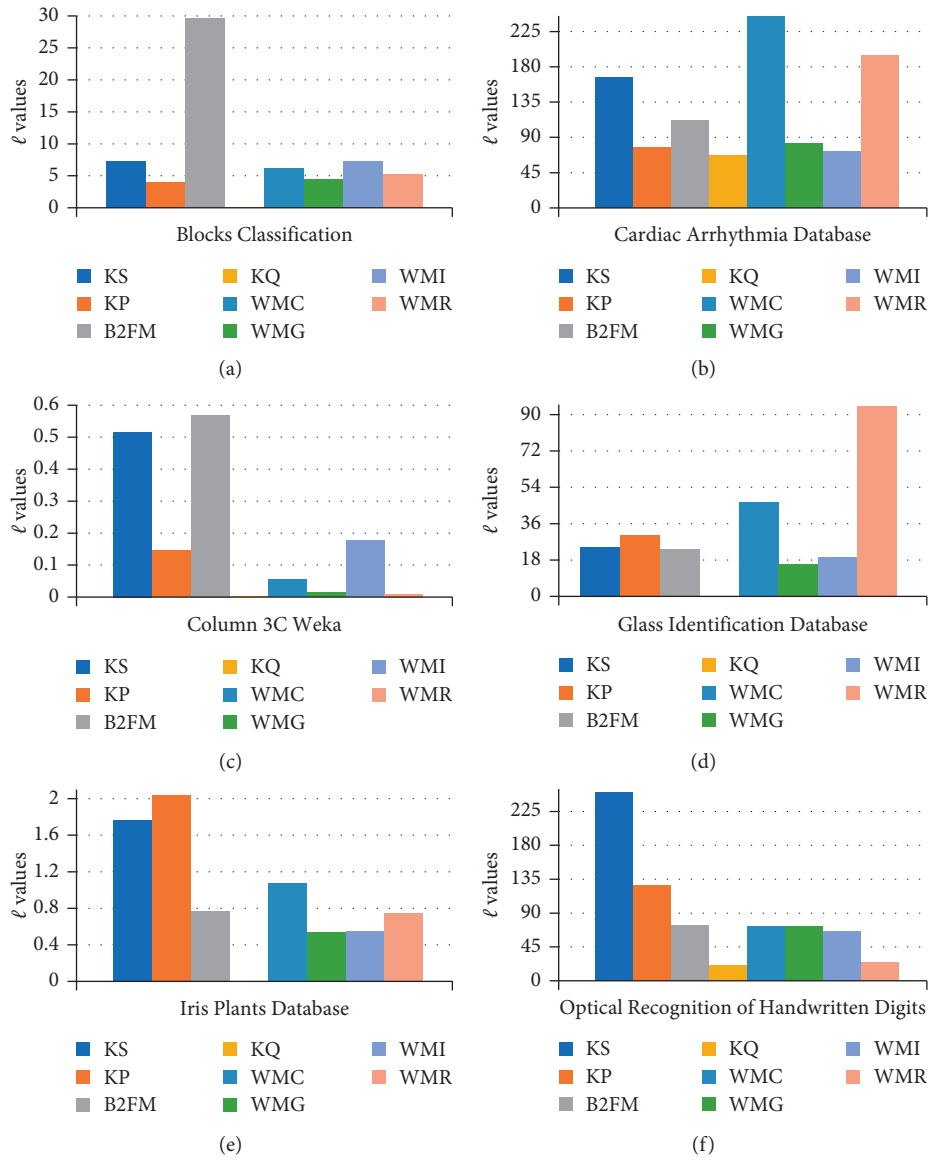

Figure 2: Continued.

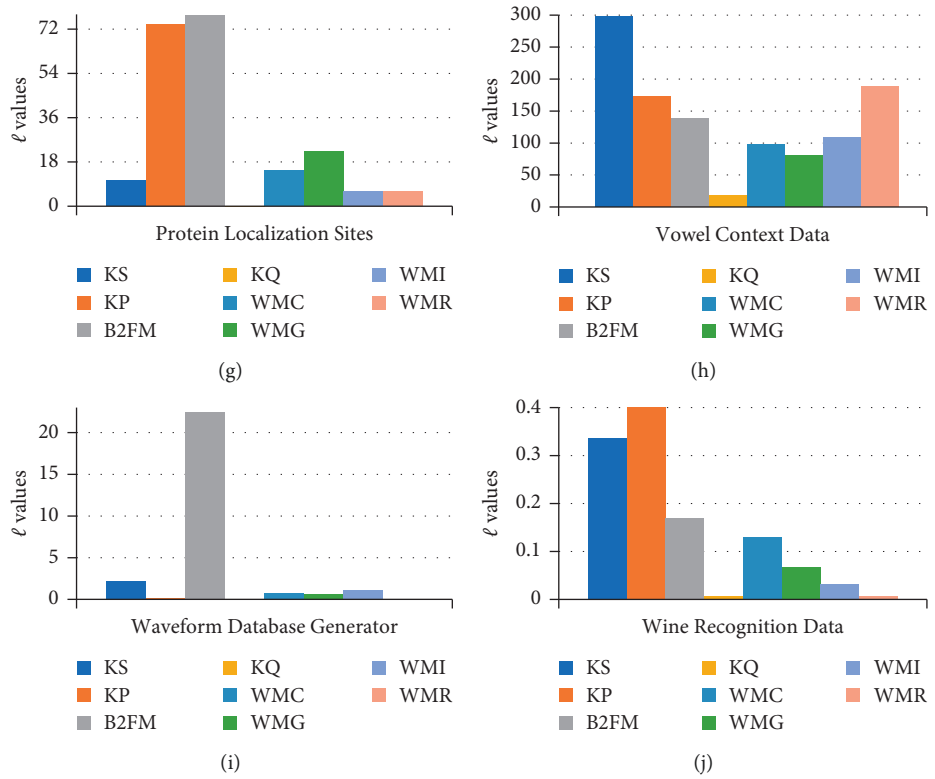

Figure 2: The ℓ values for 8 algorithms and 10 datasets having more than 2 classes.

Table 3: The means of the A_c, F_m, R_e, and P_r values (%) for 14 datasets and the standard and the weighted K-means++ clustering approaches.

Means	Standard	Correlation	Gain ratio	Info gain	Relief
$\overline{A_c}$	84.43	84.54*	85.09*	85.50**	84.48*
$\overline{F_m}$	74.01	74.37*	75.32*	76.05**	73.49
$\overline{R_e}$	78.50	77.42*	79.69*	80.03**	76.54
$\overline{P_r}$	70.71	71.48*	72.52*	73.23**	71.13*

*The A_c, F_m, R_e and P_r values (%) higher than the standard approach. **The highest means of A_c, F_m, R_e, and P_r values (%).

means of the A_c, F_m, R_e, and P_r values. Across 14 datasets, the average of the A_c values of information gain ($\overline{A_c}$) is 85.50%, whereas the standard value is 84.43%; the average of F_m values of information gain ($\overline{F_m}$) is 76.05%, whereas the standard value is 74.01%; the average of R_e values for information gain ($\overline{R_e}$) is 80.03%, whereas the standard value is 78.50%; and the average of P_r values for information gain ($\overline{P_r}$) is 73.23%, whereas the standard value is 70.71%. Gain ratio and correlation techniques also have higher average values than standard; however, it is observed that relief cannot obtain higher average values than standard for all measurements (specifically, $\overline{F_m}$ and $\overline{R_e}$ values are lower).

As a result, the values in Tables 2 and 3 support the claim that the most successful approach for determination of the weights of the features is weighted K-means++ clustering with information gain. Furthermore, in comparison with the other feature selection methods, relief is not efficient for determining the weights.

Traditional K-means++ clustering and four weighted K-means++ clustering approaches incorporating feature selection methods were compared in this analysis. The number of clusters for each dataset was assumed to be the original class number. The SSE values in Figure 3 show that distance metrics do not affect them significantly, whereas it can be seen clearly that all weighted clustering approaches incorporating feature selection methods produce fewer errors than traditional clustering for all datasets. Moreover, the weighted clustering decreased the SSE values by a large proportion, with information gain reducing SSE by 33% for the "Blocks Classification" dataset; relief reducing SSE by 75% for the "Cardiac Arrhythmia Database" dataset; gain ratio reducing SSE by 45% for the "Column 3C Weka" dataset; relief reducing SSE by 67% for the "Glass Identification Database" dataset; gain ratio, information gain, and relief reducing SSE by 73% for the "Heart Statlog" dataset; gain ratio and relief reducing SSE by 35% for the "Iris Plants Database" dataset; gain ratio reducing SSE by 54% for the "Johns Hopkins University Ionosphere Database" dataset; gain ratio reducing SSE by 34% for the "Optical Recognition of Handwritten Digits" dataset; relief reducing SSE by 56% for the "Pima Indians Diabetes Database" dataset; gain ratio reducing SSE by 42% for the "Protein localization sites" dataset; relief reducing SSE by 77% for the "Vowel Context Data" dataset; information gain reducing SSE by 51% for the "Waveform Database Generator" dataset; information gain reducing SSE by 47% for the "Wine Recognition Data" dataset; and relief reducing SSE by 76% for the "Wisconsin Breast Cancer" dataset. In particular, relief, gain ratio, and

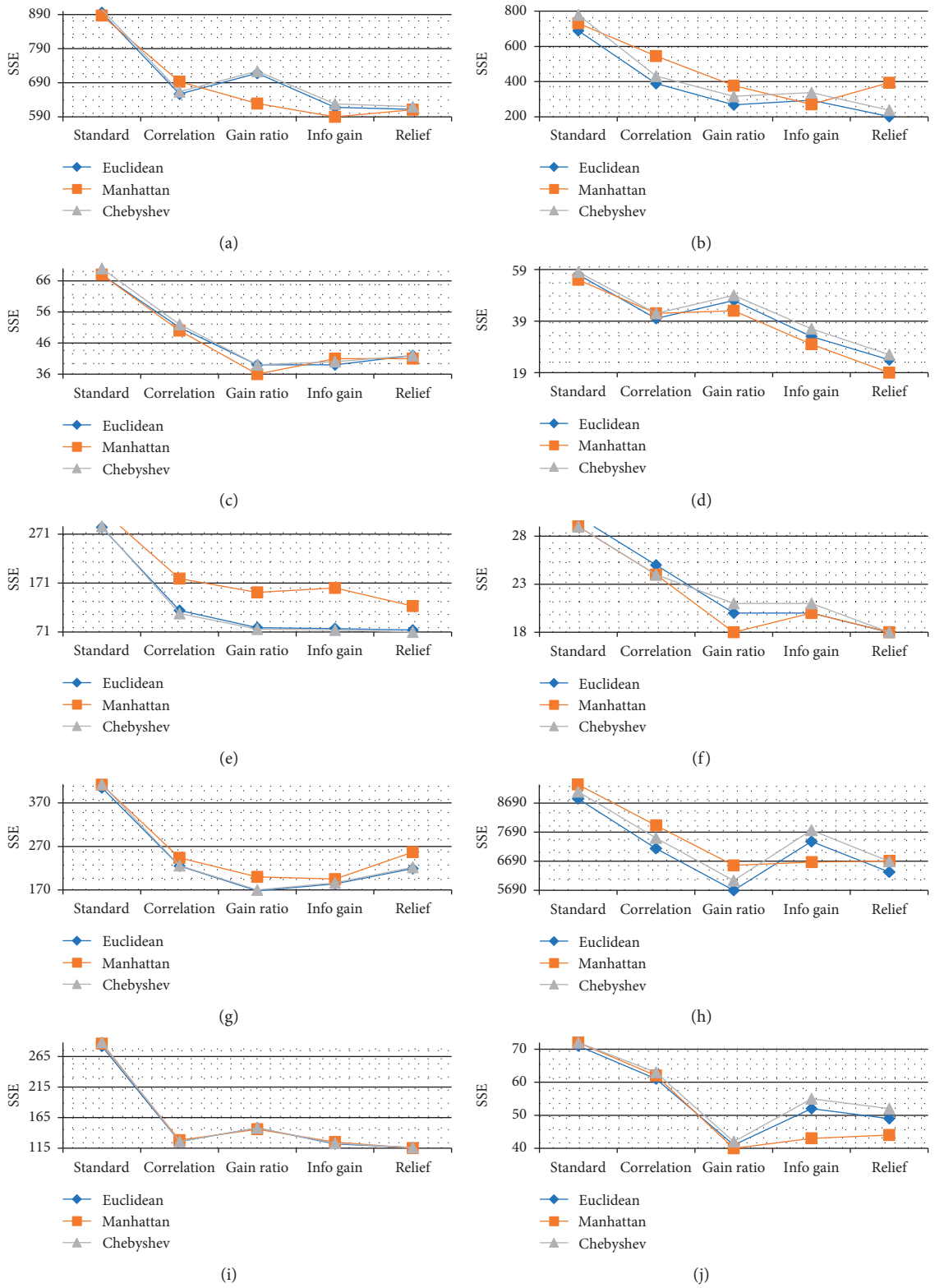

Figure 3: Continued.

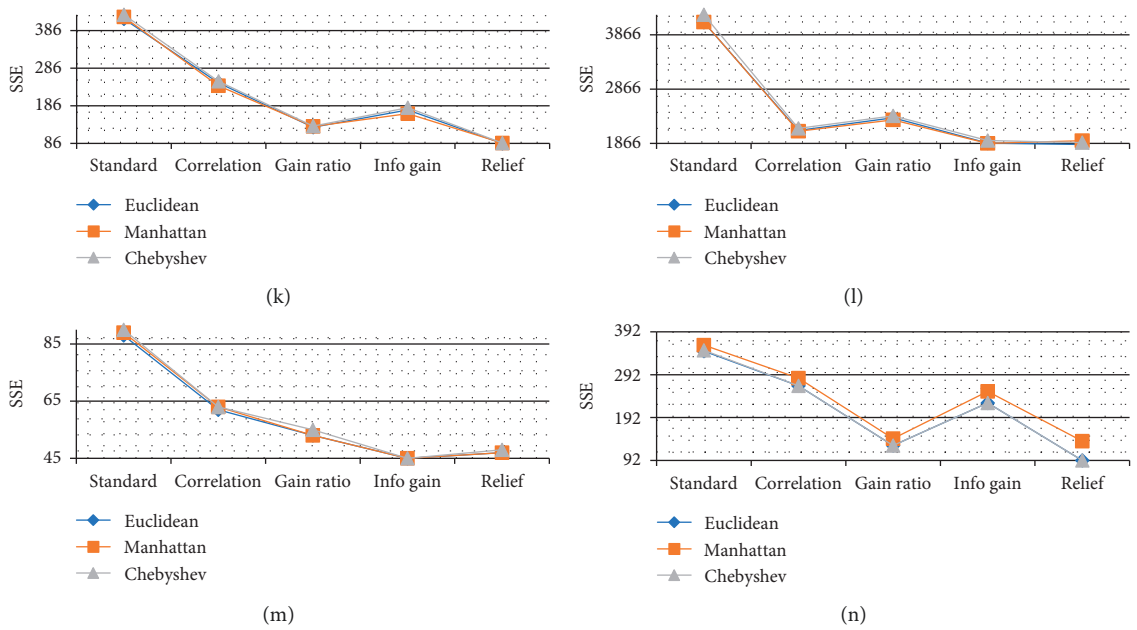

Figure 3: The SSE values that were obtained by using Euclidean, Manhattan, and Chebyshev distance measurements for 14 datasets: (a) Blocks Classification; (b) Cardiac Arrhythmia Database; (c) Column 3C Weka; (d) Glass Identification Database; (e) Heart Statlog; (f) Iris Plants Database; (g) Johns Hopkins University Ionosphere Database; (h) Optical Recognition of Handwritten Digits; (i) Pima Indians Diabetes Database; (j) Protein Localization Sites; (k) Vowel Context Data; (l) Waveform Database Generator; (m) Wine Recognition Data; (n) Wisconsin Breast Cancer.

information gain are better than the other approaches in decreasing errors. Moreover, relief is the most successful approach to reducing SSE in proportion to its accuracy results.

In Figure 2, the approaches are compared with each other separately for each dataset, revealing that KS, KP, and B2FM have generally higher differences than the other proposed approaches. However, because the ℓ values were obtained between different ranges for each dataset the ℓ values were normalized between 0 and 100 for each dataset, the average of each was calculated, and the results were obtained as the percentage value to enable clear comparison (Table 4). Finally, compatibility values were computed as 40% for KS, 50% for KP, and 39% for B2FM, while much higher values of 96% for KQ, 65% for WMC, 80% for WMG, 79% for WMI, and 69% for WMR were computed.

The results support the claim that KQ is the most compatible approach, achieving 96% for all datasets. Additionally, the weighted approaches for KS, KP, and B2FM are more compatible than the traditional approaches. Furthermore, WMG and WMI have higher compatible values than the other weighted approaches, WMC and WMR. As a result, because B2FM has a 37% compatibility result, besides two most valuable features, it can be claimed that the other features have high mapping efficiency, as well.

Table 4: The compatibility results (%) on 10 datasets for the traditional and the proposed approaches.

KS	KP	B2FM	KQ	WMC	WMG	WMI	WMR
40	50	37	96*	65	80	79	69

*The highest compatibility result (%).

7. Conclusions and Future Work

In this paper, five mapping approaches containing hybrid algorithms were proposed and presented in detail. One involves K-means++ and QGA used together, and the others are weighted approaches based on four different feature selection methods (Pearson's correlation, the gain ratio, information gain, and relief methods). According to the experimental results, by evaluation of DRV-P-MC as an optimization problem, mapping K-means++ centroids using QGA emerges as the most successful approach, with the lowest differences between the multivariate and 2D structures for all datasets and the highest compatibility value (96%). Conversely, despite its reputation as the most popular mapping algorithm, SOM did not perform as expected in compatibility tests in comparison with the proposed optimization algorithm. However, the weighted approaches based on information gain, gain ratio, and relief provided more consistent clusters than traditional clustering with means indicating approximately 50% lower SSE values. Additionally, the experiments demonstrate that weighted clustering by using information gain provides the highest average accuracy (85.5%) for all 14 datasets and achieves more accurate placements on the map than PCA, while preserving the most valuable features, unlike PCA. As a result, this study claims that the most successful hybrid approach for mapping the centroids is the integration of weighted clustering with the information gain feature selection method and QGA.

This paper proposes novel approaches that are open to improvement by the use of different hybrid structured algorithms. For example, all approaches in the paper have focused on K-means++ clustering; however, the proposed approaches can be adapted for other partitional clustering algorithms. In addition, it can be foreseen that, for 3-dimensional mapping, the weighted approach based on information gain will show higher performance than the other approaches because two most valuable features are preserved for two dimensions, and the third one can be computed by the weighted average of other features. Moreover, this weighted approach offers utility by means of its consistency and plain algorithmic structure for the visualization of stream data because the updatable entries of clustering features can determine a summary of clusters' statistics simply. Moreover, the visualization of a large-scale dataset can be achieved by using the proposed approaches after data summarization by clustering with many clusters.

The proposed algorithms can be used extensively across a wide range of fields including medicine, agricultural biology, economics, and engineering sciences. These fields have datasets potentially containing multidimensional structures. If there are lots of instances in this multidimensional structure, the size of the dataset mitigates against understanding the dataset and makes decisions difficult. Our algorithms presented in this paper shrink the dataset size vertically by means of clustering and horizontally by means of dimensional reduction.

Notation

D: Dataset
F: Set of the features in D
f: Length of F
I: Set of the instances in D
t: Target attribute
n: Length of I
C: Set of multivariate values of the centroids
E: Set of 2D values of the centroids
k: Number of centroids
M: Matrix where the centroids are mapped
r: Row length of M
c: Column length of M
ℓ: Difference among the multivariate centroids and their obtained 2D projections
S: Matrix where the distances among the multivariate centroids in C located in M are put
T: Matrix where the distances among 2D positions of the centroids in C located in M are put
Z: Matrix where the values obtained after the subtraction of S and T are put
G: Set of subcategories in a feature
g: Length of G
B: Set of subcategories in t
b: Length of B.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.

[2] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[3] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[4] D. Zhang and Z. H. Zhou, "(2D) 2PCA: two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1, pp. 224–231, 2005.

[5] A. S. N. Curman, M. A. Andresen, and P. J. Brantingham, "Crime and place: a longitudinal examination of street segment patterns in Vancouver, BC," *Journal of Quantitative Criminology*, vol. 31, no. 1, pp. 127–147, 2015.

[6] S. Cicoria, J. Sherlock, M. Muniswamaiah, and L. Clarke, "Classification of titanic passenger data and chances of surviving the disaster," in *Proceedings of Student-Faculty Research Day, CSIS, Pace University*, White Plains, NY, USA, May 2014.

[7] S. Hadi and K. U. J. Sumedang, "Clustering techniques implementation for detecting faces in digital image," *Pengembangan Sistem Penciuman Elektronik Dalam Mengenal Aroma Campuran*, vol. 1, pp. 1–9, 2006.

[8] W. Fang, K. K. Lau, M. Lu et al., "Parallel data mining on graphics processors," Technical Report HKUST-CS08-07, Hong Kong University of Science and Technology, Hong Kong, China, 2008.

[9] M. Nitsche, *Continuous Clustering for a Daily News Summarization System*, Hamburg University of Applied Sciences, Hamburg, Germany, 2016.

[10] V. Wagh, M. X. Doss, D. Sabour et al., "Fam40b is required for lineage commitment of murine embryonic stem cells," *Cell Death and Disease*, vol. 5, no. 7, p. e1320, 2014.

[11] N. Raabe, K. Luebke, and C. Weihs, "KMC/EDAM: a new approach for the visualization of K-means clustering results," in *Classification the Ubiquitous Challenge*, pp. 200–207, Springer, Berlin, Germany, 2005.

[12] A. M. De Silva and P. H. W. Leong, *Grammar-Based Feature Generation for Time-Series Prediction*, Springer, Berlin, Germany, 2015.

[13] Y. Zhang and G. F. Hepner, "The dynamic-time-warping-based K-means++ clustering and its application in phenoregion delineation," *International Journal of Remote Sensing*, vol. 38, no. 6, pp. 1720–1736, 2017.

[14] T. Sharma, D. Shokeen, and D. Mathur, "Multiple *k*-means++ clustering of satellite image using Hadoop MapReduce and Spark," 2016, https://arxiv.org/abs/1605.01802.

[15] F. Dzikrullah, N. A. Setiawan, and S. Sulistyo, "Implementation of scalable K-means++ clustering for passengers' temporal pattern analysis in public transportation system (BRT Trans Jogja case study)," in *Proceedings of IEEE*

International Annual Engineering Seminar (InAES), pp. 78–83, Daejeon, Korea, October 2016.

[16] A. P. Nirmala and S. Veni, "An enhanced approach for performance improvement using hybrid optimization algorithm with K-Means++ in a virtualized environment," *Indian Journal of Science and Technology*, vol. 9, no. 48, 2017.

[17] L. Wang, S. Li, R. Chen, S. Y. Liu, and J. C. Chen, "A segmentation and classification scheme for single tooth in microCT images based on 3D level set and K-means++," *Computerized Medical Imaging and Graphics*, vol. 57, pp. 19–28, 2017.

[18] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, USA, pp. 1027–1035, July 2007.

[19] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable K-means++," *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 622–633, 2012.

[20] A. Kapoor and A. Singhal, "A comparative study of *k*-means, *k*-means++ and fuzzy *c*-means clustering algorithms," in *Proceedings of IEEE 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, pp. 1–6, Gwalior, India, February 2017.

[21] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52–65, 2013.

[22] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[23] T. Nanda, B. Sahoo, and C. Chatterjee, "Enhancing the applicability of Kohonen self-organizing map (KSOM) estimator for gap-filling in hydrometeorological timeseries data," *Journal of Hydrology*, vol. 549, pp. 133–147, 2017.

[24] J. E. Chu, B. Wang, J. Y. Lee, and K. J. Ha, "Boreal summer intraseasonal phases identified by nonlinear multivariate empirical orthogonal function–based self-organizing map (ESOM) analysis," *Journal of Climate*, vol. 30, no. 10, pp. 3513–3528, 2017.

[25] A. Voutilainen, N. Ruokostenpohja, and T. Välimäki, "Associations across caregiver and care recipient symptoms: self-organizing map and meta-analysis," *Gerontologist*, vol. 58, no. 2, pp. e138–e149, 2017.

[26] N. Kanzaki, T. Kataoka, R. Etani, K. Sasaoka, A. Kanagawa, and K. Yamaoka, "Analysis of liver damage from radon, x-ray, or alcohol treatments in mice using a self-organizing map," *Journal of Radiation Research*, vol. 58, no. 1, pp. 33–40, 2017.

[27] W. P. Tsai, S. P. Huang, S. T. Cheng, K. T. Shao, and F. J. Chang, "A data-mining framework for exploring the multi-relation between fish species and water quality through self-organizing map," *Science of the Total Environment*, vol. 579, pp. 474–483, 2017.

[28] Y. Dogan, D. Birant, and A. Kut, "SOM++: integration of self-organizing map and k-means++ algorithms," in *Proceedings of International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 246–259, New York, NY, USA, July 2013.

[29] F. Viani, A. Polo, E. Giarola, M. Salucci, and A. Massa, "Principal component analysis of CSI for the robust wireless detection of passive target," in *Proceedings of IEEE International Applied Computational Electromagnetics Society Symposium-Italy (ACES)*, pp. 1-2, Firenze, Italy, March 2017.

[30] D. Tiwari, A. F. Sherwani, M. Asjad, and A. Arora, "Grey relational analysis coupled with principal component analysis for optimization of the cyclic parameters of a solar-driven organic Rankine cycle," *Grey Systems: Theory and Application*, vol. 7, no. 2, pp. 218–235, 2017.

[31] J. M. Hamill, X. T. Zhao, G. Mészáros, M. R. Bryce, and M. Arenz, "Fast data sorting with modified principal component analysis to distinguish unique single molecular break junction trajectories," 2017, https://arxiv.org/abs/1705.06161.

[32] K. Ishiyama, T. Kitawaki, N. Sugimoto et al., "Principal component analysis uncovers cytomegalovirus-associated NK cell activation in Ph+ leukemia patients treated with dasatinib," *Leukemia*, vol. 31, no. 1, pp. 203–212, 2017.

[33] A. D. Halai, A. M. Woollams, and M. A. L. Ralph, "Using principal component analysis to capture individual differences within a unified neuropsychological model of chronic post-stroke aphasia: revealing the unique neural correlates of speech fluency, phonology and semantics," *Cortex*, vol. 86, pp. 275–289, 2017.

[34] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, 1987.

[35] T. Hey, "Quantum computing: an introduction," *IEEE Computing and Control Engineering Journal*, vol. 10, no. 3, pp. 105–112, 1999.

[36] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum genetic optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 231–241, 2008.

[37] L. R. Silveira, R. Tanscheit, and M. M. Vellasco, "Quantum inspired evolutionary algorithm for ordering problems," *Expert Systems with Applications*, vol. 67, pp. 71–83, 2017.

[38] Z. Chen and W. Zhou, "Path planning for a space-based manipulator system based on quantum genetic algorithm," *Journal of Robotics*, vol. 2017, Article ID 3207950, 10 pages, 2017.

[39] G. Guan and Y. Lin, "Implementation of quantum-behaved genetic algorithm in ship local structural optimal design," *Advances in Mechanical Engineering*, vol. 9, no. 6, article 1687814017707131, 2017.

[40] T. Ning, H. Jin, X. Song, and B. Li, "An improved quantum genetic algorithm based on MAGTD for dynamic FJSP," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, pp. 1–10, 2017.

[41] D. Konar, S. Bhattacharyya, K. Sharma, S. Sharma, and S. R. Pradhan, "An improved hybrid quantum-inspired genetic algorithm (HQIGA) for scheduling of real-time task in multiprocessor system," *Applied Soft Computing*, vol. 53, pp. 296–307, 2017.

[42] P. Kumar, D. S. Chauhan, and V. K. Sehgal, "Selection of evolutionary approach based hybrid data mining algorithms for decision support systems and business intelligence," in *Proceedings of International Conference on Advances in Computing, Communications and Informatics*, pp. 1041–1046, Chennai, India, August 2012.

[43] N. Singhal and M. Ashraf, "Performance enhancement of classification scheme in data mining using hybrid algorithm," in *Proceedings of IEEE International Conference Computing, Communication & Automation (ICCCA)*, pp. 138–141, Greater Noida, India, May 2015.

[44] S. Z. Hassan and B. Verma, "Hybrid data mining for medical applications," in *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, pp. 523–543, Information Science Reference, New York, NY, USA, 2009.

[45] P. Thamilselvan and J. G. R. Sathiaseelan, "Image classification using hybrid data mining algorithms-a review," in *Proceedings of IEEE International Conference Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–6, Coimbatore, India, March 2015.

[46] B. Athiyaman, R. Sahu, and A. Srivastava, "Hybrid data mining algorithm: an application to weather data," *Journal of Indian Research*, vol. 1, no. 4, pp. 71–83, 2013.

[47] S. Sahay, S. Khetarpal, and T. Pradhan, "Hybrid data mining algorithm in cloud computing using MapReduce framework," in *Proceedings of IEEE International Conference Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 507–511, Ramanathapuram, India, 2016.

[48] Z. Yu, L. Li, Y. Gao et al., "Hybrid clustering solution selection strategy," *Pattern Recognition*, vol. 47, no. 10, pp. 3362–3375, 2014.

[49] P. Sitek and J. Wikarek, "A hybrid programming framework for modelling and solving constraint satisfaction and optimization problems," *Scientific Programming*, vol. 2016, pp. 1–13, 2016.

[50] E. Abdel-Maksoud, M. Elmogy, and R. Al-Awadi, "Brain tumor segmentation based on a hybrid clustering technique," *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 71–81, 2015.

[51] J. Zhu, C. H. Lung, and V. Srivastava, "A hybrid clustering technique using quantitative and qualitative data for wireless sensor networks," *Ad Hoc Networks*, vol. 25, pp. 38–53, 2015.

[52] M. A. Rahman and M. Z. Islam, "A hybrid clustering technique combining a novel genetic algorithm with K-means," *Knowledge-Based Systems*, vol. 71, pp. 345–365, 2014.

[53] R. Jagtap, "A comparative study of utilization of single and hybrid data mining techniques in heart disease diagnosis and treatment plan," *International Journal of Computer Applications*, vol. 123, no. 8, pp. 1–6, 2015.

[54] M. Dash, H. Liu, and J. Yao, "Dimensionality reduction of unsupervised data," in *Proceedings of IEEE Ninth International Conference Tools with Artificial Intelligence*, pp. 532–539, Newport Beach, CA, USA, November 1997.

[55] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 245–250, San Francisco, CA, USA, August 2001.

[56] A. Goh and R. Vidal, "Clustering and dimensionality reduction on Riemannian manifolds," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–7, Anchorage, AK, USA, June 2008.

[57] D. Napoleon and S. Pavalakodi, "A new method for dimensionality reduction using K-means clustering algorithm for high dimensional data set," *International Journal of Computer Applications*, vol. 13, no. 7, pp. 41–46, 2011.

[58] S. K. Samudrala, J. Zola, S. Aluru, and B. Ganapathysubramanian, "Parallel framework for dimensionality reduction of large-scale datasets," *Scientific Programming*, vol. 2015, Article ID 180214, 12 pages, 2015.

[59] J. P. Cunningham and M. Y. Byron, "Dimensionality reduction for large-scale neural recordings," *Nature Neuroscience*, vol. 17, no. 11, pp. 1500–1509, 2014.

[60] L. Demarchi, F. Canters, C. Cariou, G. Licciardi, and J. C. W. Chan, "Assessing the performance of two unsupervised dimensionality reduction techniques on hyperspectral APEX data for high resolution urban land-cover mapping," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 166–179, 2014.

[61] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for K-means clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2015.

[62] A. T. Azar and A. E. Hassanien, "Dimensionality reduction of medical big data using neural-fuzzy classifier," *Soft Computing*, vol. 19, no. 4, pp. 1115–1127, 2015.

[63] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu, "Dimensionality reduction for K-means clustering and low rank approximation," in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 163–172, Portland, OR, USA, June 2015.

[64] X. Zhao, F. Nie, S. Wang, J. Guo, P. Xu, and X. Chen, "Unsupervised 2D dimensionality reduction with adaptive structure learning," *Neural Computation*, vol. 29, no. 5, pp. 1352–1374, 2017.

[65] S. Sharifzadeh, A. Ghodsi, L. H. Clemmensen, and B. K. Ersbøll, "Sparse supervised principal component analysis (SSPCA) for dimension reduction and variable selection," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 168–77, 2017.

[66] Z. Yu, X. Zhu, H. S. Wong, J. You, J. Zhang, and G. Han, "Distribution-based cluster structure selection," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3554–3567, 2017.

[67] Z. Yu and H. S. Wong, "Quantization-based clustering algorithm," *Pattern Recognition*, vol. 43, no. 8, pp. 2698–2711, 2010.

[68] Z. Wang, Z. Yu, C. P. Chen et al., "Clustering by local gravitation," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1383–1396, 2018.

[69] Z. Yu, L. Li, J. Liu, J. Zhang, and G. Han, "Adaptive noise immune cluster ensemble using affinity propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3176–3189, 2015.

[70] Y. Dogan, D. Birant, and A. Kut, "A new approach for weighted clustering using decision tree," in *Proceedings of International Conference on INISTA*, pp. 54–58, Istanbul, Turkey, September 2011.

[71] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.