



Feature Subset Selection Based on Differential Evolution Algorithm and K-Mean Clustering

Talal F. Hussein^{*}, Fatima M. Hasan^{ID}, Hanadi D. Saleem^{ID}, Omar S. Qasim^{ID}

Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Mosul 41002, Iraq

Corresponding Author Email: talal.math@uomosul.edu.iq

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.120214>

ABSTRACT

Received: 28 April 2024

Revised: 8 July 2024

Accepted: 15 July 2024

Available online: 28 February 2025

Keywords:

K-mean clustering, differential evolution algorithm, feature selection, data mining

The K-means clustering technique securely used to cluster similar data points is normally inherent to spherical clusters with a centroid. The methodology proposed in this study is the DEAK-means method that incorporates the differential evolution algorithm (DEA) and K-means clustering to enhance the accuracy of clustering ignored data sets. With the help of DEAK-means, all the search spaces can be explored systematically and optimal features for classification can be identified hence clustering performance is boosted. The experimental results on 5 datasets indicate that in comparison with the commonly used K-means, DEAK-means yield better silhouette value and average selected features. This new hybrid of differential evolution and K-means algorithm proves to be a solution to discover problems that arose with the selection of features and clustering of the unlabeled dataset.

1. INTRODUCTION

Feature selection is one of the most important steps in the preprocessing of data especially in the field of machine learning and data mining. It is primarily used to eliminate all the undesired features from the data set to improve the capability of the predictive model by choosing only those features that have a strong impact on the model. Besides reducing the amount of computations required by the machine, feature selection helps the accuracy and the readability of the model [1, 2]. This process helps in dealing with big data in particular cases where many features may in fact be unhelpful or even damaging to the model. To ensure that feature selection is effective, filter methods, wrapper methods as well as embedded methods are used. Filter methods use statistical measures to assess features' relevance based on measures of association that do not involve the learning algorithm; on the other hand, wrapper methods employ the learning algorithm to assess feature subsets. Embedded methods select features while the model is being trained as a part of the model training process. As a result of developing feature selection strategies, the overburdened and wasteful models can be avoided easily; hence it is an inevitable component of current analysis and modeling processes for the researchers [3].

Differential evolution (DE) is a type of evolutionary algorithm (EA) that utilizes real numbers to represent variable values in order to address optimization problems in a continuous domain. DE, among the top EA, forms continuous optimization problems using variables with actual values, was first created by Storn and Price and is well-known for its robustness, simplicity, speed, and usability [4]. DE has a strong track record of effectively resolving optimization issues and has been utilized in various fields such as power control systems, chemical engineering, and clustering simultaneous

transit network design, among others. The algorithm utilizes processes of adaptation, emergence, and learning to try to improve an option across many generations. DE creates new solutions in each iteration by perturbing the current potential solutions with a scaled difference between two extra solutions randomly selected from the population, compared with other EA that recombines solutions within the limits of a probabilistic scheme. Differential evolution for feature selection in clustering datasets has the advantages of being robust in noisy environments and being simple to implement, potential sluggish convergence, and difficulties handling constraints or high-dimensional spaces [5].

The training outcomes of a machine learning model can be significantly improved and accelerated by carefully selecting a subset of high-correlation, non-redundant features from the feature set [6]. There are two main categories of feature selection algorithms that depend on the evaluation criterion: filter feature selection and wrapper feature selection. Filter feature selection methods evaluate the value of each characteristic before selecting a feature subset. For example, the well-known relief algorithm introduced by Kira and Rendell measures the degree of feature discrimination by computing the distance between similar and different samples. On the other hand, wrapper feature selection algorithms, such as the genetic algorithm, SVM-RFE, the estimation of distribution algorithm, and the differential evolution algorithm, combine feature selection and learning algorithms to choose a feature subset based on the learning outcome as the evaluation criterion [7, 8].

Clustering is a powerful data analysis technique that enables the categorization of data objects into groups based on their properties and relationships. This technique has diverse applications in fields such as data mining, Knowledge discovery, pattern recognition, and vector quantization. There

are two primary types of clustering algorithms: hierarchical and partitional clustering. Hierarchical clustering is a method that divides the data into a series of partitions, ranging from a single cluster that contains all objects to n clusters, each containing only one object. That hierarchy is depicted by a dendrogram, which is a tree-like diagram that depicts the relationships between groups at various levels. In contrast, partitional clustering algorithms divide data into a predetermined number of clusters, which are frequently provided by the user. Partitional clustering methods include fuzzy c-means and k-means. Both hierarchical and partitional clustering algorithms have benefits and drawbacks, and the methodology chosen is decided by the data is specific qualities and the desired analysis results [9]. One the other hand, partitional clustering algorithms divide the collection of data objects into non-overlapping clusters, placing each object in exactly one cluster. The K-means technique is a widely used partitional clustering algorithm, but it has some drawbacks such as convergence to local optima depending on how the initial cluster centroids are chosen [10]. In this paper, the objective is to improve the precision of unlabeled data gathering, such as gene analysis, by actively exploring research domains. To improve clustering results, the suggested strategy focuses on identifying the best classification characteristics. The study's findings demonstrate that the suggested method outperforms existing methods in terms of average feature selection and data similarity within clusters, as evidenced by experimentation and comparison on 5 datasets.

The rest of the essay is divided into the following sections. In Section 2, the original K-means clustering's structure is described. Introduced in Section 3 is the differential evolution algorithm. Our proposed strategy is presented in Section 4. On various data sets, Section 5 displays the experimental findings. Conclusions are provided in Section 6 at the end.

2. K-MEANS CLUSTERING

The K-means algorithm divides a dataset into K unique subgroups or clusters, which is a highly successful unsupervised clustering technique. Reducing the sum of the squared distances between data points and cluster centroids is the main goal of K-means [11]. This method keeps a considerable spacing between clusters while making sure that each cluster's data points are as comparable as possible. By calculating the average value of all data points in a cluster, K-means identifies the centroid, which represents that cluster. The algorithm continues to iterate until the centroids are stable, and the final result is a set of K clusters that optimize the homogeneity (similarity) of data points within each cluster. K-means is widely used in fields such as machine learning, image processing, and data mining because it can efficiently group large datasets based on their inherent patterns and structures [12, 13]:

$$D(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots (x_n - y_n)^2} \quad (1)$$

The K-means algorithm works as shown below:

Algorithm: K-Means Algorithm

1. K , the number of clusters, should be indicated. After shuffling the dataset initially, initialize the
 2. centroids by selecting K data points at random for the centroids.
-

-
3. Continue iterating until the centroids stop changing. In other words, the grouping of data points remains constant.
 4. The summation of the squared lengths between each centroid and data point is carried out.
 5. Place each data point in the most nearby cluster (centroid).
 6. By averaging the data points in each cluster, calculate its centroids.
-

3. DIFFERENTIAL EVOLUTION ALGORITHM

Differential evolution (DE) is a widely used optimization algorithm that is particularly effective for solving continuous optimization problems. DE has proven to be more resilient and less biased when compared to another evolutionary algorithm (EA) like genetic algorithm (GA) and Particle Swarm Optimization (PSO). For optimal results, it is essential to carefully adjust the three key parameters of DE optimization: population size, crossover rate, and scaling factor [PS]. Over the past few decades, a lot of research has been conducted to assess the impact of these factors on DE performance and to find the optimal values for them. At the start of the optimization process, DE produces a population of persons at random with a uniform distribution across the decision space (N). The algorithm then employs each of the three main evolutionary operators-mutation, crossover, and selection-iteratively to enhance each member (x_i^{\rightarrow}) in the population, ultimately producing results that are close to ideal [14].

The conventional DE method, developed by Storn and Price, has three main operators, each of which is described in depth in the subsections that follow factors utilized in the standard DE algorithm, as originally formulated by Storn and Price [5, 15].

3.1 Mutation operator

A mutation results in a putative progeny (or mutant vector) as follows:

$$\vec{v}_{i,g+1} = \vec{x}_{r_1,g} + F(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}), r_1 \neq r_2 \neq r_3 \neq i \quad (2)$$

where, the scale factor F controls how much the disparity in variation between the two vectors is amplified. $\vec{x}_{r_2,g}$ and $\vec{x}_{r_3,g}(\vec{x}_{r_2,g} - \vec{x}_{r_3,g})$, to determine how distant the produced offspring should be from the point $\vec{x}_{r_1,g}$. Whereas $\vec{x}_{r_1,g}$, $\vec{x}_{r_2,g}$ and $\vec{x}_{r_3,g}$ are three mutually unique people chosen at random from the population who are not equal to one another, F normally falls within the range [0,1] [16].

Eq. (2), often known as the DE/rand/1 scheme, represents the most basic sort of mutation.

As an alternative, the *DE/best/1* technique can be used to create a mutant vector by including the population member with the highest fitness value of that generation:

$$\vec{v}_{i,g+1} = \vec{x}_{best,g} + F \times (\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) \quad (3)$$

where, x is the generation g is top-performing individual vector $\vec{x}_{best,g}$.

The following DE mutation variations have been reported in the literature.

$$\begin{aligned} & DE/cur - to - best/1 \\ \vec{v}_{i,g+1} &= \vec{x}_{r_1,g} + F \times (\vec{x}_{best,g} - \vec{x}_{r_1,g}) + F \times \\ & (\vec{x}_{r_2,g} - \vec{x}_{r_3,g}), \\ & r_1 \neq r_2 \neq r_3 \neq i \end{aligned} \quad (4)$$

$$\begin{aligned} & DE/rand/2 \\ \vec{v}_{i,g+1} &= \vec{x}_{r_1,g} + F \times (\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) \\ & + F \times (\vec{x}_{r_4,g} - \vec{x}_{r_5,g}), \\ & r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i \end{aligned} \quad (5)$$

$$\begin{aligned} & DE/best/2 \\ \vec{v}_{i,g+1} &= \vec{x}_{best,g} + F \times (\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) + F \times \\ & (\vec{x}_{r_3,g} - \vec{x}_{r_4,g}), \\ & r_1 \neq r_2 \neq r_3 \neq r_4 \neq i \end{aligned} \quad (6)$$

$$\begin{aligned} & DE/rand - to - best/2 \\ \vec{v}_{i,g+1} &= \vec{x}_{r_1,g} + F \times (\vec{x}_{best,g} - \vec{x}_{i,g}) \\ & + F \times (\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) \\ & + F \times (\vec{x}_{r_4,g} - \vec{x}_{r_5,g}), \\ & r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i \end{aligned} \quad (7)$$

3.2 Crossover operator

The recombination or crossover operator in Differential evolution (DE) is a key mechanism that enables the algorithm to explore the search space and generate new and ideal solutions [4]. The method can generate trial vectors that are compared to parent vectors by merging people from the previous generation with newly developed mutants at a predefined crossover rate. The better solutions are then selected to advance to the next generation, while the inferior once is discarded. This iterative process continues until the termination condition is met, resulting in a population of individuals that offer optimal solutions. The efficacy of the crossover operator is influenced by various factors such as selection strategies, mutation, and population size. The crossover operator is widely used in optimization problems due to its ability to enhance the diversity and quality of the population. Overall, the DE algorithm with its efficient recombination or crossover operator has proven to be a powerful and effective approach for solving complex optimization problems. The two most common crossover types in DE are exponential and binomial, both of which employ a random number generator and the crossover rate to obtain continuous results. The binomial crossover selects a single element from the mutant vector with a probability of cr , while the exponential crossover operates on the entire vector and replaces the elements with probabilities that exponentially decrease with distance from the selected element. By enabling the exchange of information between individuals in the population, the crossover operator enhances the diversity of the population and helps to prevent premature convergence to suboptimal solutions, ultimately leading to better overall performance of the DE algorithm.

Eq. (4) describes the process of binomial vectors in Differential evolution (DE). With a probability of cr for some components from the mutant vector and a likelihood of $1-cr$ for others from the current target vector, this method generates trial vectors. The condition $J=a_j$ ensures that at least one component is selected from the mutant vector. This process can effectively combine the information from both the mutant and target vectors, leading to better solutions. However, the

choice of crossover rate cr can greatly impact the performance of the algorithm. A high cr value can increase the diversity of the population, but it may also cause a premature convergence rate. But it may also decrease the diversity. Therefore, the optimal value of cr should be carefully selected based on the problem at hand [17].

$$u_{i,g+1}^j = \begin{cases} v_{i,g+1}^j & \text{rand}(j) \leq cr \text{ or } j = a_j \\ x_{i,g}^j & \text{otherwise} \end{cases} \quad (8)$$

in which $i=1, 2, 3, \dots$, and $j=1, 2, 3, \dots, N$. The j^{th} assessment of a uniform random number generator within $[0,1]$ is denoted by PS and $rand(j)$. The likelihood of a spanning between $[0,1]$ is represented by PS ; cr . To guarantee that at least one member of $u_{i,g+1}$ is taken from the mutant vector, a_j is defined as a randomly selected element.

Similar to a two-point crossover, an exponential crossover replicates L subsequent components (counted cyclically) from the mutant vector after a cut point is selected at random from $[2, N-1]$. The likelihood of changing the K^{th} element in the sequence $\{1, 2, 3, \dots, N\}$ ($L \leq N$) decreases exponentially with growing K . Algorithm 1 illustrates the pseudo-code required to generate an exponential crossover [18].

Algorithm 1 exponential crossover pseudo-code $u_{i,g} \leftarrow x_{i,g}, j$ randomly taken from $[1, M]$:

1. $L=1$;
2. while $rand[0,1] < cr$ and $L < N$ do
3. $u_{i,g}^j = v_{i,g}^j$
4. $j=(j+1) \text{ module } N$
5. $L=L+1$;
6. end while

3.3 Selection operator

Differential evolution (DE) algorithms “one-to-one spawning” selection method is a simple yet effective technique for selecting individuals for the next generation. This approach is comparable to the selection process employed in other swarm intelligence systems, like the PSO method, or particle swarm optimization. In order to determine which individual would survive and go on to the following generation ($g+1$), each individual in the trial vector is compared with the corresponding individual in the previous generation vector (g). The comparison is based on the fitness or profit value of the individuals, with the one having. The person with the highest fitness value is selected and the other is removed. One-to-one-spawning ensures that the best individuals are selected for the next generation, improving the overall quality of the population and convergence to optimal solutions. The most commonly used formula for this selection process is the greed selection method, which selects the individual with the higher fitness value for survival. This selection process allows for the retention of the fittest individuals in the population and ensures that their desirable traits are carried forward into the next generation. The one-to-one-spawning selection method, combined with other DE operators like mutation and crossover, enables DE to explore a wide range of solution spaces and converge to optimal solutions more efficiently [18].

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g+1}, & f(\vec{u}_{i,g+1}) < f(\vec{x}_{i,g}) \\ \vec{x}_{i,g}, & \text{otherwise} \end{cases} \quad (9)$$

The DE algorithm keeps running until one of two thresholds

is reached: the time allotted or the number of generations. The maximum permissible evaluations of the fitness (objective) function can also be used to predict when the function will stop. The algorithm representation is shown in Figure 1.

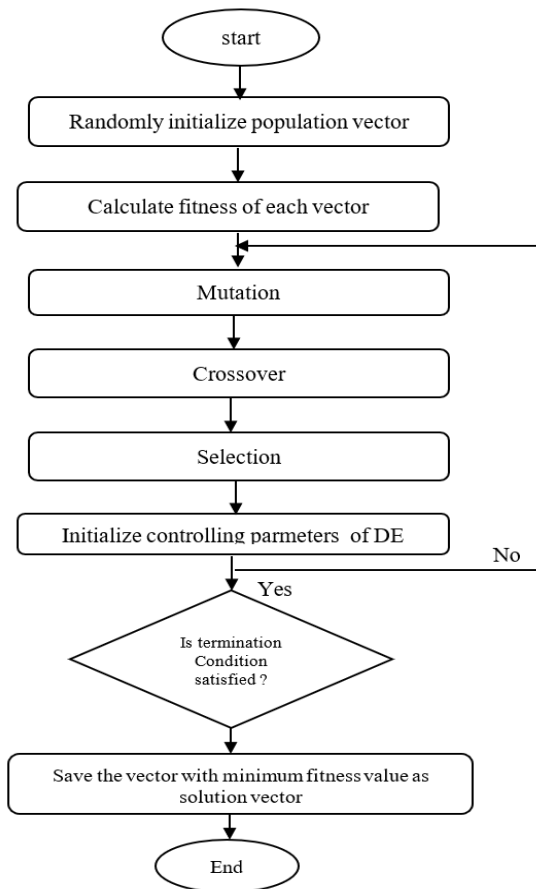


Figure 1. Detailed steps of differential evolution algorithm

3.4 BEST-matched value (BMV)

DE is a powerful optimization algorithm that can handle continuous variables. However, for discrete optimization problems, DE requires a mapping method that converts continuous variables into their corresponding binary values to calculate fitness values [19]. The recommended method for mapping continuous variables to binary values in DE is the BMV (best-mutant-vector) method, which is designed specifically for discrete optimization problems. In this method, each continuous variable is mapped to a binary variable using a threshold value that is determined based on the range of the variable. The BMV method is effective in improving the convergence speed. The DEA confirms a high efficiency of calculations in connection with a discrete optimization problem. This technique facilitates the ability of DEA in optimization tasks when dealing with discrete variables because it turns the continuous values of each person into discrete ones. Further, the BMV technique augments DEA's efficiency by directing the new individuals towards optimality by including features of the current generation's optimum solution. Hence, it allows DEA to make a deeper investigation of the search space to determine more efficient solutions. When integrating the BMV mapping approach with DEA's operators such as mutation, crossover, and selection, the system becomes greatly optimized. One amazing thing about this form of synergy is that the algorithm of one problem

can be transported to another field of application successfully and efficiently. Applying the strengths of BMV technique and DEA together in the research resolves problems of optimization with greater accuracy and speed [18, 20].

4. THE PROPOSED APPROACH

The proposed approach DEAK-means involves the enhancement of K-Means Clustering in high-dimensional dataset using the DEA. This method combines two existing optimization techniques, namely the K-means algorithm and the DEA. It works by adjusting the parameters of DEA and selecting the best features to find the optimal solution. The DEAK-means approach has been tested on five datasets from the UCI repository in order to evaluate its effectiveness in comparison to original algorithms. The results have shown that it can effectively improve the intra-cluster distances and Silhouette value. Figure 2 shows the depiction of the solution.

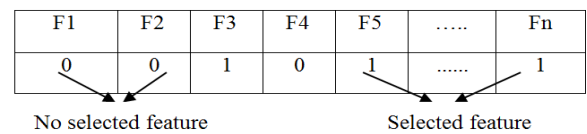


Figure 2. Representation of the features in DEAK-means

The DEAK-means algorithm utilizes the silhouette value as a measure of the quality of the clustering solution. Samples are assigned a value of 1 if they possess relevant features and 0 otherwise. The silhouette value for each sample is calculated by measuring the difference between the average distance to other samples in its own cluster and the average distance to samples in the nearest cluster. This provides a means of determining how well a sample is assigned to its own cluster compared to other clusters. Higher silhouette values indicate better matches to their own cluster and poorer matches to other clusters. The algorithm involves a series of steps that are tailored to optimize clustering performance.

Step:1	There are N populations, and T is the maximum number of iterations.
Step:2	The K number of clusters is produced at random from the $K \sim U(0,10)$ uniform distribution. The feature-representing rest positions are created as $U(0, 10)$ [21].
Step:3	Use the total within-cluster variance is used to define the fitness function. The positions are changed using Eq. (6). The selection of features is handled using binary DEA. The representation for each member in this case is a p-bit binary string. A binary space frequently updates the position using the transfer function. A transfer function can be used to create this binary vector. However, the result can only include binary values.
Step:4	$x^{t+1} = f(x) = \begin{cases} 1 & \text{if } T(\Delta x^{t+1}) > rand \\ 0 & \text{otherwise,} \end{cases} \quad (10)$ where, $rand \in [0,1]$ is a random number, $T(x) = (1 / (1 + \exp(-x)))$ is the sigmoid transfer function
Step:5	Steps 4 and 5 must be repeated in order to reach T.

5. RESULTS AND DISCUSSION

The ability of a DEAK-means algorithm to cluster samples was studied on five different public data with different characteristics that are accessible online. As a way of supporting its claim to higher potency than the obsolete K-means algorithm, the differential evolution algorithm (DEA) was used. Through the DEA method, it was possible to carry out the optimization of the Silhouette evaluation and the establishment of the most appropriate number of features for each dataset. Afterward, the DEAK-means proposed here were compared with the basic K-means algorithm about performance. Thus, the goal of this comparative analysis was to show that DEAK-means is capable of achieving better clustering results and more efficient feature selection in order to support the further study of the given method as the advanced approach to data clustering and analysis in a wide range of disciplines.

Five real-world datasets were used for the experiments and these came from the UCI Machine Learning Repository. These datasets are different in terms of size, number of attributes or features, and the actual number of clusters as shown in Table 1. On the same note, the approach ensures that the algorithm undergoes a diversified assessment depending on the datasets available in the given domain. Therefore, the results obtained on different data sets would give a clear indication of the applicability and the performance of the proposed method in different clustering and feature selection problems.

Table 1. Description of the datasets

Dataset Name	Number of Samples	Number of Features
Data 1 (Biodeg)	1055	41
Data 2 (Cmc)	1473	9
Data 3 (Glass)	214	9
Data 4 (Ionosphere)	315	34
Data 5 (Raisin)	900	7

To measure the performance of the applied algorithms, the Silhouette score is employed as the criterion. The Silhouette score is an external index that checks whether it is erroneous to partition the data set by increasing the distance between clusters as well as reducing the distance in constitution with the same cluster. This measurement evaluates the circle circumradius of the clusters and its next distance. It is determined by dividing the mean distance of a sample with all the points in its own cluster and then subtracting the average of the distance with the points in the nearest neighbor cluster. If a Silhouette score is higher for a specific sample, it means that this sample belongs to its own cluster and might contribute to the clustering quality. On the other hand, a low or negative Silhouette score would mean that the sample is possibly inappropriately clustered and that the configuration probably has more than what it should or less than what it actually should have of clusters. Therefore, the Silhouette score is an ideal measure for evaluating the clustering efficiency by addressing the factors of cohesion within the clusters and the separation of clusters from each other. The silhouette score ranges from -1 to +1. When the bulk of the items have high values, the layout of the clusters [6, 7] For point i , the silhouette widths i is defined:

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}} \quad (11)$$

where, $a(i)$ is the average distance between i and all other data points in the same cluster C_i , and $b(i)$ is the average distance between i and all other data points in the other cluster C_j .

$$b(i) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j) \quad (12)$$

where, $d(i, j)$ represents the distance between i and j .

The Silhouette value presented in Table 2 proves the fact that, compared to K-means, DEAK-means yields higher clustering accuracy for all referred datasets. These raise the performance level and imply that DEAK-means is a better algorithm for feature selection in clustering problems concerning the normal K-means algorithm. Moreover, it is shown that the DEAK-means concept is beneficial when dealing with the high-dimensional dataset, making it stable for using in several applications of cluster analysis. DEAK-means is the extension of K-means where a differential evolution algorithm is incorporated into K-means; this integration benefits from the advantages of both algorithms which leads to better clustering results and attributes selection (Figure 3).

Table 3 shows the comparative analysis of different methods and proves that the DEAK-means algorithm is better in feature selection on all the datasets. This means that on average and in response to the test employed regarding the formation of clusters, DEAK-means has the capacity of coming up with ideal clusters as expected. Moreover, the results show that DEAK-means proves to be superior to K-means not only in terms of clustering which we can observe through clustering quality measurement but also in terms of time complexity. This double benefit puts DEAK-means in a very good position for doing feature selection and clustering when working with big and/or high-dimensional data. Thus, DEAK-means enhances the differential evolution and K-means advantages improving the producing clustering accuracy and reducing the necessary computational time (Figure 4).

Table 2. The Silhouette value results for DEAK-means and K-means clustering accuracy

Dataset Name	Number of Samples	Number of Features
Data 1 (Biodeg)	0.3164	0.2325
Data 2 (Cmc)	0.5279	0.3480
Data 3 (Glass)	0.4914	0.4511
Data 4 (Ionosphere)	0.2888	0.2609
Data 5 (Raisin)	0.5625	0.5608

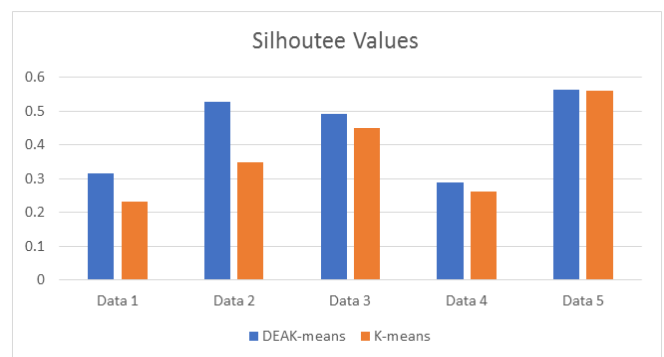


Figure 3. Comparison between DEAK-means and K-means in Silhouette value results

Table 3. Demonstrates a feature selection comparison between the DEAK-means and K-means algorithms

Dataset Name	Number of Samples	Number of Features
Data 1 (Biodeg)	22.4	41
Data 2 (Cmc)	3.8	9
Data 3 (Glass)	5	9
Data 4 (Ionosphere)	22	34
Data 5 (Raisin)	2.6	7

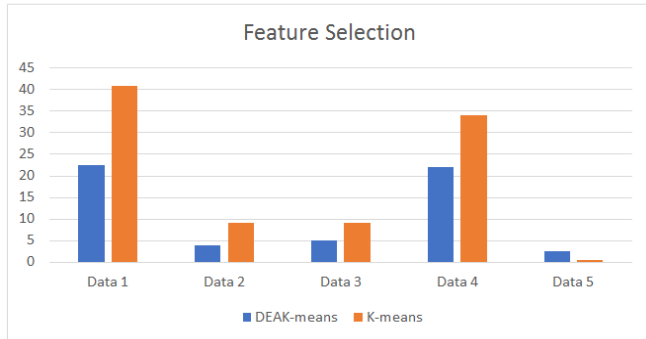


Figure 4. Comparison between DEAK-means and K-means in average feature selection

6. CONCLUSIONS

This work introduces the differential evolution algorithm (DEA) as a new method to improve K-means clustering. Therefore, the performance of the proposed DEAK-means algorithm was examined on five datasets where the primary metrics were intra-cluster distance and the average quantity of the selected features. Based on the results demonstrated in Tables 2-3 and Figures 3-4, it is concluded that the DEAK-means algorithm is more effective than the traditional K-means for the feature subset selection and silhouette values. Another study could therefore explore more validation and enhancement of the DEAK-means algorithm with a particular focus on the datasets from the different domains or on the alteration of the key concepts of the said algorithm. In practical terms, these findings hold important implications in areas such as bioinformatics and data analytics, providing enhanced efficiency and accuracy in assembly, which may lead to better data insights and decision-making processes in real-world applications.

REFERENCES

- [1] Ismael, O.M., Qasim, O.S., Algamal, Z.Y. (2021). A new adaptive algorithm for v-support vector regression with feature selection using Harris hawks optimization algorithm. *Journal of Physics: Conference Series*, 1897(1): 012057. <https://doi.org/10.1088/1742-6596/1897/1/012057>
- [2] Faris, H., Abukhurma, R., Almanaseer, W., Saadeh, M., Mora, A.M., Castillo, P.A., Aljarah, I. (2020). Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: A case from the Spanish market. *Progress in Artificial Intelligence*, 9: 31-53. <https://doi.org/10.1007/s13748-019-00197-9>
- [3] Kashef, S., Nezamabadi-pour, H., Nikpour, B. (2018). Multilabel feature selection: A comprehensive review and guiding experiments. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2): e1240. <https://doi.org/10.1002/widm.1240>
- [4] Das, S., Mullick, S.S., Suganthan, P.N. (2016). Recent advances in differential evolution—An updated survey. *Swarm and Evolutionary Computation*, 27: 1-30. <https://doi.org/10.1016/j.swevo.2016.01.004>
- [5] Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y., Xu, J. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing*, 25: 5277-5298. <https://doi.org/10.1007/s00500-020-05527-x>
- [6] Qasim, O.S., Algamal, Z.Y. (2021). A gray wolf algorithm for feature and parameter selection of support vector classification. *International Journal of Computing Science and Mathematics*, 13(1): 93-102. <https://doi.org/10.1504/IJCSM.2021.114185>
- [7] Pattanshetti, T., Attar, V. (2019). Performance evaluation and analysis of feature selection algorithms. In *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2018, Volume 1*, pp. 47-60. https://doi.org/10.1007/978-981-13-1402-5_4
- [8] Tang, C., Zheng, X., Zhang, W., Liu, X., Zhu, X., Zhu, E. (2023). Unsupervised feature selection via multiple graph fusion and feature weight learning. *Science China Information Sciences*, 66(5): 152101. <https://doi.org/10.1007/s11432-022-3579-1>
- [9] Solikhun, S., Yasin, V., Nasution, D. (2022). Optimization of the number of clusters of the K-means method in grouping egg production data in Indonesia. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, 4(1): 39-47. <https://doi.org/10.25139/ijair.v4i1.4328>
- [10] Al-Kababchee, S.G.M., Qasim, O.S., Algamal, Z.Y. (2021, May). Improving penalized regression-based clustering model in big data. *Journal of Physics: Conference Series*, 1897(1): 012036. <https://doi.org/10.1088/1742-6596/1897/1/012036>
- [11] Krishnasamy, G., Kulkarni, A.J., Paramesran, R. (2014). A hybrid approach for data clustering based on modified cohort intelligence and K-means. *Expert Systems with Applications*, 41(13): 6009-6016. <https://doi.org/10.1016/j.eswa.2014.03.021>
- [12] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S., Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, 2(3): 267-279. <https://doi.org/10.1109/TETC.2014.2330519>
- [13] Rylko, N., Stawiarz, M., Kurtyka, P., Mityushev, V. (2024). Study of anisotropy in polydispersed 2D micro and nano-composites by Elbow and K-Means clustering methods. *Acta Materialia*, 120116. <https://doi.org/10.1016/j.actamat.2024.120116>
- [14] Kupin, A., Kosei, M. (2024). Analysis of swarm intelligence algorithms. *System Technologies*, 3(152): 69-80. <https://doi.org/10.34185/1562-9945-3-152-2024-07>
- [15] Mohamed, A.W. (2017). An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems. *International Journal of Machine Learning and Cybernetics*, 8: 989-1007.

- <https://doi.org/10.1007/s13042-015-0479-6>
- [16] Dhal, K.G., Das, A., Sahoo, S., Das, R., Das, S. (2021). Measuring the curse of population size over swarm intelligence based algorithms. *Evolving Systems*, 12: 779-826. <https://doi.org/10.1007/s12530-019-09318-0>
- [17] Zhang, X., Yuen, S.Y. (2015). A directional mutation operator for differential evolution algorithms. *Applied Soft Computing*, 30: 529-548. <https://doi.org/10.1016/j.asoc.2015.02.005>
- [18] Zeng, Z., Zhang, M., Chen, T., Hong, Z. (2021). A new selection operator for differential evolution algorithm. *Knowledge-Based Systems*, 226: 107150. <https://doi.org/10.1016/j.knosys.2021.107150>
- [19] Bala, I., Yadav, A., Kim, J.H. (2024). Optimization for cost-effective design of water distribution networks: A comprehensive learning approach. *Evolutionary Intelligence*, 1-33. <http://doi.org/10.1007/s12065-024-00922-x>
- [20] Li, S., Li, W., Tang, J., Wang, F. (2023). A new evolving operator selector by using fitness landscape in differential evolution algorithm. *Information Sciences*, 624: 709-731. <https://doi.org/10.1016/j.ins.2022.11.071>
- [21] Ali, I.M., Essam, D., Kasmari, K. (2020). A novel design of differential evolution for solving discrete traveling salesman problems. *Swarm and Evolutionary Computation*, 52: 100607. <https://doi.org/10.1016/j.swevo.2019.100607>