Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

Vol. 6, No. 3, September, 2025

A Self-Adaptive Weights for K-Means Classification Algorithm

Cui Chenghu [1], Arit Thammano [1*]

[1] *Computational Intelligence Laboratory, School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.*

Received 11 March 2025; Revised 18 July 2025; Accepted 07 August 2025; Published 01 September 2025

Abstract

This paper presents an improved K-means clustering algorithm that addresses the traditional algorithm's sensitivity to outlier and susceptibility to local optima by introducing an adaptive weight adjustment mechanism. It employs an exponential decay function to dynamically reduce the feature weights of outlier data points, effectively suppressing outliers while preserving the structure of the normal data. The proposed method retains the computational efficiency of standard K-means. Key contributions include: (a) A novel distance-based weighting strategy that progressively reduces the influence of noisy points, mitigating the impact of outliers on clustering performance. (b) An innovative form of "local dimensionality reduction" for outlier points via weight decay, which interferes only with the feature space of noisy regions while preserving the global topological structure of clean data. Extensive experiments on three benchmark datasets Iris (4-dimensional, balanced classes), Wine (13-dimensional, correlated features), and Wisconsin Breast Cancer Diagnosis (30-dimensional, imbalanced data) demonstrate the effectiveness of the approach. Compared to standard K-means, the proposed algorithm achieves accuracy improvements of 7.47% on Iris, 13.89% on Wine, and 19% on WBCD. This adaptive strategy offers a practical and efficient solution for clustering in noisy, high-dimensional environments, without the added complexity of mixture models.

Keywords: K-Means Classification; Adaptive Weights; Classification; Machine Learning.

1. Introduction

Clustering algorithms are fundamental tools in data mining, with applications ranging from customer segmentation to bioinformatics. Among them, the K-means algorithm is widely used due to its simplicity and efficiency [1]. In machine learning algorithms, these attributes are called features, and classification decisions are usually based on distance metrics in a spatial coordinate system [2]. Currently, K-means plays a central role in the field of data mining [3]. In addition, K-means has been widely used in various industries such as pharmaceuticals, manufacturing, robotics, and finance [4]. Machine learning aims to extract valuable potential information from existing datasets to predict future trends [5-10]. The K-means algorithm is very widely used in practical applications, it also has some obvious limitations, especially when dealing with datasets containing outliers or outliers. Since K-means relies on randomly initialized cluster centers and assigns data points based on minimizing the Euclidean distance, it tends to converge to local minima [11, 12]. In addition, the algorithm is highly sensitive to initial conditions, often resulting in overlapping clusters or blurred boundaries, which intensifies the impact of outliers and outliers on the final clustering results [13].

* Corresponding author: arit@it.kmitl.ac.th

http://dx.doi.org/10.28991/HIJ-2025-06-03-010

Although studies have attempted to improve the robustness of K-means by pausing, optimizing initialization methods, or introducing other heuristic algorithms, a core challenge has been fully addressed: how to dynamically adapt to the distribution of noise and outliers in the data. Most existing methods deal with noise statically or based on statics, such as using cleaning filters or dimensionality reduction techniques. However, these methods often fail or are inefficient when dealing with high-dimensional, dynamic, or unstructured data sets.

This study aims to introduce adaptive weighting mechanisms to dynamically adjust the importance of data points in alarms based on their distance from the cluster center. This method assigns more weight to the point distant from the cluster centre, effectively reducing the influence of noise and outliers on the overall centre. Related studies have shown that this method can effectively improve the accuracy and stability [14, 15]. To verify the effectiveness of this mechanism, we further compared and analyzed the current mainstream K-means improved algorithms, such as LBKC [16], SKM-AGR [17], OWAK-Means [18], KMF [19], and HCSA [20]. Although these methods have their own advantages, they have not yet achieved effective integration in terms of noise processing and adaptability. Therefore, the algorithm proposed in this paper, as an integrated hybrid model, will integrate the advantages of multiple algorithms on the basis of maintaining the core structure of K-means to adapt to the needs of different types of data. In addition, this paper will use statistical indicators such as rainbow and standard deviation to evaluate the classification model results to quantify the performance of the model under different data noise levels.

The structure of this paper is as follows: Section 2 reviews K-means and its improved algorithms and noise processing methods; Section 3 introduces the proposed adaptive weighted gain mechanism and its mathematical principles; Section 4 proposes experimental settings and benchmark datasets; Section 5 presents experimental results and compares them with traditional K-means; Section 6 summarizes the full paper and discusses future research directions.

2. Relative Works

Clustering algorithms, with a particular emphasis on k-means, have garnered significant attention and application in various fields. In this section, we explore several notable recent studies in the realm of clustering methods and related classification approaches.

Traditional clustering methods often assign equal weight to all features in high-dimensional data, making them sensitive to noise and irrelevant variables. They also struggle with uncertainty, fuzzy boundaries, overlapping clusters, and outliers. To address these issues, recent studies have introduced feature weighting and adaptive mechanisms [21]. This paper proposes an adaptive K-means method that dynamically determines the number of clusters based on data characteristics, enabling effective under-sampling for class imbalance [22]. While some methods enhance robustness to outliers or perform feature selection [23], few can address both simultaneously. Improved K-means via adaptive guided differential evolution (AGDE-KM), optimizing initial centers for better performance [24]. Other approaches use perceptions to build decision boundaries, reducing the need for frequent distance calculations [25].

Recently Research pointed out in their latest paper that the reason for the poor performance of the K-means algorithm is that the algorithm has difficulty in discovering the size and density of clusters. To solve this problem, they proposed a new multi-view K-means clustering method. Using fuzzy K-means, the new approach learns a bipartite connection probability matrix for each view and constructs a unified structured connection probability matrix that aligns closely with these view-based matrices [26]. It is as artificial intelligence continues to empower various fields of social development. Believe that the clustering of large-scale data sets has become important, but its performance still needs to be improved due to factors such as existing technologies. This study mainly studies the linear relationship between the algorithm's computational time, memory size overhead, and the number of samples [27].

Some others believe that to solve the problem clustering requires manually setting the k value. Proposed a clustering algorithm that automatically finds the k value [28]. The algorithm combines the following four algorithms: 1. Noise algorithm, 2. Genetic algorithm (GA), 3. Ant colony optimization (ACO), 4. Adaptive fuzzy system (AFS). In their paper, compared the performance of three clustering algorithms, namely: 1. Kernel Fuzzy C-Means (KFCM), 2. K-Means (KM), 3. Fuzzy C-Means (FCM). The experimental results show that the KFCM algorithm has a significant improvement in noise enhancement and recognition of speech signals [29]. The application of the k-means algorithm in financial fraud detection, can effectively identify abnormal patterns and behaviors and is safer than traditional detection methods [30].

There are more research shows that to solve decreasing performance problems for classification Models caused by a class imbalance in data since the k-means needs to preset a k value to determine the number of clusters [31].

Reviewed variants of the k-means and identified new challenges emerging in the big data era. Their research highlighted that the predominant focus of the classification Model lies in addressing algorithm initialization problems [4]. Highlighted the issue of slow convergence in clustering performance attributed to the utilization of random seeds as initial centroids in clustering. They introduced a remedy by employing fixed centroids as the initial clustering centers, termed FC-Means [32].

And finally, Proposed a segmentation technique to solve the overlapping problem of clustering. A centroid is placed at the center of the overlapping area as a new cluster, and then the data of this cluster is segmented, and finally, the neural network algorithm is integrated for classification. This is an effective solution for some data sets that are difficult to classify effectively [33].

However, most existing studies have shown that they seek better classification performance by integrating other algorithms, but this will increase the space complexity and time complexity. In the end, the model becomes very bloated and takes longer to calculate. In summary, we summarize the most effective classification algorithms currently as follows: Partition clustering [34, 35], Hierarchical clustering [36], Density clustering [37].

These studies have promoted the continuous updating and improvement of the K-means clustering algorithm. The current research mainly focuses on the problems of unclear clustering boundaries, optimal parameter selection, high-dimensional data, etc. Unlike previous studies, we focus more on optimizing the algorithm to improve its performance. This paper proposes a new algorithm to solve the problem of f outlier data classification. It is shown in Figure 1. Unlike the model method of the hybrid algorithm, our new solution is easier to understand and implement. In this study, we propose to add a dimension variable (weight) to improve the classification model. The newly added variable is used to change the similarity of the noise, thereby improving the accuracy of the classification model. The variable is an initialization parameter that must be defined before the model is run. In the following sections, we will outline the parameter setting of the new algorithm and evaluate model classification performance.

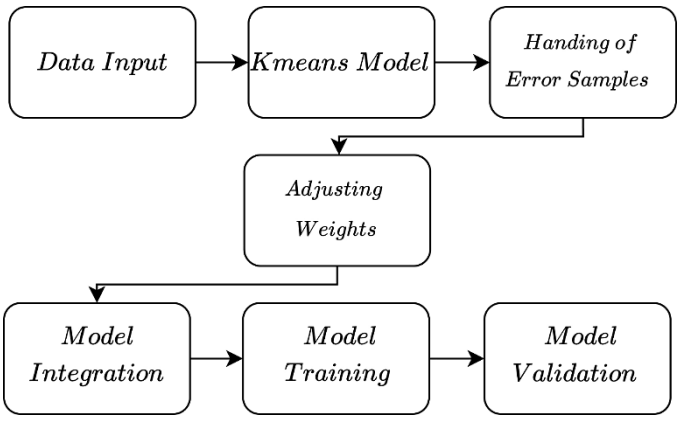

Figure 1. Research workflow of the proposed model

Contribution of this study:

1. Adaptive Weight Mechanism

A distance measurement method for dynamically adjusting the weight of outlier points is proposed. Through the exponential decay formula (Equation 2), the progressive weight reduction of outlier points is achieved, which solves the problem of traditional K-means being sensitive outliers and noise.

2. Local Dimension Compression

Innovatively "locally reduce the dimension" of outlier points through weight decay, only disturbing the feature space of the outlier area and retaining the topological structure of normal data.

3. Research Methodology

This section explains the structure of our proposed model, outlier handling, model integration, and Parameter Setting and Performance Validation.

3.1. Proposed Hybrid Model Structure

The traditional K-means algorithm is enhanced by introducing an adaptive weight mechanism to handle noise and outliers. The key steps are as follows (see Figure 2):

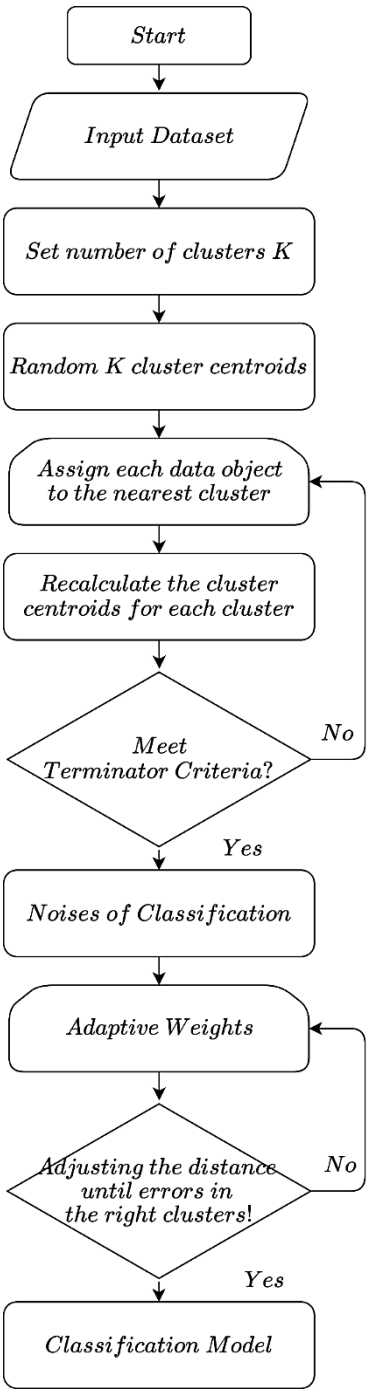

Figure 2. Workflow for the proposed model

1. Initialization:

- Randomly select K initial cluster centroids.
- Initialize weights w_i=1 for all dimensions.

2. Distance Calculation:

The modified Euclidean distance between a data point p and centroid q is defined as:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2 w_i} \quad (1)$$

3. Cluster Assignment:

- Assign each data point to the nearest cluster based on the weighted distance.

4. Centroid Update:

- Recompute centroids as the mean of all points in each cluster.

5. Outlier Handling:

- Identify outlier points as those misclassified or with distances exceeding a threshold θ.

Adjust weights for outlier points iteratively:

$$w_i^{(t+1)} = a \cdot w_i^{(t)} \quad (2)$$

where α is the learning rate, which controls the speed of weight decay.

6. Termination:

- Repeat until centroids stabilize or a maximum number of iterations is reached.

3.2. How the Adaptive Weight Mechanism Works

In this section, we detail our proposed outlier dimensionality reduction technique aimed at resolving outlier challenges in classification problems. The new variable is used to reclassify the outlier. Our approach outlines this approach and describes how to implement it, as shown in Figure 3.

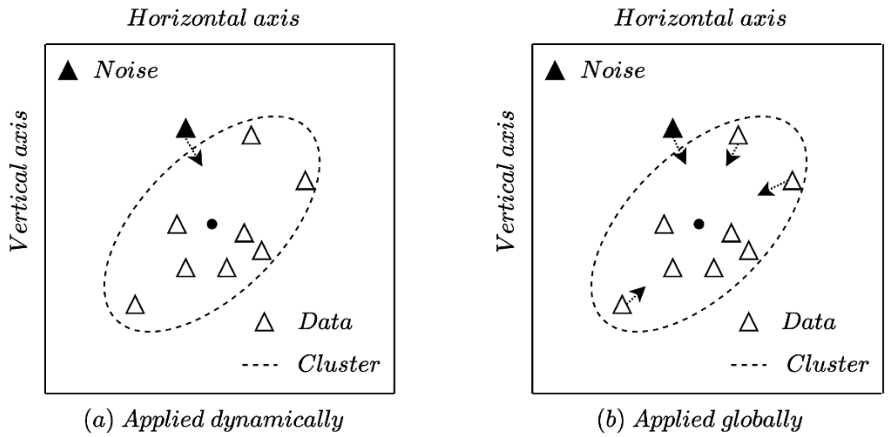

Figure 3. How the adaptive weight mechanism works

The adaptive weight mechanism adjusts the distance of outlier points so that they can be correctly classified. The key points include:

1. Dynamic weight adjustment (Figure 3-a):

- Weight adjustment is only performed on outlier points to avoid affecting the clustering structure of normal data points.
- Global weight adjustment (Figure 3-b) will cause the distance of all data points to change synchronously, and it is impossible to optimize the classification of outlier points in a targeted manner.

2. Dimension reduction effect of outlier points (Table 1):

- Through weight adjustment, the distance of outlier points in multidimensional space is "compressed", which is equivalent to local dimensionality reduction.
- Compared with feature space transformation, weighted methods act more directly on outlier points and retain the stability of normal data.

3. Outlier point judgment criteria:

If the current classification of a data point does not match its true label, it is judged as an outlierpoint. As shown in Figure 3-a, the dynamic weight adjustment targets outlier points only, while Table 1 contrasts this approach with feature space transformation methods.

Table 1. Transformation of feature space or a Weighting of features

Dimensions	Feature space transformation	Weighting of features
Origin Determinism	Preset	Iterative Update
Generation Direction	Center → Outward Dimension	Data → Inward Center
Dynamic	Static Structure	Dynamic Optimization
Dimensionality Reduction of outlier point	Reduce feature degradation noise points	Weights reduce data outlier points

3.3. Parameter Setting and Performance Validation

This section details the general parameter settings of the classification model and the formulas for evaluating model performance. To evaluate the performance of the proposed algorithm, the experiments have been conducted using Performance measurement functions. The parameter settings of the model are shown in Table 2.

$$Accuracy = \left(\frac{TN+TP}{TP+FP+TN+FN}\right) \times 100 \quad (3)$$

$$Precison = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1-score = 2 \times \left(\frac{Precision \times Recall}{Precision+Recall}\right) \quad (6)$$

$$\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N}} \quad (7)$$

$$MD = \frac{1}{N}\sum_{i=1}^{n}|x_i - \bar{x}| \quad (8)$$

The effectiveness of all tested models was assessed using key metrics, including accuracy, precision, recall, and F1-score [38, 39]. TP represents the accurate identification of anomaly instances, TN denotes the correct detection of normal instances, FP indicates the misclassification of anomalies, and FN reflects the failure to identify normal instances [40, 41].

1. New Parameters:

- The improved version adds weights w_i , learning rate α, and noise threshold θ to optimize outliers handling.
- Traditional K-means lacks weight mechanisms and is sensitive to noise and outliers.

2. Compatibility:

- The improved version retains traditional parameters for seamless integration.

3. Experimental Setup:

- Learning rate α=0.1, noise threshold θ dynamically calculated.
- Weight adjustment frequency: Updated per iteration for outliers' points.

The following is a table of K-means algorithm parameter configurations, covering the key parameters of traditional K-means and the adaptive weighted improved version proposed in the paper, with a comparison explanation as Table 2.

Table 2. General Parameters Settings of Classification Model

Parameter	Traditional K-means	Adaptive Weighted means (Improved)	Description
Number of Clusters (K)	Predefined	Same as left	Determines the final number of clusters, typically selected empirically or via evaluation metrics.
Initial Centroids	Random	Same as left	The improved algorithm retains traditional initialization to avoid added complexity.
Max Iterations	Default: 300	Same as left	Prevents infinite iteration, usually used with a convergence threshold.
Convergence Tolerance (tol)	Default: 1e-4	Same as left	Stops iteration if centroid movement is smaller than this value.
Distance Metric	Euclidean (default)	Weighted Euclidean Distance	The improved algorithm adjusts outlier point distances via weights.
Weight Initialization	N/A	Initial value: 1.0	All dimensions start with weight 1; only outlier points are dynamically adjusted.
Learning Rate	N/A	Default: 0.1 (adjustable)	Controls the decay speed of outlier point weights
Noise Threshold	N/A	1.5 × average intra-cluster distance	Distance threshold to identify outlier points; triggers weight adjustment if exceeded.
Adjustment Frequency	N/A	Update weights every 10 iterations	Avoids excessive adjustments that could destabilize results.

The improved version enhances robustness through four new parameters (weight, learning rate, outlier threshold, adjustment frequency). The remaining parameters are consistent with traditional K-means, balancing performance and ease of use. In practical applications, α and θ need to be adjusted according to data characteristics.

3.4. Specific Public Datasets Used in This Study

This study used a specific public dataset from UCI Machine Learning Repository [42]. To verify its generality, we tried to use a more diverse or noisier dataset. The following shows the details of the dataset used in the experiment. Figure 4 shows the dataset scatter plot. Tables 3 and 4 show the detailed properties of the dataset.

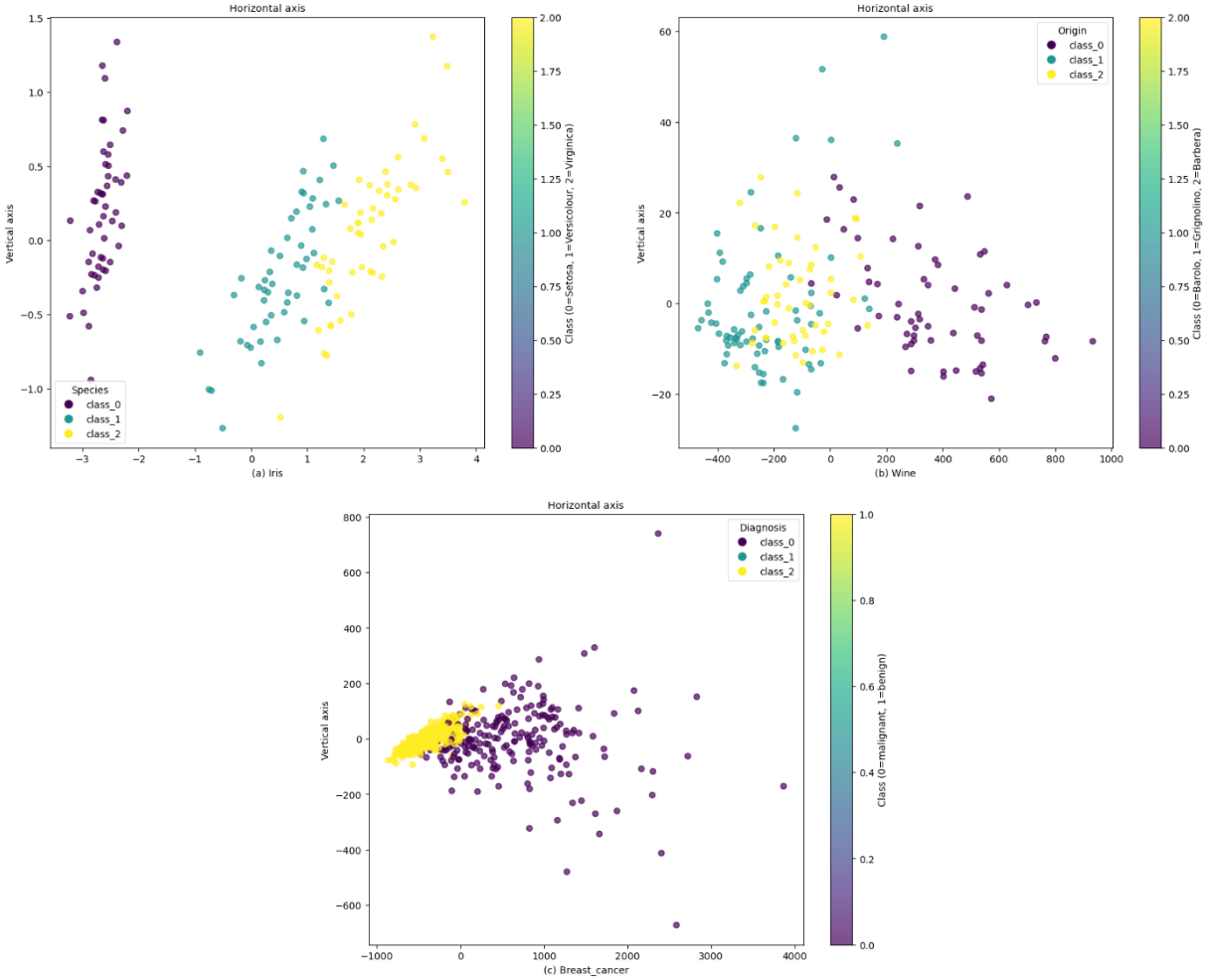

Figure 4. Scatter plots of datasets used in the experiments

Table 3. Datasets details

Dataset	Attributes	Class	Numbers	Samples	Tasks	Subject	Mission
Iris	4	3	150	[50, 50, 50]	Classicization	Biology	No
Wine	13	3	178	[59, 71, 48]	Classicization	Physics & Chemistry	No
WBCD	30	2	569	[212, 357]	Classicization	Health and Medicine	No

Table 4. Datasets characteristics

Dataset	Iris	Wine	Breast Cancer (Diagnostic)
Feature Type	Continuous	Continuous	Continuous
Noise level	Low	Medium	High
Missing Values	None	None	None
Feature mean range	Length: 5.84 ± 0.83 Width: 1.20 ± 0.76	Alcohol: 13.0 ± 0.8 Flavonoids: 2.03 ± 1.07	Radius Mean: 14.13 ± 3.52 Texture Mean: 19.29 ± 4.30
Category distribution	Balanced (50 per class)	Slightly Imbalanced [59, 71, 48]	Imbalanced (212 Malignant, 357 Benign)
Main Challenges	Linear Separability	High Feature Correlation	Dimensionality + Class Imbalance

3.5. Evaluate Model Performance and Stability

k-fold cross-validation is a commonly used model evaluation method, mainly used to improve the evaluation accuracy of model generalization ability. 5-fold cross-validation process shown in Figure 5.

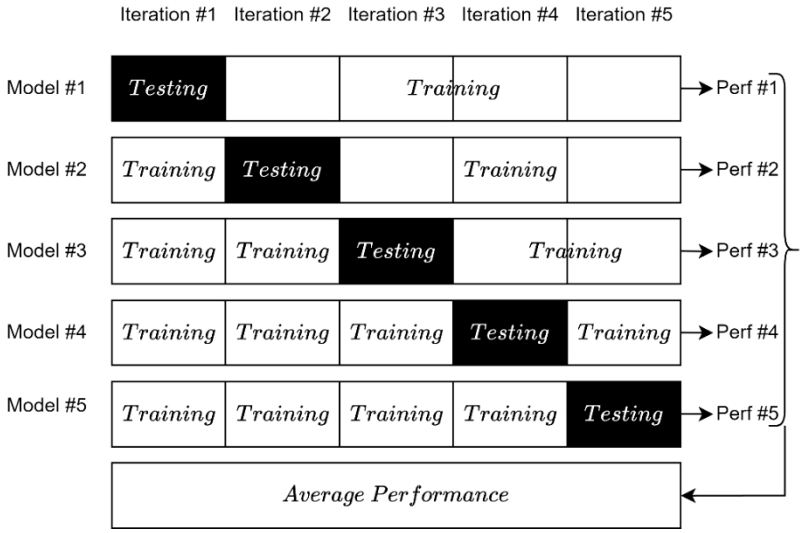

Figure 5. K-fold cross-validation

To evaluate the classification model, the k-fold cross-validation technique was employed [43, 44]. The dataset is split into 5 subsets, where each subset is used once as the test set while the others form the training set. This procedure is repeated k times, and the final performance is derived from the average results (Equation 9). Figure 6 demonstrates the k-fold cross-validation process.

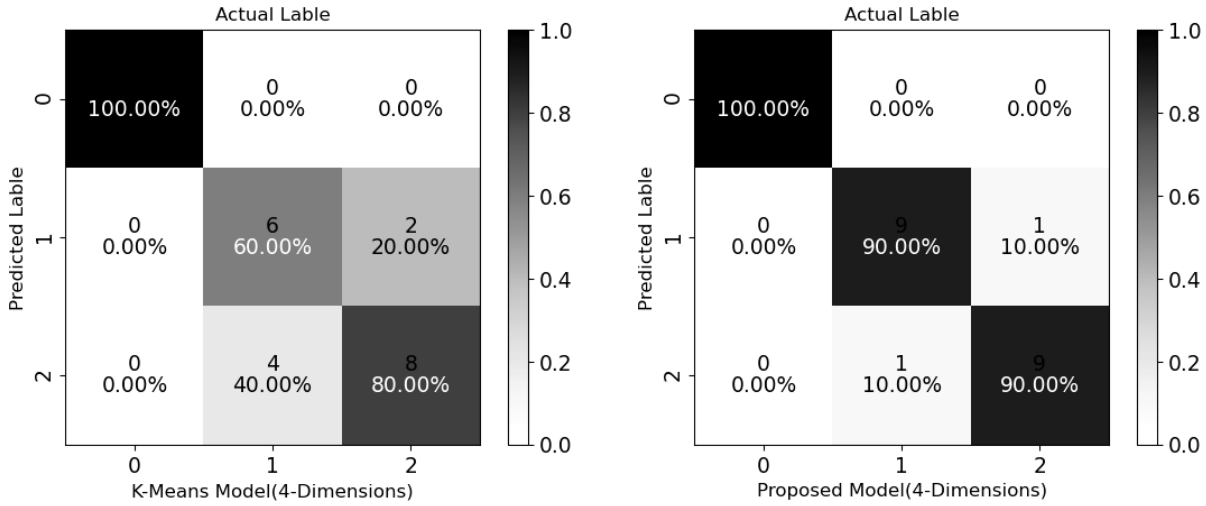

Figure 6. Confusion matrix showing our model accuracy in comparison with K-Means model (4-Dimensional Iris dataset)

The k-fold validation error (k=5) is calculated as:

$$E = \frac{1}{N}\sum_{i=1}^{N} M_i \tag{9}$$

where N is the number of cross-validation folds, $M_i$is Performance metrics for the i-fold cross validation, and E is The average of all fold evaluation indicators is used as the final performance of the model.

The computer hardware used in this experiment is Windows 10 Education, version 22H2, Intel(R) Core (TM) (i7-6700 CPU) (3.40GHz,3.41 GHz), (64-bit) computer operating system, (x64-based processor), and memory is 16.0 GB. The software used is version 3.9.12 and version 4.13.0 in Python and Anaconda. The details of the computer performance are shown in Table 5.

Table 5. The computer performance in the experimental environment

Operating system	Central Processing Unit	Processor	Random-Access Memory	Python	Anaconda
Windows 10 Education, version 22H2,	Intel(R) Core (TM) (i7-6700 CPU) (3.40GHz, 3.41 GHz)	(64-bit) Operating system, (x64-based) processor	Random-Access Memory (16.0 GB)	Version 3.9.12	Version 4.13.0

4. Experiment Results and Discussion

This section introduces the performance results of the models. The detailed performance comparison of our proposed model are shows by Figures 7 and 8.

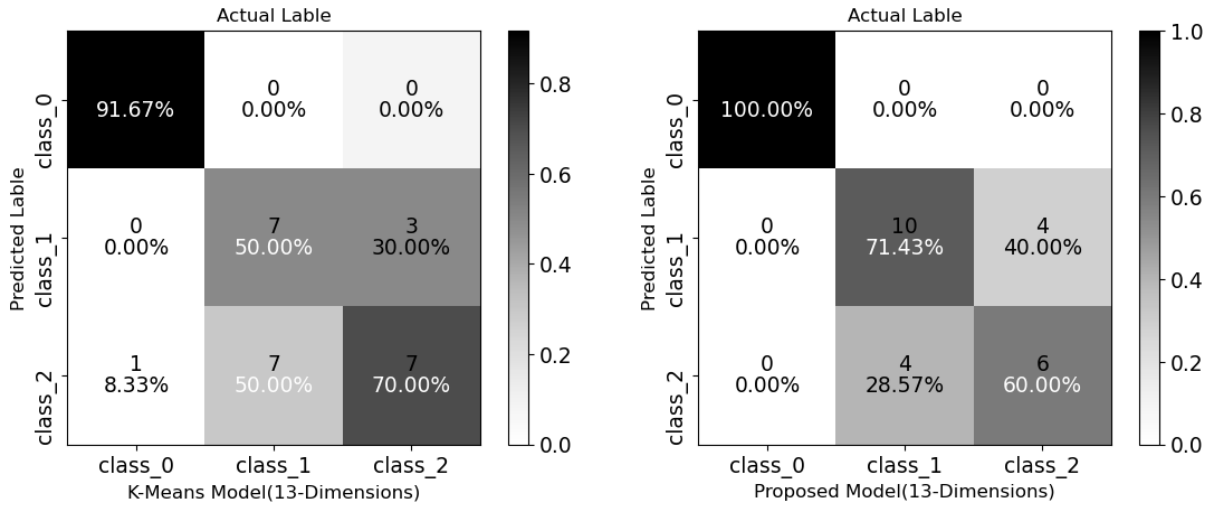

Figure 7. Confusion matrix showing our model accuracy in comparison with K-Means model (13-Dimensional Wine dataset)

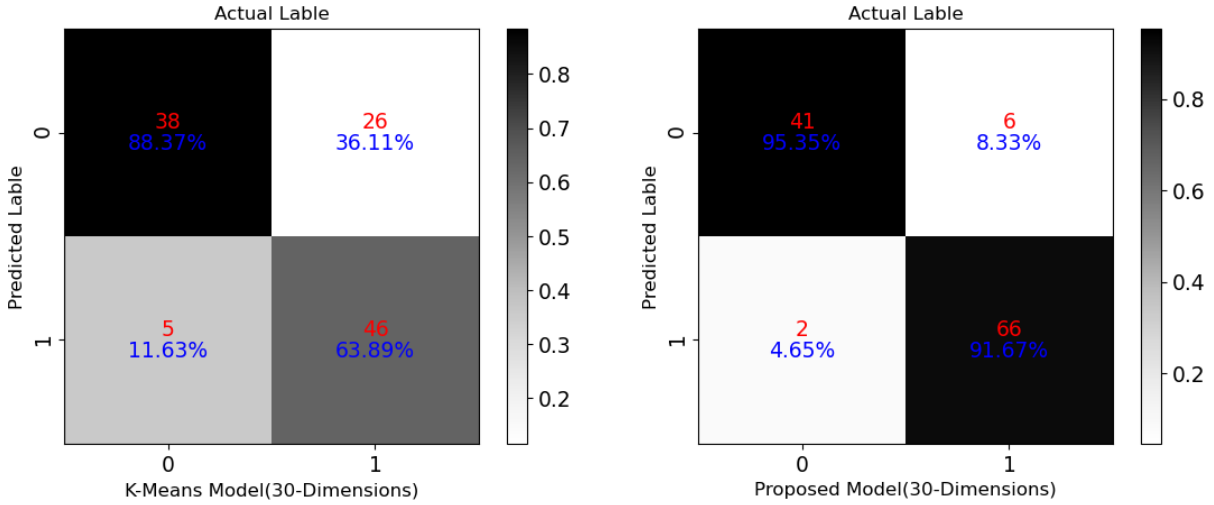

Figure 8. Confusion matrix showing our model accuracy in comparison with K-Means model (30 Dimensional WBCD dataset)

We compare the TDABC variant [45] as a baseline method for clustering stability, as shown in Tables 6 to 8. We also show the running time required for the models to achieve the same CPU performance in Tables 9 to 11.

Table 6. Performance comparison on Iris dataset

Model(iris)	Accuracy	Precision	Recall	F1-Score	S. D	Mean
K-MEANS	0.8666	0.866	0.866	0.866	0.8808	0.0584
TDABC- A [45]	0.9610	0.960	0.927	0.943	N/A	N/A
TDABC-M [45]	0.9200	0.917	0.859	0.883	N/A	N/A
TDABC-R [45]	0.9360	0.934	0.885	0.906	N/A	N/A
wk-NN [45]	0.9770	0.976	0.957	0.966	N/A	N/A
k-NN [45]	0.9800	0.979	0.962	0.970	N/A	N/A
PROPOSED	0.9413	0.934	0.936	0.931	0.8760	0.0517

Table 7. Performance comparison on Wine data

Model(wine)	Accuracy	Precision	Recall	F1-Score	S. D	Mean
K-MEANS	0.6666	0.677	0.662	0.667	0.7059	0.0998
TDABC- A [45]	0.7690	0.765	0.622	0.683	N/A	N/A
TDABC-M [45]	0.7670	0.763	0.619	0.680	N/A	N/A
TDABC-R [45]	0.7680	0.764	0.621	0.684	N/A	N/A
wk-NN [45]	0.7390	0.762	0.590	0.648	N/A	N/A
k-NN [45]	0.7080	0.761	0.547	0.608	N/A	N/A
PROPOSED	0.8055	0.816	0.810	0.811	0.6890	0.0817

Table 8. Performance comparison on WBCD data

Model (WBCD)	Accuracy	Precision	Recall	F1-Score	S. D	Mean
K-MEANS	0.74	0.72	0.63	0.69	0.7662	0.0406
TDABC- A [45]	0.91	0.91	0.90	0.91	N/A	N/A
TDABC-M [45]	0.92	0.92	0.91	0.91	N/A	N/A
TDABC-R [45]	0.92	0.92	0.91	0.91	N/A	N/A
wk-NN [45]	0.93	0.92	0.93	0.93	N/A	N/A
k-NN [45]	0.93	0.92	0.93	0.93	N/A	N/A
PROPOSED	0.93	0.92	0.93	0.81	0.8137	0.0356

Figures 9 to 11show the classification accuracy comparison of different algorithms on three benchmark datasets (Iris, Wine, and Wisconsin Breast Cancer Diagnosis (WBCD)). Figure 9 shows that all methods perform nearly perfectly on the Iris dataset (0.92-0.98), with the proposed method achieving an accuracy of 0.94. Figure 10: shows the consistent results on the Wine dataset, where the proposed method remains competitive (0.81) compared to other algorithms (range: 0.71-0.77). Figure 11: shows that the proposed method achieves excellent performance (accuracy 0.93) in breast cancer detection compared to the baseline methods.

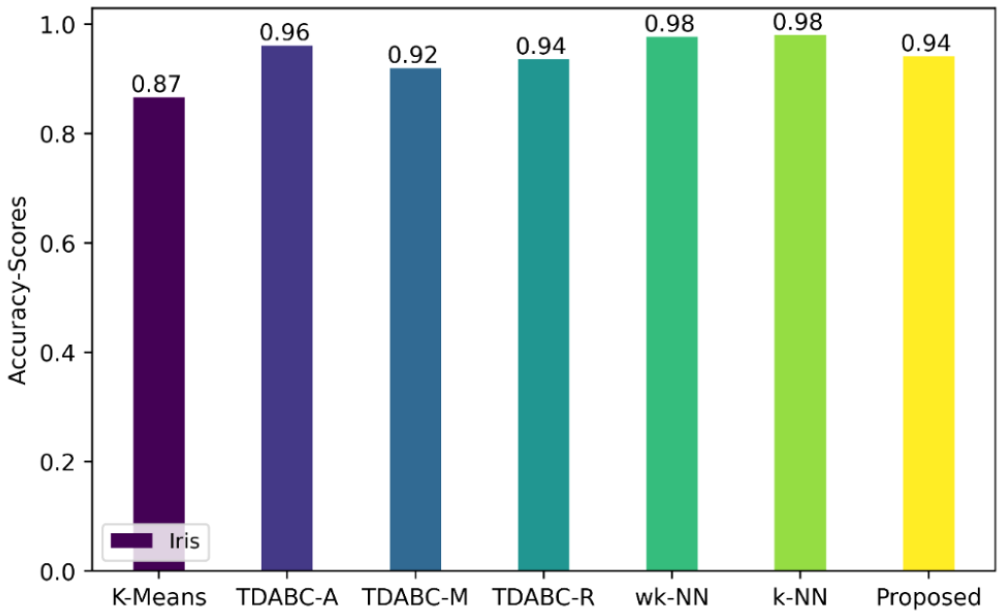

Figure 9. The bar chart showing the model accuracy in comparison with other classification models (4-Dimensional Iris dataset)

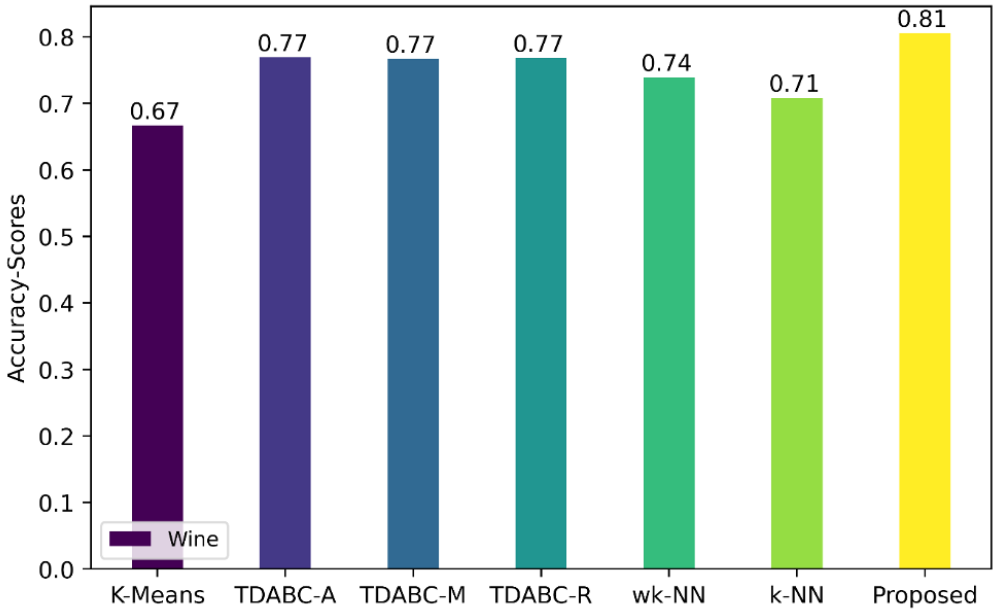

Figure 10. The bar chart showing the model accuracy in comparison with other classification models (13-Dimensional Wine dataset)

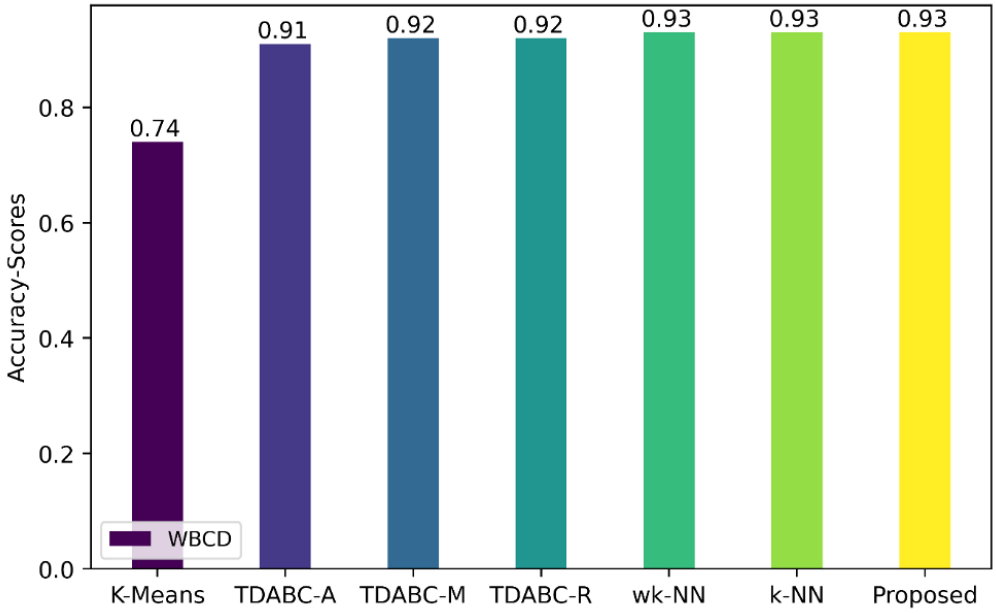

Figure 11. The bar chart showing the model accuracy in comparison with other classification models (30 Dimensional WBCD dataset)

In summary, these results demonstrate the robustness of the proposed approach across a variety of classification tasks, especially in complex medical diagnosis (WBCD), while maintaining competitive performance on simpler datasets. The ranking patterns of our proposed models show systematic algorithmic improvements on all three datasets (4D, 13D, 30D).

Table 6: Intuitively shows the classification results of our model and other similar models in the same dataset (Iris). In order to understand the model performance in detail, we compared the four evaluation indicators of Accuracy, Precision, Recall and F1-Score. The prediction accuracy of our model is 0.94, while the other models K-MEANS, TDABC-A, TDABC-M, TDABC-R, wk-NN, k-NN are 0.86, 0.96, 0.92, 0.93, 0.97, 0.98 respectively. Our model performance ranks third, only behind wk-NN and k-NN.

Table 7: Intuitively shows the classification results of our model and other similar models in the same dataset (Wine). In order to understand the model performance in detail, we compared the four evaluation indicators of Accuracy, Precision, Recall and F1-Score. The prediction accuracy of our model is 0.80, while the other models K-MEANS, TDABC-A, TDABC-M, TDABC-R, wk-NN, k-NN are 0.66, 0.76, 0.76, 0.76, 0.73, 0.70 respectively. Our model has the best performance.

Table 8: Intuitively shows the classification results of our model and other similar models in the same dataset (WBCD). In order to understand the model performance in detail, we compared the four evaluation indicators of Accuracy, Precision, Recall and F1-Score. The prediction accuracy of our model is: 0.93, while the other models K-MEANS, TDABC-A, TDABC-M, TDABC-R, wk-NN, k-NN are: 0.74, 0.91, 0.92, 0.92, 0.93, 0.93 respectively. The performance of our model is the best compared with that of wk-NN and k-NN models.

Figures 6 to 8: Visually show the distribution of model predictions (our model and K-means). The evaluation relies on four standard metrics: true positives, true negatives, false positives, and false negatives. The confusion matrix is based on the comparison of model predictions with actual results.

Tables 9 to 11 show the CPU training time of the models. Among them, (Iris) has an average model training time of 1.44E+01, a minimum training time of 1.38E+01, a maximum training time of 1.57E+01, and a total training time of 7.18E+01. (Wine) has an average model training time of 1.81E+01, a minimum training time of 1.66E+01, a maximum training time of 2.34E+01, and a total training time of 9.18E+01. (WBCD) The average model training time is 2.14E+01, the shortest training time is 1.84E+01, the longest training time is 2.57E+01, and the total training time is 13.18E+01

Table 9. CPU time required to run the Iris dataset

CPU (Iris)	Training Cost	Average -training	Min-Training	Max-Training	Total
Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz 3.41 GHz	Low-Cost	1.44E+01	1.38E+01	1.57E+01	7.18E+01

Table 10. CPU time required to run the Wine dataset

CPU (Wine)	Training Cost	Average -training	Min-Training	Max-Training	Total
Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz 3.41 GHz	Low-Cost	1.81E+01	1.66E+01	2.34E+01	9.18E+01

Table 11. CPU time required to run the WBCD dataset

CPU (WBCD)	Training Cost	Average -training	Min-Training	Max-Training	Total
Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz 3.41 GHz	Low-Cost	2.14E+01	1.84E+01	2.57E+01	13.18E+01

5. Conclusion

This study improves the K-means classification algorithm. The improved algorithm can spatially fold the outlier points in the cluster, shorten their distance from the straight-line points in the cluster, effectively reduce the dimension of the outlier points, and make them enter the correct cluster.

The experimental data comparison analysis is divided into indicators such as accuracy, precision, recall, and F1 index. A comprehensive summary of accuracy, precision, recall, and F1 index is made, and the performance of the model is evaluated in detail. In addition, in order to evaluate the proposed model, we conducted experiments using 4-, 13-, and 30-dimensional data sets. For the Iris dataset, when compared with the original K-means classification model, the performance of the improved model is improved by an average of 7.47%. Compared with other classification models, our model performs better than K-MEANS, TDABC-A, TDABC-M, and TDABC-R on the Iris data set and is close to the performance of wk-NN and k-NN. For the Wine dataset, when compared with the original K-means classification model, the performance of the improved model is improved by an average of 13.89%. Compared with other classification models, our model outperforms TDABC-A, TDABC-M, TDABC-R, wk-NN, and k-NN on the wine dataset. For the WBCD dataset, when compared with the original K-means classification model, the performance of the improved model is improved by 19% on average. Compared with other classification models, our model outperforms TDABC-A and TDABC-M on the Wine dataset and performs comparably to TDABC-R, wk-NN, and k-NN models.

In summary, we evaluated the performance of other models on public datasets. The experimental results demonstrate the effectiveness of our proposed method. In summary, we evaluated the performance of other models on public datasets. The experimental results demonstrate the effectiveness of our proposed method.

6. Declarations

6.1. Author Contributions

Conceptualization, C.C. and A.T.; methodology, C.C. and A.T.; software, C.C.; validation, C.C.; formal analysis, C.C. and A.T.; investigation, C.C. and A.T.; resources, C.C. and A.T.; data curation, C.C.; writing—original draft preparation, C.C.; writing—review and editing, C.C. and A.T.; supervision, A.T.; project administration, C.C.; funding acquisition, A.T. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available in the article.

6.3. Funding and Acknowledgments

This work was supported by King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

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

7. References

[1] Oyewole, G. J., & Thopil, G. A. (2023). Data clustering: application and trends. Artificial Intelligence Review, 56(7), 6439–6475. doi:10.1007/s10462-022-10325-y.

[2] Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3). doi:10.1007/s42979-021-00592-x.

[3] Han, J., Pei, J., & Tong, H. (2022). Data mining: concepts and techniques. Morgan kaufmann. Burlington, United States.

[4] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622, 178–210. doi:10.1016/j.ins.2022.11.139.

[5] Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P. V., Janssen, M., Jones, P., Kar, A. K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., … Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International Journal of Information Management, 57, 101994. doi:10.1016/j.ijinfomgt.2019.08.002.

[6] Hosseinzadeh, M., Koohpayehzadeh, J., Bali, A. O., Asghari, P., Souri, A., Mazaherinezhad, A., Bohlouli, M., & Rawassizadeh, R. (2021). A diagnostic prediction model for chronic kidney disease in internet of things platform. Multimedia Tools and Applications, 80(11), 16933–16950. doi:10.1007/s11042-020-09049-4.

[7] Wallace, W., Chan, C., Chidambaram, S., Hanna, L., Iqbal, F. M., Acharya, A., Normahani, P., Ashrafian, H., Markar, S. R., Sounderajah, V., & Darzi, A. (2022). The diagnostic and triage accuracy of digital and online symptom checker tools: a systematic review. NPJ Digital Medicine, 5(1), 118. doi:10.1038/s41746-022-00667-w.

[8] Hamzehi, M., & Hosseini, S. (2022). Business intelligence using machine learning algorithms. Multimedia Tools and Applications, 81(23), 33233–33251. doi:10.1007/s11042-022-13132-3.

[9] Djabalul Lael, T. A., & Pramudito, D. A. (2023). Use of Data Mining for The Analysis of Consumer Purchase Patterns with The Fpgrowth Algorithm on Motor Spare Part Sales Transactions Data. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 4(2), 128–136. doi:10.34306/itsdi.v4i2.582.

[10] Santoso, M. H. (2021). Application of Association Rule Method Using Apriori Algorithm to Find Sales Patterns Case Study of Indomaret Tanjung Anom. Brilliance: Research of Artificial Intelligence, 1(2), 54–66. doi:10.47709/brilliance.v1i2.1228.

[11] Selim, S. Z., & Ismail, M. A. (1984). K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(1), 81–87. doi:10.1109/tpami.1984.4767478.

[12] Veenman, C. J., Reinders, M. J. T., & Backer, E. (2002). A maximum variance cluster algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(9), 1273–1280. doi:10.1109/TPAMI.2002.1033218.

[13] Xu, Z., Shen, D., Nie, T., Kou, Y., Yin, N., & Han, X. (2021). A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data. Information Sciences, 572, 574–589. doi:10.1016/j.ins.2021.02.056.

[14] Sieranoja, S., & Fränti, P. (2022). Adapting k-means for graph clustering. Knowledge and Information Systems, 64(1), 115–142. doi:10.1007/s10115-021-01623-y.

[15] Gupta, A. K., Seal, A., Khanna, P., Krejcar, O., & Yazidi, A. (2021). AWkS: adaptive, weighted k-means-based superpixels for improved saliency detection. Pattern Analysis and Applications, 24(2), 625–639. doi:10.1007/s10044-020-00925-1.

[16] Zhang, H., Li, J., Zhang, J., & Dong, Y. (2024). Speeding up k-means clustering in high dimensions by pruning unnecessary distance computations. Knowledge-Based Systems, 284, 111262. doi:10.1016/j.knosys.2023.111262.

[17] Yang, X., Zhao, W., Xu, Y., Wang, C. D., Li, B., & Nie, F. (2024). Sparse K-means clustering algorithm with anchor graph regularization. Information Sciences, 667, 120504. doi:10.1016/j.ins.2024.120504.

[18] Wan, B., Huang, W., Pierre, B., Cheng, Y., & Zhou, S. (2024). K-Means algorithm based on multi-feature-induced order. Granular Computing, 9(2), 45. doi:10.1007/s41066-024-00470-w.

[19] Kouadio, K. L., Liu, J., Liu, R., Wang, Y., & Liu, W. (2024). K-Means Featurizer: A booster for intricate datasets. Earth Science Informatics, 17(2), 1203–1228. doi:10.1007/s12145-024-01236-3.

[20] Qtaish, A., Braik, M., Albashish, D., Alshammari, M. T., Alreshidi, A., & Alreshidi, E. J. (2024). Optimization of K-means clustering method using hybrid capuchin search algorithm. Journal of Supercomputing, 80(2), 1728–1787. doi:10.1007/s11227-023-05540-5.

[21] Liu, Z., Qiu, H., & Letchmunan, S. (2024). Self-adaptive attribute weighted neutrosophic c-means clustering for biomedical applications. Alexandria Engineering Journal, 96, 42–57. doi:10.1016/j.aej.2024.03.092.

[22] Zhou, Q., & Sun, B. (2024). Adaptive K-means clustering based under-sampling methods to solve the class imbalance problem. Data and Information Management, 8(3), 100064. doi:10.1016/j.dim.2023.100064.

[23] Li, H., Sugasawa, S., & Katayama, S. (2024). Adaptively Robust and Sparse K-means Clustering. arXiv Preprint, arXiv:2407.06945. doi:10.48550/arXiv.2407.06945.

[24] An, L., Sun X. H., & Wang, Y. (2024). K-Means Clustering Algorithm Based on Improved Differential Evolution. Information Dynamics and Applications, 3(3), 200-210. doi:10.56578/ida030305.

[25] Long, J., & Liu, L. (2025). K*-Means: An Efficient Clustering Algorithm with Adaptive Decision Boundaries. International Journal of Parallel Programming, 53(1), 1–27. doi:10.1007/s10766-024-00779-8.

[26] Zhang, Z., Chen, X., Wang, C., Wang, R., Song, W., & Nie, F. (2025). Structured multi-view k-means clustering. Pattern Recognition, 160, 111113. doi:10.1016/j.patcog.2024.111113.

[27] Pei, S., Sun, Y., Nie, F., Jiang, X., & Zheng, Z. (2025). Adaptive Graph K-Means. Pattern Recognition, 161, 111226. doi:10.1016/j.patcog.2024.111226.

[28] Ran, X., Suyaroj, N., Tepsan, W., Ma, J., Zhou, X., & Deng, W. (2024). A hybrid genetic-fuzzy ant colony optimization algorithm for automatic K-means clustering in urban global positioning system. Engineering Applications of Artificial Intelligence, 137, 109237. doi:10.1016/j.engappai.2024.109237.

[29] Abdullah, A. A., Ahmed, A. M., Rashid, T., Veisi, H., Rassul, Y. H., Hassan, B., ... & Shamsaldin, A. S. (2024). Advanced clustering techniques for speech signal enhancement: A review and metanalysis of fuzzy c-means, k-means, and kernel fuzzy c-means methods. arXiv preprint arXiv:2409.19448. doi:10.48550/arXiv.2409.19448.

[30] Huang, Z., Zheng, H., Li, C., & Che, C. (2024). Application of Machine Learning-Based K-means Clustering for Financial Fraud Detection. Academic Journal of Science and Technology, 10(1), 33–39. doi:10.54097/74414c90.

[31] Hassan, M. M., Eesa, A. S., Mohammed, A. J., & Arabo, W. K. (2021). Oversampling method based on gaussian distribution and K-means clustering. Computers, Materials and Continua, 69(1), 451–469. doi:10.32604/cmc.2021.018280.

[32] Ay, M., Özbakır, L., Kulluk, S., Gülmez, B., Öztürk, G., & Özer, S. (2023). FC-Kmeans: Fixed-centered K-means algorithm. Expert Systems with Applications, 211, 118656. doi:10.1016/j.eswa.2022.118656.

[33] Chenghu, C., & Thammano, A. (2024). A Novel Classification Model Based on Hybrid K-Means and Neural Network for Classification Problems. HighTech and Innovation Journal, 5(3), 716–729. doi:10.28991/HIJ-2024-05-03-012.

[34] Celebi, M. E. (Ed.). (2015). Partitional Clustering Algorithms. Springer, Cham, Switzerland. doi:10.1007/978-3-319-09259-1.

[35] MacQueen, J. (1967). Multivariate observations. Proceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability, Vol. 1, 281-297, University of California Press, Berkeley, United States.

[36] Murtagh, F., & Contreras, P. (2011). Algorithms for hierarchical clustering: an overview. WIREs Data Mining and Knowledge Discovery, 2(1), 86–97. doi:10.1002/widm.53.

[37] Kriegel, H., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. WIREs Data Mining and Knowledge Discovery, 1(3), 231–240. doi:10.1002/widm.30

[38] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061. doi:10.48550/arXiv.2010.16061.

[39] Naidu, G., Zuva, T., Sibanda, E.M. (2023). A Review of Evaluation Metrics in Machine Learning Algorithms. Artificial Intelligence Application in Networks and Systems. CSOC 2023. Lecture Notes in Networks and Systems, Vol 724. Springer, Cham, Switzerland. doi:10.1007/978-3-031-35314-7_2.

[40] Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. Advances in Artificial Intelligence, AI 2006, Lecture Notes in Computer Science, vol 4304, Springer, Berlin, Germany. doi:10.1007/11941439_114.

[41] Tharwat, A. (2020). Classification assessment methods. Applied Computing and Informatics, 17(1), 168–192. doi:10.1016/j.aci.2018.08.003.

[42] Aeberhard, S. & Forina, M. (1992). Wine [Dataset]. UCI Machine Learning Repository, *Noida, India*. doi:10.24432/C5PC7J.

[43] Wong, T.-T., & Yeh, P.-Y. (2019). Reliable Accuracy Estimates from k-Fold Cross Validation. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1586–1594. doi:10.1109/tkde.2019.2912815.

[44] Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. Statistics and Computing, 21(2), 137–146. doi:10.1007/s11222-009-9153-8.

[45] Kindelan, R., Frías, J., Cerda, M., & Hitschfeld, N. (2021). Classification based on topological data analysis. doi:10.48550/arXiv.2102.03709.