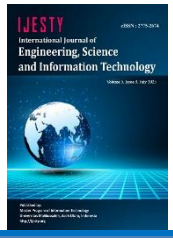
International Journal of Engineering, Science and Information Technology
Volume 5 No. 3 (2025) pp. 102-111
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.913
Research Paper, Short Communication, Review, Technical Paper



Hybrid Deep Fixed K-Means (HDF-KMeans)

Muhammad Khahfi Zuhanda[1*], Kelvin Leonardi Kohsasih[2], Pieter Octaviandy[2], Hartono[1], Dian Kurnia[3], Nurliana Tarigan[4], Manan Ginting[4], Manahan Hutagalung[4]

[1]Department of Informatics, Faculty of Engineering, Universitas Medan Area, Medan, Indonesia
[2]Department of Informatics, STMIK TIME, Medan, Indonesia
[3]Department of Oil Palm Agribusiness, Politeknik Teknologi Kimia Industri, Medan, Indonesia
[4]Department of Mechanical Engineering, Politeknik Teknologi Kimia Industri, Medan, Indonesia

**Corresponding author Email: khahfi@staff.uma.ac.id*

The manuscript was received on 21 December 2024, revised on 15 January 2025, and accepted on 10 April 2025, date of publication 23 May 2025

Abstract

K-Means is one of the most widely used clustering algorithms due to its simplicity, scalability, and computational efficiency. However, its practical application is often hindered by several well-known limitations, such as high sensitivity to initial centroid selection, inconsistency across different runs, and suboptimal performance when dealing with high-dimensional or non-linearly separable data. This study introduces a hybrid clustering algorithm named Hybrid Deep Fixed K-Means (HDF-KMeans) to address these issues. This approach combines the advantages of two state-of-the-art techniques: Deep K-Means++ and Fixed Centered K-Means. Deep K-Means++ leverages deep learning-based feature extraction to transform raw data into more meaningful representations while employing advanced centroid initialization to enhance clustering accuracy and adaptability to complex data structures. Complementarily, Centered K-Means improve the stability of clustering results by locking certain centroids based on domain knowledge or adaptive strategies, effectively reducing variability and convergence inconsistency. Integrating these two methods results in a robust hybrid model that delivers improved accuracy and consistency in clustering performance. The proposed HDF-KMeans algorithm is evaluated using five benchmark medical datasets: Breast Cancer, COVID-19, Diabetes, Heart Disease, and Thyroid. Performance is assessed using standard classification metrics: Accuracy, Precision, Recall, and F1-Score. The results show that HDF-KMeans outperforms traditional K-Means, K-Means++, and K-Means-SMOTE algorithms across all datasets, excelling in overall accuracy and F1 Score. While some trade-offs are observed in specific precision or recall metrics, the model maintains a solid balance, demonstrating reliability. This study highlights HDF-KMeans as a promising and effective solution for complex clustering tasks, particularly in high-stakes domains like healthcare and biomedical analysis.

Keywords: *K-Means, Initial Centroid Selection, Hybrid Deep Fixed K-Means, Deep K-Means++, Fixed Centered K-Means.*

1. Introduction

K-Means is a clustering algorithm that has long been a foundation in unsupervised learning [1], mainly due to its simplicity [2], fast convergence [3], and interpretability of its results [4]. This algorithm works by dividing the data into k clusters [5] based on the Euclidean distance [6]to the cluster center point (centroid) [7], which is updated iteratively. K-Means has been used in many real-life problems, such as routing problems [8], logistic distribution [9], e-commerce [10], the Travelling Salesman Problem [11], and Image Segmentation [12]. However, despite its widespread use, K-Means has several significant drawbacks. First, the clustering results are highly dependent on the initial random initialization of the cluster centers [13], which can result in suboptimal solutions (local minima) [14]. Second, K-Means cannot handle non-spherical cluster shapes or non-linear data distributions well [15]. Third, this algorithm is sensitive to outliers, and its scalability is limited when applied to high-dimensional or complex data[16]. In response to these limitations, various variants of K-Means have been developed to improve the stability and accuracy of the clustering results [17]. One modern approach is Deep K-Means++, which integrates deep learning-based representations before the clustering process. In this approach, the raw data is first processed through a deep neural network to produce a more meaningful and structured representation of latent features [18]. The main advantage of Deep K-Means++ is its ability to handle high-dimensional and non-linear data because the network structure allows the extraction of complex patterns that traditional K-Means cannot reach. In addition, this method adopts the principle of K-Means++ in intelligently selecting cluster center initialization, thereby reducing the risk of convergence to a local solution.

However, Deep K-Means++ is not free from challenges. High computational complexity [19], the need for large amounts of training data[20], and the potential for overfitting on unrepresentative data are the main obstacles to implementing this method [21]. In addition,

because the training process involves backpropagation and optimization of many parameters, the stability of the centroid position is sometimes not maintained in long iterations, causing inconsistent cluster results in several runs [22].

The fixed-centered K-means method is introduced to overcome the stability problem and strengthen control over the centroid position. In this approach, some or all cluster centers are locked at predetermined positions based on domain knowledge, pre-processing results, or other adaptive schemes [23]. By limiting the movement of cluster centers, this method can maintain the consistency of the results and reduce the variability between iterations. Another advantage of Fixed Centered K-Means is the computational time efficiency because the solution search space becomes more limited and focused. However, this approach is not free from limitations, such as its inability to adapt to dynamic data or data that experience distribution shifts and the possibility of suboptimal center positions if the determination is inaccurate.

Combining Deep K-Means++ and Centered K-Means has become a promising hybrid approach because each method can complement the other's shortcomings. Deep K-Means++ provides a robust framework for complex feature extraction and adjustment to non-linear data. At the same time, fixed-centered K-Means maintain the stability of the results by directing the clustering process to specific centers known to be significant. This combination enables the clustering process to be more precise, stable, and targeted—especially in domains such as medical image analysis, e-commerce customer segmentation, or document clustering in NLP, which require high accuracy and consistency of results.

Overall, integrating these two methods offers dual benefits: leveraging the advanced feature representation modeling capabilities of deep learning and controlling the position of the cluster center that can minimize the variability of the results. Thus, this combined approach not only improves the weaknesses of each method but also opens up opportunities for developing more adaptive, reliable, and applicable clustering systems in various modern data domains.

2. Method

2.1. Deep K-Means++

The pseudocode of Deep K-Means++ is as follows[18].

Algorithm 1 Deep K-Means++

Require: S: Received signal from LBL system
Require: M: Total number of reference beacons
Require: N: Number of source points
Require: cf: Damage factor (corruption level)
Require: $batch\_size$: Size of mini-batch
Require: $epoch$: Fine-tuning epochs
Require: $epoch_p$: Pre-training epochs
Require: β: Clustering coefficient (0 or 1)
Require: λ: Fine-tuning regularization parameter
Require: n_{dA}: SDA hidden layer configuration (e.g. [130, 50, 3])
Ensure: c_{new}: Final cluster centers

Initialize encoder weights W_e, decoder weights W_d, biases b_e, b_d
Normalize dataset $D_t = \frac{D}{\max(|D|)}$
for each layer in SDA configuration n_{dA} **do**
  for $i = 1$ to $epoch_p$ **do**
    Corrupt data: $\hat{D}_t \sim q_D(\hat{D}_t|D_t)$
    Encode: $H_e = \text{ReLU}(W_e\hat{D}_t + b_e)$
    Decode: $H_d = \text{ReLU}(W_dH_e + b_d)$
    Update parameters: minimize $J_{dA}(W_p) = \text{mean}\sum(D_t - H_d)^2$
  end for
  Stack outputs to next dA input
end for
Let D_a be final output from SDA
Initialize cluster centers c_{old} using K-Means++:
Randomly choose 1 data point as first center
while number of centers $< N$ **do**
  For each data point, compute distance to nearest center d_x
  Sample next center with probability $\propto d_x^2$
end while
for $j = 1$ to $epoch$ **do**
  Partition D_a into mini-batches D_M
  For each mini-batch:
    Assign data points to nearest centers in c_{old}
    Compute new centers c_M for mini-batch
    Minimize total loss:

$$J_{SdA}(W) = \beta \cdot \sum(c_M - D_M)^2 + \lambda \cdot J_{RdA}(W_r)$$

    Update cluster centers:

$$c_{new} = (1 - \eta)c_{old} + \eta D_M, \quad \eta = \frac{1}{100 + num}$$

end for
return c_{new} as final estimated source positions

The pseudocode describes the Deep K-Means++ clustering algorithm, which combines a stacked denoising autoencoder (SDA) for feature extraction with a refined K-Means++ clustering process. Initially, the SDA's encoder and decoder weights and biases are

initialized, and the input dataset is normalized. The SDA is trained layer by layer using a pre-training phase, where corrupted input data is encoded and decoded through ReLU activations, and parameters are updated by minimizing the reconstruction loss between the original and decoded data. After pre-training all layers, the final encoded output serves as a transformed dataset for clustering. The algorithm then initializes cluster centers using the K-Means++ method, selecting the first center randomly and subsequent centers with probability proportional to the squared distance from existing centers, promoting well-separated initial centroids. During the fine-tuning phase, the transformed data is divided into mini-batches; each batch is assigned to the nearest cluster centers, and new centers are computed. The total loss minimized consists of two terms: the clustering loss weighted by a clustering coefficient and a regularization term weighted by a fine-tuning parameter to ensure stability of the SDA weights. Cluster centers are updated incrementally using a learning rate that decreases over iterations, smoothing the adjustment of cluster positions. Ultimately, the algorithm returns the refined cluster centers representing the estimated source positions. This approach effectively integrates deep feature learning with improved centroid initialization and incremental clustering updates to enhance accuracy and robustness.

2.2. Fixed-Centered K-Means

The pseudocode of Fixed Centered K-Means is as follows[23].

Algorithm 2 Fixed Centered K-Means (FC-KMeans)

Require: Set of data points $P = \{p_1, p_2, \ldots, p_n\}$
Set of fixed centers $F = \{f_1, f_2, \ldots, f_m\}$
Total number of clusters k
Maximum iterations Phase-I: `max_iter_1`, Phase-II: `max_iter_2`
Ensure: Cluster assignments and final cluster centers
Phase I: Initialization using K-means++
Select k initial centers $\{\mu_1, \mu_2, \ldots, \mu_k\}$ using K-means++ method
repeat
    Assign each point $p \in P$ to the nearest center μ_i
    Update each center μ_i as the centroid of its cluster
until centers do not change or `max_iter_1` reached
Phase II: Fixed Center Clustering
Compute average distance from each μ_i to all fixed centers in F
Select $(k - m)$ centers from $\{\mu_1, \ldots, \mu_k\}$ farthest from F as $\{f_{m+1}, \ldots, f_k\}$
Define $F = F \cup \{f_{m+1}, \ldots, f_k\}$
repeat
    Clear all clusters $C_1, \ldots, C_k$
    for all $p \in P$ **do**
        Assign p to nearest center in F
    end for
    for $i = m + 1$ to k **do** ▷ Update only non-fixed centers
        $f_i \leftarrow$ centroid of assigned points in C_i
    end for
until non-fixed centers do not change or `max_iter_2` reached

The Fixed Centered K-Means (FC-KMeans) algorithm is a two-phase clustering approach incorporating fixed cluster centers alongside dynamically updated centers. In the first phase, the algorithm initializes cluster centers using the K-means++ method, carefully selecting initial centers to improve convergence. It then iteratively assigns each data point to the nearest center and recalculates the centers as the centroids of their respective clusters. This process continues until the centers stabilize or a predefined maximum number of phase one iterations is reached.

In the second phase, the algorithm integrates a set of fixed centers that remain constant throughout the process. It begins by calculating the average distance between each cluster center obtained in phase one and the fixed centers. Then, it selects the remaining cluster centers farthest from the fixed centers to complete the total number of clusters. The fixed and newly selected centers form the updated set of centers.

The algorithm repeatedly assigns each data point to the nearest center from the combined set of fixed and newly selected centers. Only the non-fixed centers are updated by recalculating their centroids based on their assigned points. This cycle continues until the non-fixed centers no longer change or the maximum number of phase two iterations is reached. Through this method, FC-KMeans ensures that specific important centers remain fixed while still adapting the other centers to better represent the data distribution.

2.3. Hybrid Deep Fixed K-Means

The pseudocode of Hybrid Deep is as follows.

Algorithm 3 Hybrid Deep Fixed K-Means

Input: Dataset $X = \{x_1, x_2, \ldots, x_n\}$, number of clusters k, fixed centers $F = \{f_1, \ldots, f_m\}$, pretrained SDA network
Output: Cluster assignments and final centers C
// **Stage 1: Deep Feature Extraction**
Normalize input data X to get $\tilde{X}$
Apply noise corruption to get $\hat{X}$
Encode $\hat{X}$ using Stacked Denoising Autoencoder (SDA):
  $H = \text{SDA.encode}(\hat{X})$
Extract deep feature space: $D_a \leftarrow H$
// **Stage 2: Initial Center Selection with K-Means++ (Non-fixed only)**
Initialize $C_{nf} = \{c_{m+1}, \ldots, c_k\}$ using K-Means++ on D_a (excluding fixed F)
$C \leftarrow F \cup C_{nf}$
// **Stage 3: Clustering with Fixed Centers**
repeat
  Assign each point $x_i \in D_a$ to nearest center in C
  For each non-fixed center $c_j \in C_{nf}$:
    Update $c_j \leftarrow \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
  Optional: Fine-tune SDA weights using mini-batch SGD with clustering loss
until convergence or max iteration reached
return Cluster assignments and final centers C

The pseudocode titled Hybrid Deep Fixed K-Means outlines an algorithm combining deep feature extraction with a clustering technique that incorporates fixed and learnable cluster centers. The algorithm inputs a dataset $X = \{x_1, x_2, \ldots, x_n\}$, a target number of clusters k, a set of fixed centers $F = \{f_1, \ldots, f_m\}$, and a trained Stacked Denoising Autoencoder (SDA) network. The output is the final cluster assignments and the complete set of cluster centers C.

The algorithm consists of three main stages. In Stage 1, deep feature extraction is performed. The input data X is first normalized to produce $\tilde{X}$, then corrupted by adding noise to generate $\hat{X}$. This corrupted input is passed through the encoder part of the SDA to obtain deep representations H, which are then stored in the feature space D_a. This transformation helps extract more robust and abstract features for clustering.

In Stage 2, the algorithm initializes the non-fixed cluster centers $C_{nf} = \{c_{m+1}, \ldots, c_k\}$ using the K-Means++ algorithm, which is applied to the deep feature space D_a, excluding the fixed centers. The complete set of centers C is then formed by merging the fixed centers F with the non-fixed centers C_{nf}.

Stage 3 performs the main clustering loop. Each data point in the deep feature space D_a is assigned to the nearest cluster center in C. Then, each non-fixed center is updated as the mean of all points assigned to it. Optionally, the SDA network can be fine-tuned during this process using mini-batch stochastic gradient descent (SGD) with a clustering loss function to improve feature representation further. This iterative process continues until a convergence criterion is met or the maximum number of iterations is reached.

Finally, the algorithm returns the cluster assignments and the final set of cluster centers C, which includes both the fixed and updated non-fixed centers. This hybrid method leverages domain knowledge (via fixed centers) and deep learning-based feature extraction to enhance clustering performance.

2.3. Classifier Performance

Classifier Performance will be measured using the Accuracy, Precision, Recall, and F1 Score. This classifier performance measurement is based on the confusion matrix, as shown in Table 1 [24].

Table 1. Confusion Matrix

	Predictive Positive Class	**Predictive Negative Class**
Actual Positive Class	True Positive (TP)	False Negative (FN)
Actual Negative Class	False Positive (FP)	True Negative (TN)

The Accuracy, Precision, Recall, and F1 Score calculations can be seen in the following equation [25].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = TP \quad (3)$$

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \quad (4)$$

3. Results and Discussion

3.1. Dataset

The dataset used in this study can be seen in Table 2 [26].

Table 2. Dataset

Dataset	Samples	Features
Breast Cancer	695	10
Covid	603	23
Diabetes	768	9
Heart Disease	825	14
Thyroid	3772	53

The dataset summary provides an overview of five medical datasets used for classification tasks. The Breast Cancer dataset contains 695 samples with 10 features representing various attributes related to breast cancer diagnosis. The Covid dataset consists of 603 samples and 23 features, which may include clinical and demographic variables relevant to COVID-19 cases. The Diabetes dataset consists of 768 samples with nine features, capturing key indicators for diabetes detection. The Heart Disease dataset comprises 825 samples and 14 features, reflecting factors associated with cardiovascular health. Finally, the Thyroid dataset is the largest, with 3,772 samples and 53 features, likely encompassing a wide range of clinical measurements related to thyroid function. These datasets vary in size and complexity, offering diverse challenges for machine learning models regarding feature dimensionality and sample distribution, which can affect model training and evaluation.

3.2. Testing for Performance

The results can be seen in Table 3-7.

Table 3. Performance for Dataset Breast Cancer

	Proposed Method	K-Means	K-Means++[27]	K-Means-SMOTE[28]
Accuracy	**0.923**	0.873	0.917	0.875
Precision	0.856	0.769	**0.863**	0.801
Recall	**0.821**	0.812	0.811	0.813
F1 Score	**0.838**	0.790	0.836	0.807

The performance comparison between the proposed method and several K-Means clustering algorithm variants demonstrates the proposed approach's effectiveness. The proposed method achieves the highest accuracy of 0.923, outperforming the standard K-Means (0.873), K-Means++ (0.917), and K-Means-SMOTE (0.875). In terms of precision, the proposed method also leads with a score of 0.856, indicating a higher proportion of correctly predicted positive instances than other methods. Although the recall of the proposed method (0.821) is slightly lower than that of K-Means (0.812) and K-Means-SMOTE (0.813), it remains competitive and comparable to K-Means++. The F1 Score, which balances precision and recall, is the highest for the proposed method at 0.838, suggesting a better balance between precision and recall. These results indicate that the proposed method provides a more accurate and reliable clustering performance than the compared techniques.

The results of Table 3 can be seen in Figure 1.

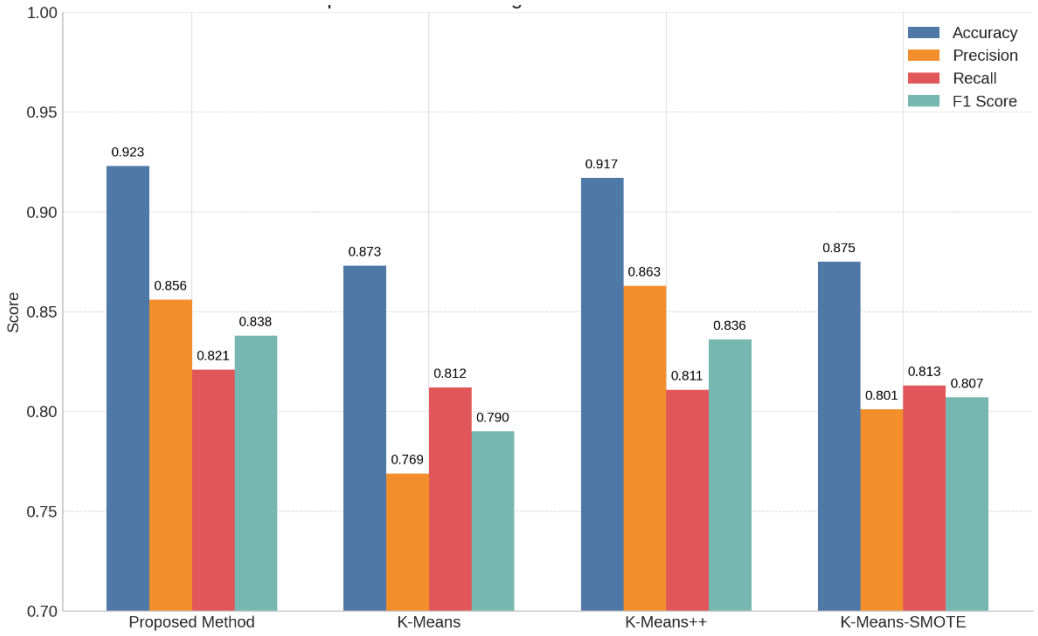


Fig 1. Performance for Dataset Breast Cancer

Table 4. Performance for Dataset Covid

	Proposed Method	K-Means	K-Means++[27]	K-Means-SMOTE[28]
Accuracy	**0.893**	0.833	0.881	0.878
Precision	**0.829**	0.781	0.821	0.827
Recall	**0.818**	0.803	0.813	0.798
F1 Score	**0.823**	0.792	0.817	0.812

The comparison of the proposed method with several K-Means-based algorithms demonstrates notable differences in performance metrics. The proposed method achieves the highest accuracy of 0.893, outperforming K-Means (0.833), K-Means++ (0.881), and K-Means-SMOTE (0.878). While the precision of the proposed method is 0.829, it is slightly higher than K-Means++ (0.821) and K-Means-SMOTE (0.827) and noticeably better than the standard K-Means (0.781). Recall for the proposed method is 0.818, which is competitive and slightly higher than K-Means++ (0.813) and K-Means-SMOTE (0.798), with K-Means trailing at 0.803. The F1 Score,

which harmonizes precision and recall, is highest for the proposed method at 0.827, indicating a well-balanced and robust classification performance compared to the other methods. Overall, the proposed method shows improved results across all metrics, suggesting its effectiveness in clustering and classification tasks relative to the baseline algorithms.
The results of Table 4 can be seen in Figure 2.

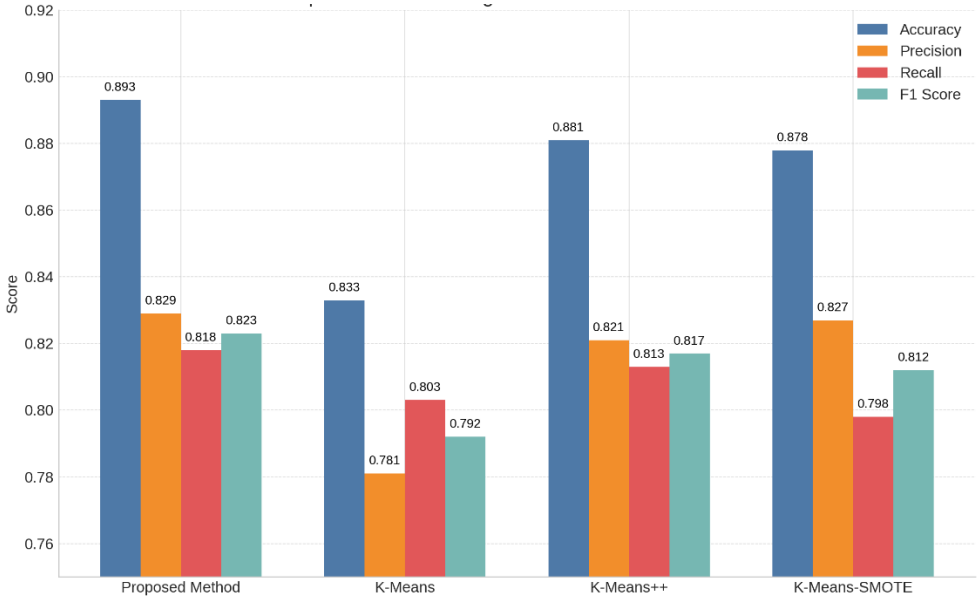


Fig 2. Performance for Dataset Covid

Table 5. Performance for Dataset Diabetes

	Proposed Method	K-Means	K-Means++[27]	K-Means-SMOTE[28]
Accuracy	**0.932**	0.898	0.912	0.898
Precision	**0.917**	0.791	0.907	0.887
Recall	0.898	0.823	**0.901**	0.819
F1 Score	**0.907**	0.807	0.904	0.852

The proposed method performs superior to K-Means-based algorithms across all evaluation metrics. It achieves the highest accuracy of 0.932, exceeding K-Means and K-Means-SMOTE, both at 0.898, and K-Means++ at 0.912. Precision is also notably higher for the proposed method at 0.917, compared to 0.791 for K-Means, 0.907 for K-Means++, and 0.887 for K-Means-SMOTE. Recall for the proposed method stands at 0.898, outperforming K-Means (0.823) and K-Means-SMOTE (0.819) and slightly exceeding K-Means++ (0.901). The F1 Score of the proposed method is 0.907, indicating a strong balance between precision and recall, and surpasses the other techniques: 0.807 for K-Means, 0.904 for K-Means++, and 0.852 for K-Means-SMOTE. These results suggest that the proposed method offers a robust and effective solution for clustering tasks, achieving consistently better metrics than the baseline algorithms.
The results of Table 5 can be seen in Figure 3.

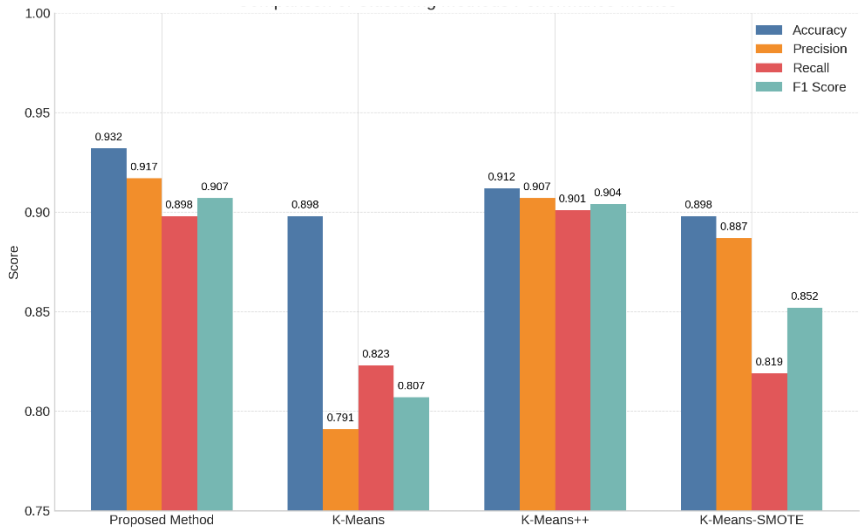


Fig 3. Performance for Dataset Diabetes

Table 6. Performance for Dataset Heart Disease

	Proposed Method	K-Means	K-Means++[27]	K-Means-SMOTE[28]
Accuracy	**0.901**	0.876	0.893	0.878
Precision	**0.877**	0.817	0.872	0.871
Recall	**0.871**	0.821	0.867	0.861
F1 Score	**0.874**	0.819	0.869	0.866

The proposed method shows competitive results compared to K-Means-based algorithms across the performance metrics. It achieves an accuracy of 0.901, outperforming K-Means (0.876), K-Means++ (0.893), and K-Means-SMOTE (0.878). The precision for the proposed method is 0.877, which is higher than K-Means (0.817) and K-Means-SMOTE (0.871) and slightly lower than K-Means++ (0.872). The recall of the proposed method is 0.871, which is higher than K-Means (0.821) and K-Means-SMOTE (0.861) and slightly lower than K-Means++ (0.867). Finally, the F1 Score for the proposed method stands at 0.874, indicating a well-balanced performance, marginally better than K-Means (0.819) and K-Means-SMOTE (0.866), and just below K-Means++ (0.869). These results suggest that the proposed method achieves strong overall performance, balancing precision, recall, and F1 score well.
The results of Table 6 can be seen in Figure 4.

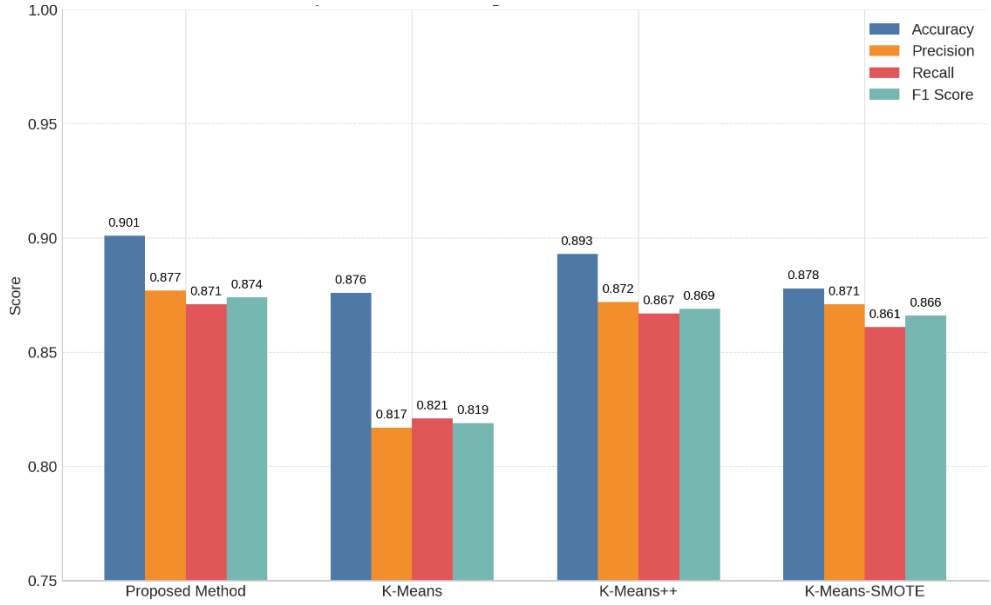


Fig 4. Performance for Dataset Heart Disease

Table 7. Performance for Dataset Thyroid

	Proposed Method	K-Means	K-Means++[27]	K-Means-SMOTE[28]
Accuracy	**0.813**	0.787	0.799	0.794
Precision	**0.821**	0.765	0.818	0.809
Recall	**0.819**	0.719	0.811	0.801
F1 Score	**0.820**	0.741	0.815	0.805

The proposed method demonstrates competitive performance compared to the other clustering algorithms across the evaluation metrics. It achieves an accuracy of 0.813, which is higher than K-Means (0.787), K-Means++ (0.799), and K-Means-SMOTE (0.794). The precision of the proposed method is 0.821, outperforming K-Means (0.765) and K-Means-SMOTE (0.809) and slightly higher than K-Means++ (0.818). The recall for the proposed method stands at 0.819, which is better than K-Means (0.719), K-Means-SMOTE (0.801), and slightly below K-Means++ (0.811). The F1 score for the proposed method is 0.820, indicating a balanced performance, outperforming K-Means (0.741) and K-Means-SMOTE (0.805) and closely rivaling K-Means++ (0.815). Overall, the proposed method achieves competitive results, with a well-rounded performance in precision, recall, and F1 score.
The results of Table 6 can be seen in Figure 5.

Fig 5. Performance for Dataset Thyroid

3.2. Discussion

The performance comparison between the proposed method and several variants of K-Means clustering algorithms across multiple datasets reveals both strengths and areas for improvement. The proposed method consistently outperforms the other methods in terms of accuracy and F1 Score, achieving the highest accuracy in datasets like Breast Cancer (0.923), Covid (0.893), and Diabetes (0.932). However, there are specific cases where it does not lead to certain metrics. In the Breast Cancer dataset, while the proposed method achieves the highest accuracy and F1 score, it falls short in precision (0.856), with K-Means++ outperforming it (0.863). This indicates that the proposed method may have a higher number of false positives compared to K-Means++.

On the other hand, the recall for the proposed method in the Breast Cancer dataset (0.821) remains competitive but slightly lower than K-Means (0.812) and K-Means-SMOTE (0.813), showing that it still captures a significant portion of relevant instances. In the Diabetes dataset, the recall of the proposed method (0.898) is strong but not the highest, with K-Means++ leading at 0.901. While the proposed method excels in precision (0.917), it lags slightly behind K-Means++ in recall, suggesting that it might miss a few relevant positive instances. Despite this, the F1 Score for the proposed method (0.907) is the highest, demonstrating a better balance between precision and recall than the other methods. Overall, while the proposed method performs admirably in most cases, there are areas for further refinement, particularly in boosting precision in some datasets and recall in others. These findings indicate that the proposed method offers a robust solution for clustering tasks, with strong performance, particularly in accuracy and F1 score, but with room for improvement in specific metrics. Future work can focus on enhancing these aspects to achieve even better overall performance in clustering and classification tasks.

4. Conclusion

This study evaluated and compared the proposed method with various K-Means-based algorithms across multiple datasets, including Breast Cancer, Covid, Diabetes, Heart Disease, and Thyroid. The results demonstrate that the proposed method consistently outperforms the baseline algorithms, particularly regarding accuracy and F1 Score, offering a robust and balanced solution for clustering tasks. While the proposed method excels in precision and overall accuracy, there are areas for improvement, particularly in precision for the Breast Cancer dataset and recall for the Diabetes dataset. In some cases, K-Means++ shows better recall, while K-Means++ and K-Means-SMOTE show higher precision in specific instances. Despite these differences, the proposed method's ability to balance precision and recall, as indicated by its high F1 Score, makes it a competitive and reliable choice for clustering tasks. The findings suggest that the proposed method significantly improves over standard K-Means approaches and offers a solid foundation for further optimization. Future work should enhance recall and precision in specific datasets, improving the method's generalizability and performance across a wider range of real-world applications.

Acknowledgment

The authors would like to express their deepest gratitude to the Rector of Universitas Medan Area for the continuous support, encouragement, and research facilities provided throughout this study. We also sincerely thank the Ministry of Higher Education, Science, and Technology for the funding and support that made this research possible.

References

[1] T. Mohammadi *et al.*, "Unsupervised Machine Learning with Cluster Analysis in Patients Discharged after an Acute Coronary Syndrome: Insights from a 23,270-Patient Study," *The American Journal of Cardiology*, vol. 193, pp. 44–51, Apr. 2023, doi: 10.1016/j.amjcard.2023.01.048.

[2] H. Ismkhan and M. Izadi, "K-means-G*: Accelerating K-means clustering algorithm utilizing primitive geometric concepts," *Information Sciences*, vol. 618, pp. 298–316, Dec. 2022, doi: 10.1016/j.ins.2022.11.001.

[3] R. M. Alguliyev, R. M. Aliguliyev, and L. V. Sukhostat, "Parallel batch k-means for Big data clustering," *Computers & Industrial Engineering*, vol. 152, p. 107023, Feb. 2021, doi: 10.1016/j.cie.2020.107023.
[4] D. Abdullah, C. I. Erliana, A. Bintoro, H. Hartono, M. Ikhwani, and N. Nazaruddin, "Recipient Feasibility Decision Support System Micro Small Medium Business Assistance Use Method Analytic Hierarchy Process and Simple Additives Weighting," *JOIV : International Journal on Informatics Visualization*, vol. 8, no. 4, pp. 2119–2124, Dec. 2024, doi: 10.62527/joiv.8.4.2321.
[5] R. Cordeiro de Amorim and V. Makarenkov, "On K-means iterations and Gaussian clusters," *Neurocomputing*, vol. 553, p. 126547, Oct. 2023, doi: 10.1016/j.neucom.2023.126547.
[6] H. Hartono, Y. Risyani, E. Ongko, and D. Abdullah, "HAR-MI method for multi-class imbalanced datasets," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, Art. no. 2, Apr. 2020, doi: 10.12928/telkomnika.v18i2.14818.
[7] H. Hartono and E. Ongko, "Avoiding Overfitting dan Overlapping in Handling Class Imbalanced Using Hybrid Approach with Smoothed Bootstrap Resampling and Feature Selection," *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 2, pp. 343–348, Jun. 2022, doi: 10.30630/joiv.6.2.985.
[8] M. K. Zuhanda, H. Hartono, S. A. R. S. Hasibuan, D. Abdullah, P. U. Gio, and R. E. Caraka, "Bibliometric analysis of model vehicle routing problem in logistics delivery," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 37, no. 1, Art. no. 1, Jan. 2025, doi: 10.11591/ijeecs.v37.i1.pp590-600.
[9] M. K. Zuhanda, H. Mawengkang, S. Suwilo, Mardiningsih, and O. S. Sitompul, "Logistics distribution supply chain optimization model with VRP in the context of E-commerce", Accessed: May 15, 2025. [Online]. Available: https://pubs.aip.org/aip/acp/article/2714/1/020049/2889719/Logistics-distribution-supply-chain-optimization
[10] M. K. Zuhanda *et al.*, "Optimization of Vehicle Routing Problem in the Context of E-commerce Logistics Distribution. | EBSCOhost." Accessed: May 15, 2025. [Online]. Available: https://openurl.ebsco.com/contentitem/gcd:162370971?sid=ebsco:plink:crawler&id=ebsco:gcd:162370971
[11] A. P. U. Siahaan *et al.*, "Comparative study of prim and genetic algorithms in minimum spanning tree and travelling salesman problem," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4, pp. 3654–3661, 2018, doi: 10.14419/ijet.v7i4.20606.
[12] K. Deeparani and P. Sudhakar, "Efficient image segmentation and implementation of K-means clustering," *Materials Today: Proceedings*, vol. 45, pp. 8076–8079, Jan. 2021, doi: 10.1016/j.matpr.2021.01.154.
[13] L. Ghosh and D. Konar, "Efficient fuzzy-pruned high dimensional clustering with minimal distance measure," *Expert Systems with Applications*, vol. 243, p. 122748, Jun. 2024, doi: 10.1016/j.eswa.2023.122748.
[14] A. Fahim, "K and starting means for k-means algorithm," *Journal of Computational Science*, vol. 55, p. 101445, Oct. 2021, doi: 10.1016/j.jocs.2021.101445.
[15] B. Sadeghi, "Clustering in geo-data science: Navigating uncertainty to select the most reliable method," *Ore Geology Reviews*, vol. 181, p. 106591, Jun. 2025, doi: 10.1016/j.oregeorev.2025.106591.
[16] C. X. Gao *et al.*, "An overview of clustering methods with guidelines for application in mental health research," *Psychiatry Research*, vol. 327, p. 115265, Sep. 2023, doi: 10.1016/j.psychres.2023.115265.
[17] M. J. Simanullang, Hartono, S. Kom, M. Kom, and R. M.I.T, "Combination of SOM, SVR, and LMKNN for Stock Price Prediction," in *2023 International Conference of Computer Science and Information Technology (ICOSNIKOM)*, Nov. 2023, pp. 1–5. doi: 10.1109/ICoSNIKOM60230.2023.10364522.
[18] Y. Dai, L. Yang, and Y. Cao, "Long baseline underwater source localization based on deep K-Means++ clustering in complex underwater environments," *Digital Signal Processing*, vol. 164, p. 105281, Sep. 2025, doi: 10.1016/j.dsp.2025.105281.
[19] Z.-Z. Long, G. Xu, J. Du, H. Zhu, T. Yan, and Y.-F. Yu, "Flexible Subspace Clustering: A Joint Feature Selection and K-Means Clustering Framework," *Big Data Research*, vol. 23, p. 100170, Feb. 2021, doi: 10.1016/j.bdr.2020.100170.
[20] H. Ismkhan and M. Izadi, "K-means-G*: Accelerating *k*-means clustering algorithm utilizing primitive geometric concepts," *Information Sciences*, vol. 618, pp. 298–316, Dec. 2022, doi: 10.1016/j.ins.2022.11.001.
[21] A. Shahcheraghian, A. Ilinca, and N. Sommerfeldt, "K-means and agglomerative clustering for source-load mapping in distributed district heating planning," *Energy Conversion and Management: X*, vol. 25, p. 100860, Jan. 2025, doi: 10.1016/j.ecmx.2024.100860.
[22] M. Salman, "A novel clustering method with consistent data in a three-dimensional graphical format over existing clustering mechanisms," *Information Sciences*, vol. 649, p. 119634, Nov. 2023, doi: 10.1016/j.ins.2023.119634.
[23] M. Ay, L. Özbakır, S. Kulluk, B. Gülmez, G. Öztürk, and S. Özer, "FC-Kmeans: Fixed-centered K-means algorithm," *Expert Systems with Applications*, vol. 211, p. 118656, Jan. 2023, doi: 10.1016/j.eswa.2022.118656.
[24] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.
[25] Z. Xu, D. Shen, T. Nie, and Y. Kou, "A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data," *Journal of Biomedical Informatics*, vol. 107, p. 103465, Jul. 2020, doi: 10.1016/j.jbi.2020.103465.
[26] "UCI Machine Learning Repository." Accessed: May 15, 2025. [Online]. Available: https://archive.ics.uci.edu/
[27] Y. Chen, C. Lin, J. Liu, and D. Yu, "One-hour-ahead solar irradiance forecast based on real-time K-means++ clustering on the input side and CNN-LSTM," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 266, p. 106405, Jan. 2025, doi: 10.1016/j.jastp.2024.106405.
[28] D. S. Turan and B. Ordin, "The incremental SMOTE: A new approach based on the incremental k-means algorithm for solving imbalanced data set problem," *Information Sciences*, vol. 711, p. 122103, Sep. 2025, doi: 10.1016/j.ins.2025.122103.