International Journal of Electrical and Computer Engineering (IJECE)
Vol. 16, No. 3, June 2026, pp. 1434~1448
ISSN: 2088-8708, DOI: 10.11591/ijece.v16i3.pp1434-1448

Variance-k-means++: A deterministic centroid initialization method based on variance for enhanced clustering stability

Widodo[1,2], Jiel Vayyad Ramadhan[1,2], Muhammad Ficky Duskarnaen[1], Via Tuhamah Fauziastuti[1], Chelsea Zaomi Pondayu[1,2], Mada Rekadarma Septianda[1,2]
[1]Department of Informatics and Computer Engineering Education, Universitas Negeri Jakarta, Jakarta, Indonesia
[2]Research Group on Machine Learning and Natural Language Processing, Universitas Negeri Jakarta, Jakarta, Indonesia

Article Info

Article history:

Received Sep 16, 2025
Revised Jan 27, 2026
Accepted Mar 16, 2026

Keywords:

Davies-Bouldin index
Initial centroid
K-means++
Pseudo-centroid
Variance-k-means++

ABSTRACT

K-means++ is developed to improve the performance of k-means when choosing a starting centroid. However, both algorithms in clustering still select an initial centroid randomly. Randomly selecting initial centroids has the potential to produce unstable clusters. This paper proposes a deterministic centroid initialization method called variance-k-means++, which utilizes statistical properties—mean and variance—to generate pseudo-centroids and derive initial centroids. The method aims to improve clustering stability and reduce the number of iterations. For the initial study, we used low-dimensional data to conduct the experiment series. Then, we employed two baseline methods for benchmarking, k-means and k-means++. The results show that variance-k-means++ outperformed the baseline method on average. Evaluating in Davies-Bouldin index (DBI) and convergence analysis, we obtained DBI values at 0.756 and 0,771 for vertical and horizontal variance k-means++ with Iris dataset. At the same time, baseline methods have 0.802 and 0.830 for k-means++ and k-means, respectively. In convergence analysis, the results are 5.158 for vertical and 5.474 for horizontal, while baseline methods are 9.000 and 8.842. The primary contribution of this study lies in its achievement of minimizing the number of iterations while enhancing cluster stability.

Corresponding Author:

Widodo
Research Group on Machine Learning and Natural Language Processing, Department of Informatics and Computer Engineering Education, Universitas Negeri Jakarta
Jl. Rawamangun Muka, East Jakarta, Jakarta, Indonesia
Email: widodo@unj.ac.id

1. INTRODUCTION

Clustering is a task in machine learning that aims to group data into clusters. This task is categorized as unsupervised learning, where the learning process does not need to be guided by class labels. Majorly, there are two types of clustering: hierarchical and partitional clustering [1]. We focus on one partitional clustering algorithm called k-means clustering. K-means has become a popular algorithm because of its simplicity, computation efficiency, ease of implementation, and work in many data types [2]–[4]. The entire k-means clustering process is straightforward, whether applied to simple or complex data. This leads to the algorithm's simplicity and provides efficiency [5]. However, k-means has several limitations. Those limitations are in the centroid initialization process, determination of the number of k, the curse of dimensionality, and the difference in cluster density. K-means is also sensitive to outlier data, where outliers lead to unstable results [6]. With random centroid initialization, k-means tends to produce unstable clusters.

In addition, if clustering is conducted using different centroids, there is a possibility of producing different clusters [7], [8].

When centroids are initialized randomly, the algorithm may converge to a local optimum instead of the global optimum. This implies that multiple algorithm executions on the same dataset may provide disparate outcomes. For instance, clusters that are not indicative of the underlying data distribution may result from initial centroids that are not well picked. Additionally, the random selection of initial centroids yields inconsistent results [9]. This decreases the algorithm's dependability because using a different centroid may yield different clustering results. A method called k-means++ has been developed to overcome one of the limitations of k-means [10]. An improved initialization of cluster centers is the goal of k-means++, a sophisticated variant of the classic k-means clustering algorithm. Choosing starting centroids in a way that improves clustering performance and lowers the possibility of subpar clustering outcomes because of random initialization is the main objective of k-means++. However, k-means++ has the same issues as traditional k-means because the initial centroids are chosen at random, just like in traditional k-means.

This paper proposes a deterministic centroid initialization method using variance-based pseudo-centroids to improve clustering stability and reduce number of iterations. This method avoids a random centroid for initial choice. After identifying the data's center point, the variance value is calculated and then drawn to the right and left by half of the variance value each. The stability of clusters, a low Davies-Bouldin index (DBI), and a minimum number of iterations are the primary objectives of this method.

The contributions of this paper are i) proposing a method for initializing centroid without random choice, specifically for the first of two centroids, ii) minimizing the number of iterations until obtaining the stable clusters, and iii) developing pseudo-centroid, a pre-centroid that is created using variance-distance. The remaining of this paper is organized as follows: section 2 describes the literature review, section 3 discusses the proposed method, results and discussion are described in section 4, and we conclude in section 5.

2. LITERATURE REVIEW

Several studies on centroid initialization using probabilistic and seed distribution methods have been conducted effectively. Hybrid k-means++ [11], supervoxel segmentation [12], Intuitive k-prototype [13], lasso-based k-means++ [14], and Beyond k-means++ [15] were developed to address the issue of random initialization. Almost all of those are still based on traditional k-means. Although the results are commendable and outperform the baseline method across several categories, the fact that it still adheres to traditional k-means indicates that the determination of the initial centroid does not effectively control the stability of the resulting cluster.

The statistical approach has also been utilized for the initialization of the centroid, including measures such as the mean and median. A novel technique referred to as depth difference (DeD), which enhances the efficacy of the k-means algorithm [16]. Alternative statistical methods, including a hybrid of the fixed row initial centroid technique with naive Bayes [17], determinantal point process [18], and Adaptive k-means clustering [19], have been proposed to mitigate the instability of clusters associated with k-means. Nevertheless, these approaches still struggle to maintain stability and the number of iterations due to the reliance on random initial centroids.

Numerous researchers have introduced optimization and metaheuristic approaches to address this issue. Tian *et al.* [20] suggested a method called partial label via weighted centroid clustering (PL-CC). In addition, Karim *et al.* [21] proposed the max stable set problem (MSSP) for centroid selection. Although this approach optimizes the initial centroid selection for k-means, it may result in unstable outcomes because k-means can potentially get trapped in local minima. Other optimization techniques, such as invasion weed optimization (IWO) [22], [23] and genetic algorithm (GA) [24], [25], have also performed effectively. Both IWO and GA have been integrated into a more accurate method [26]. An optimization method based on an evolutionary technique using random swap was proposed by Nigro and Cicirelli [27]. Unfortunately, those all still address the initial centroid randomly, which leads to unstable clusters.

A further emphasis of research that has been performed is grounded in density-based techniques and principal component analysis (PCA). One of the techniques is adaptive k-means (AK-Means). AK-Means significantly boosts clustering efficiency by tackling the key shortcomings of standard k-means; however, it still incurs a heavy computational burden when applied to high-dimensional data and has potential for further refinement. Furthermore, there are three additional methods referred to as the k-means algorithm based on nearest neighbor density peak (k-NNDP) [28], spherical k-means [29], and PCA [30], [31]. These methods operate effectively and exceed the performance of the traditional k-means++ algorithm. Regrettably, since the initialization process remains randomly determined, they are also likely to produce unstable clusters.

3. METHOD

The goal of this research is to propose a method that overcomes the limitations of the k-means and k-means++ algorithms. Both algorithms use a random centroid for initialization, which potentially creates unstable clusters. With different initial centroids, it is also possible to yield different clusters and different numbers of iterations. Our proposed method aims to address these issues. This section begins with a description of the rationale of the proposed method, followed by a brief review of k-means and k-means++, and finally, variance-k-means++.

3.1. Rationale

The randomness in the initial centroid selection process of k-means and k-means++ represents a significant limitation. Consequently, variance-k-means++ relies on two statistical methods: mean and variance. To eliminate the randomness associated with initial centroid selection, the mean value of the data is utilized. This approach ensures that centroid selection remains consistent, as it is always anchored to the mean value of the data. Variance is the expected value of a random variable's squared deviation from its mean. Variance provides information about how spread out the points are [32]. The incorporation of the variance value stems from the understanding that variance reflects the distribution of the data; thus, the data point that is farthest from the mean would not exceed the variance distance. Therefore, by dividing the variance by two and measuring the distance from the mean, two centroids can be positioned further apart, leading to improved and more stable clustering.

3.2. K-means

K-means is an algorithm for clustering problems, and it is commonly used because the algorithm is simple and easy to use. However, k-means has some problems mentioned in section 1. Suppose a numerical dataset $X = \{x_1, x_2, \ldots, x_n\}$, where n is the number of data. First, the number of clusters (k) is denoted. Then, the algorithm randomly determines the centroid as many as k. Each of the data is measured to each centroid. A precise distance calculation using a certain formula is employed to measure each data point to each centroid at each iteration. The closest centroid is given the location. Next, the centroid is recalculated to get the midpoint for each cluster. Iteration ends if the centroid does not change; if it does, iteration proceeds [33]. Algorithm 1 shows the k-means clustering algorithm pseudocode.

Algorithm 1. K-means clustering

```
Input:
    D ← dataset containing n data points {x₁, x₂, ..., xₙ}
    k ← desired number of clusters
    max_iter ← maximum number of iterations (optional)

Output:
    C ← set of centroids {c₁, c₂, ..., cₖ}
    L ← cluster labels for each data point

Begin
    1: Initialize centroids C = {c₁, c₂, ..., cₖ} randomly from D
    2: iter ← 0
    3: repeat
    4:     for each data point xᵢ ∈ D do
    5:         Compute Euclidean distance between xᵢ and each centroid cⱼ
    6:         Assign label lᵢ ← argminⱼ distance(xᵢ, cⱼ)
    7:     end for
    8:     for each centroid cⱼ ∈ C do
    9:        Update cⱼ ← mean of all xᵢ such that lᵢ = j
    10:    end for
    11:    iter ← iter + 1
    12: until convergence or iter ≥ max_iter
    13: Return C and L
End
```

3.3. K-means++

K-means++ is developed to handle the limitations of k-means clustering. This algorithm differs from classical k-means when figuring out the initial centroid. The k-means++ algorithm enhances the conventional k-means method by systematically choosing the initial centroids, thereby improving clustering efficiency and decreasing the probability of converging to suboptimal local minima [4], [15]. Algorithm 2 shows k-means clustering algorithm pseudocode.

Algorithm 2. K-means++ Initialization

```
Input:
    D ← dataset containing n data points {x1, x2, ..., xn}
    k ← desired number of clusters

Output:
    C ← set of initial centroids {c1, c2, ..., ck}

Begin
    1: Choose the first centroid c1 randomly from D
    2: C ← {c1}
    3: while |C| < k do
    4:     for each data point x ∈ D do
    5:         Compute D(x) ← distance from x to the nearest centroid in C
    6:     end for
    7:     Compute probability distribution P(x) ∝ D(x)² for all x ∈ D
    8:     Select next centroid ci from D using probability P(x)
    9:    Add ci to C
    10: end while
    11: Return C
End
```

3.4. Variance k-means++ (proposed method)

A limitation of both k-means and k-means++ algorithms is their reliance on randomly selected initial centroids. This stochastic initialization can lead to unstable clustering outcomes, as different choices of initial centroids may yield varying cluster configurations. Variance k-means aims to mitigate this problem by evaluating the variance of the data. Initially, the algorithm computes the mean of the dataset, which serves as the central point. Subsequently, the variance is determined and halved. This half-variance is then employed to draw lines extending from the central point outward to the right and left. The endpoints of these lines are designated as pseudo-centroids. The real two centroids are determined based on these pseudo-centroids. The algorithm of Variance-k-means++ consists of three main stages, as shown in Figure 1, those are statistical processing, variance distance measurements, and centroid initialization. This algorithm commences upon the introduction of the dataset into the first stage. Below is the detailed description of each stage:

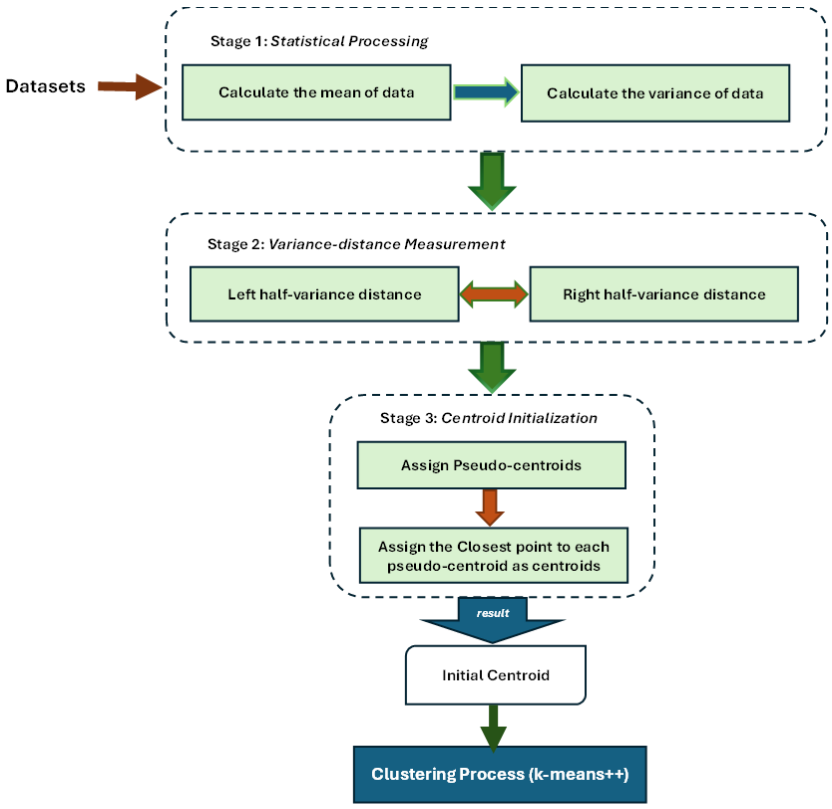

Figure 1. Framework outline of variance k-means++

3.4.1. Statistical processing

This step has two processes, calculating the mean of the data and calculating the variance of data. First, calculate the mean of the data. This process helps find the midpoint. This midpoint acts as a solution to ensure the starting points, or centroids, are not chosen randomly but are instead organized in a way that makes the final clusters more stable. The formula for the mean is shown in (1).

$$x = \frac{\sum x}{n} \tag{1}$$

x is the mean value, $\sum x$ is the sum of data, while n is number of the data.

Calculating variance aims to understand the distribution of data. When the furthest point is measured from the midpoint, it should not be farther than the half-variance point. Since this characteristic, half-variance can be employed to obtain the furthest points (left and right) and choose them as pseudo-centroids. pseudo-centroid is a pseudo-point that is used as a basis for determining the initial centroid. Formula (2) shows the variance.

$$S^2 = \frac{\sum(x_i - x)^2}{n} \tag{2}$$

S^2 is variance, x_i is the i^{th} datapoint, x is mean, while n is the number of data points.

3.4.2. Variance-distance measurement

Variance-distance measurement (VDM) is used to find the furthest point measured based on the variance value. For horizontal VDM, measurements are taken both to the right and left of the mean position. Vertical VDMs are measured both up and down. The distances are half of the variance from the mean spot. Formulas (3) and (4) describe the process.

$$VDM_{left-down} = mean - (\frac{S^2}{2}) \tag{3}$$

$$VDM_{right-up} = mean + (\frac{S^2}{2}) \tag{4}$$

VDM is the pseudo-centroid points, while *mean* is the mean value. $\frac{S^2}{2}$ is half-variance.

Figure 2 depicts the illustration of VDM. The green points show datapoints, the red one represents the mean point, while the yellows are the pseudo-centroids. The dashed lines exhibit the variance distance. VDM is a pseudo-centroid point, which is the starting point for obtaining the initial centroid.

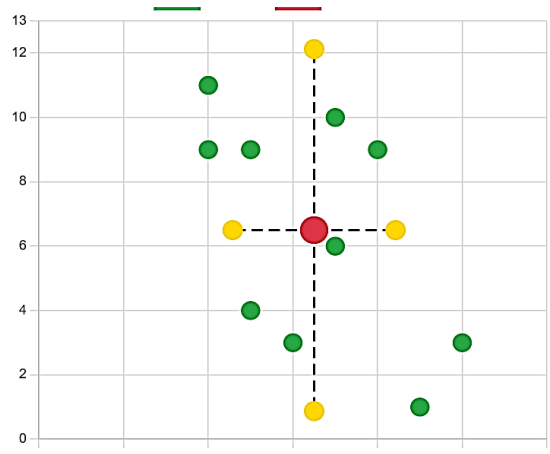

Figure 2. Illustration of VDM

3.4.3. Centroid initialization

Two processes are conducted in this stage. An assignment of pseudo-centroid and assignment of the closest point to the pseudo-centroid as the initial centroid. Assignment of pseudo-centroid is to assign each of

the endpoints of VDM as a pseudo-centroid. This is called pseudo-centroid since it is not a real centroid, but a basis point for finding the proper initial centroid. From Figure 2, the pseudo-centroids are represented by the yellow points.

The next process is to assign the closest point to the pseudo-centroid as an initial centroid. This process is conducted to the most left pseudo-centroid and the most right pseudo-centroid for horizontal variance, and the most up and the most bottom pseudo-centroid for vertical variance. The distance can be measured using Euclidean distance to obtain the closest one. The centroid initialization results in two initial centroids, because the focus of this research is to identify two initial centroids. Algorithm 3 describes the procedure of variance-k-means++.

Algorithm 3. Variance-k-means++

```
Input:
    Dataset X = {x1, x2, ..., xn}, number of clusters k

Output:
    Initial centroids for k-means

Procedure VarianceKMeansPlusPlus(X, k):
Begin
1. Compute the global mean of the dataset:
       mean_x = (1/n) * Σ xi.x
       mean_y = (1/n) * Σ xi.y
       mean_point = (mean_x, mean_y)
2. Compute variance along each axis:
       var_x = Variance({xi.x})
       var_y = Variance({xi.y})
3. Generate pseudo-centroids around the mean:
       p1 = (mean_x + (var_x/2), mean_y)   // right
       p2 = (mean_x - (var_x/2), mean_y)   // left
       p3 = (mean_x, (mean_y + var_y/2))   // up
       p4 = (mean_x, (mean_y - var_y))   // down
4. For each pseudo-centroid pj:
       Find the data point x* in X that minimizes distance(x*, pj)
       Mark x* as a candidate centroid
5. Select initial centroids:
       If k ≤ 4:
            Choose the first k candidate centroids
       Else:
            Use all four candidate centroids
            Remaining (k-4) centroids will be chosen in step 6
6. Continue with standard k-means++ seeding:
       While number of chosen centroids < k:
            For each data point xi in X:
                 Compute D(xi) = squared distance to the nearest chosen centroid
            Select a new centroid with probability:
                 P(xi) = D(xi) / Σ D(xj) over all j
       End While
7. Return the set of k initial centroids
End
```

3.5. Evaluation

In clustering problems, the algorithm is evaluated using DBI. The DBI is an internal evaluation metric used to assess the performance of clustering algorithms. Its objective is to measure the quality of separation and density among clusters without requiring ground truth labels [34], [35]. Formula (5) explains how DBI works.

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}\left(\frac{s_i + s_j}{d(c_i, c_j)}\right) \quad (5)$$

DBI is the value of Davies Bouldin index, k is the number of clusters, s_i is the average distance between points in cluster i and its centroid, while $d(c_i, c_j)$ is the distance between centroid i and centroid j.

The DBI value is utilized to evaluate the degree of cohesiveness within clusters and the distance separating them. The closeness of members within a cluster indicates its cohesiveness, while the distance between clusters is evaluated by the degree to which they are separated from each other. A smaller DBI value points to better cluster formation. A favorable DBI value is within the range of 0 to 1, whereas a DBI value that surpasses 1 suggests potential overlap among the clusters.

Another approach to evaluate clusters is by determining their convergence level. This convergence can be observed from the number of iterations that take place until the cluster achieves stability [21], [36].

Minimizing the number of iterations contributes to more effective convergence. This indicates that these clusters attain a stable level at a faster rate.

4. RESULTS AND DISCUSSION

To evaluate our proposed method, a series of experiments was conducted. The results were assessed using the DBI and the number of iterations as a measure of convergence analysis. The lower DBI value indicates that the clusters are better cohesive and are well separated, while the fewer the number of iterations, the more efficient the cluster convergence.

4.1. Datasets

Datasets were collected from UCI machine learning, an online repository. Two datasets are used, Iris dataset and Seeds dataset. The use of two datasets. These two datasets were selected because their characteristics fit the criteria for this study. Both the Iris dataset and the Seeds dataset are frequently used in various experiments related to classification and clustering tasks.

The Iris dataset consists of 150 records, and this data is typically used for classification problems. The dataset consists of three distinct classes, each with 150 instances, where every class corresponds to a particular type of iris plant. In this experiment, we removed the class labels since the task of our research is clustering, rather than classification. Another dataset, Seeds, contains 210 instances and 7 features with 3 classes. For our research purposes, the dataset was tested in a low dimension. This initial investigation is only to understand the impact of the proposed method on low-dimensional data. The Seeds dataset was also treated similarly to Iris dataset. This initial investigation was conducted for low-dimensional data; therefore, both datasets were set to only two features. To implement this, dimension reduction is performed through the use of principal component analysis (PCA).

4.2. Results

For benchmarking, two methods, k-means and k-means++, were employed as baseline methods. The number of data is 151 for Iris and 221 for Seeds. The number of clusters, k, is set between 2 and 20, because the minimum number of clusters should be 2, and the maximum number of clusters is 20. The proposed method and the baseline methods were set to 2 features for initial investigation.

Figure 3 shows the result of our series of experiments using Iris datasets, where VKM++ consistently records a lower DBI across k, which points to tighter intra-cluster cohesion and improved inter-cluster separation. The dispersion-aware anchors help to reduce the chances of Seeds being poorly positioned near the boundaries. This indicates more stable partitions and a better structure. In the graph, the orange line signifies VKM++ vertical, the blue line denotes VKM++ horizontal, the green line signifies k-means++, and the red line depicts traditional k-means respectively.

Figure 4 shows the comparison of the average value of DBI for Iris datasets, all the values are below 1.00. The proposed method has a lower mean than k-means and k-means++, which indicates that, on average, the vertical and horizontal VKM++ have well-separated clusters. The proposed methods have the horizontal axis represents the methods, while the vertical axis represents the average values. The bar of VKM++ vertical represented by blue, VKM++ horizontal by yellow, k-means++ by green, and k-means by red.

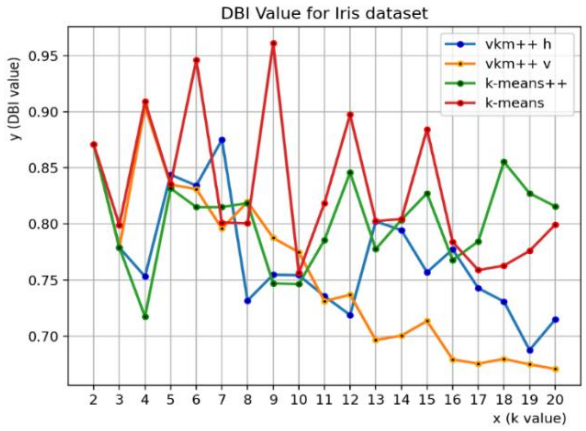

Figure 3. Davies-Bouldin index result for Iris dataset

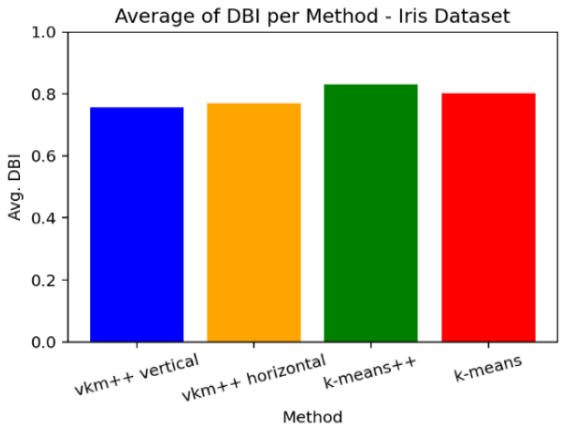

Figure 4. The average values of DBI for Iris dataset

Figure 5 shows the number of iterations required from k=2 to k=20 for Iris dataset, where VKM++ demands fewer iterations to reach convergence. The early assignments stabilize more quickly owing to Seeds that are situated near dispersion-informed extremes, thereby minimizing reassignment fluctuation. This produces notable runtime benefits in practical clustering scenarios. In this representation, the orange line illustrates VKM++ vertical, the blue line corresponds to VKM++ horizontal, the green line depicts k-means++, and the red line represents k-means.

The average values of iteration numbers for Iris dataset are depicted in Figure 6. The average number of iterations for vertical and horizontal VKM++ is less than that of k-means and k-means++. This demonstrates that the proposed technique is more efficient and converges more swiftly. The iteration average of each method is represented by its color. The vertical axis corresponds to the average value, while the horizontal axis represents the methods. The bar of VKM++ vertical represented blue, VKM++ horizontal by orange, k-means++ by green, and k-means by red.

For the Seeds dataset, we used the same scenario to experiment. The experimental results of the DBI using the Seeds dataset are shown in Figure 7. In essence, VKM++ is still more simplified than the other two baseline methods, with the unique aspect being that both vertical and horizontal VKM++ possess identical values. Consistent with the prior figures, the brown shade indicates VKM++ vertical, the blue shade represents VKM++ horizontal, while the green and red shades correspond to k-means++ and k-means, respectively. The graph shows that all methods appear to be relatively balanced, except for the k-means.

The average of DBI values is depicted in Figure 8. The graphic appears relatively balanced, except for the traditional k-means. This indicates that the dataset derived from the seed generates well-balanced clusters, with the exception of traditional k-means. The attributes are blue for VKM++ vertical, orange for VKM++ horizontal, while green and red for k-means++ and k-means.

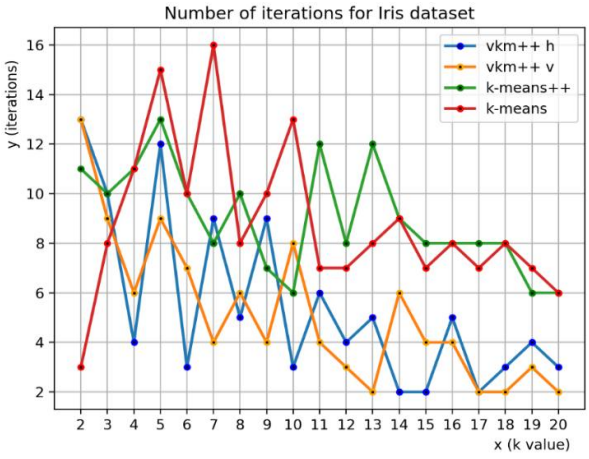

Figure 5. Number of iterations Iris dataset

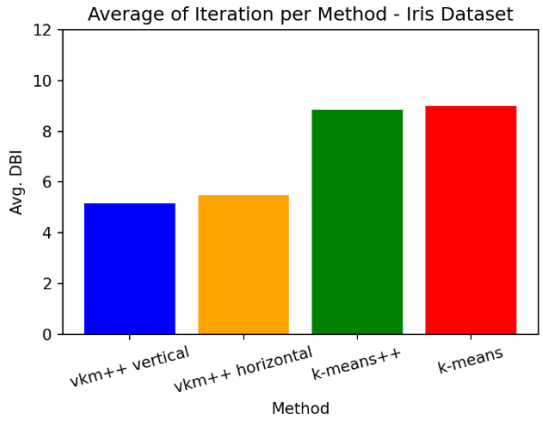

Figure 6. The average iterations for Iris dataset

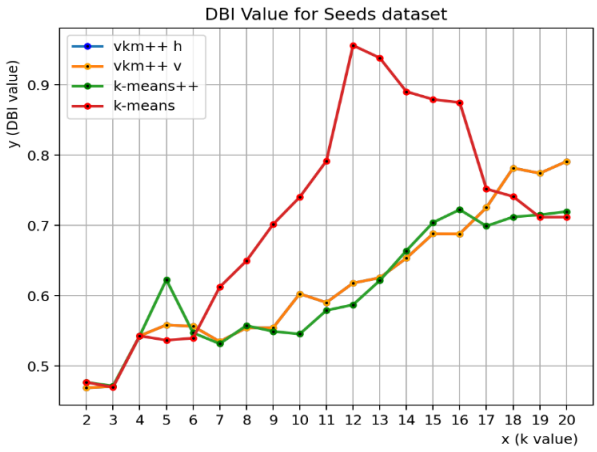

Figure 7. Daves-Bouldin index Seeds dataset

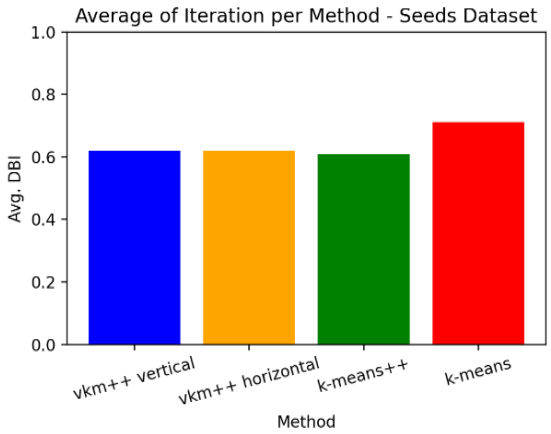

Figure 8. The average DBI for Seeds dataset

Figure 9 shows that variance-k-means++ consistently demands fewer iterations than both k-means and k-means++. This indicates the robustness of deterministic centroid initialization, which circumvents the variable cluster assignments that often happen when initial Seeds are randomly chosen. The vertical and horizontal VKM++ perform with similar iterations number, while Figure 10 shows the average of iterations for

Seeds dataset. The results indicate that the number of iterations for variance-k-means++ is better on average than k-means and k-means++. To summarize, we analyzed four different methodologies: Traditional k-means, k-means++, variance-k-means++ (horizontal), and variance-k-means++ (vertical). Each methodology was assessed with k parameter from 2 to 20, and the outcomes were averaged over multiple trials.

As demonstrated in Tables 1 and 2, the DBI value and the iterations of VKM++, whether vertical or horizontal, exceed the performance of k-means and k-means++. A lower DBI value suggests that VKM++ provides a more resilient cluster, while fewer iterations in VKM++ reflect a more rapid convergence into stable clusters.

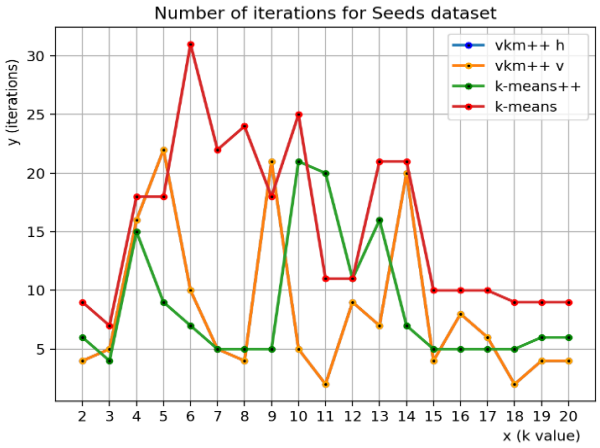

Figure 9. The number of iterations Seeds dataset

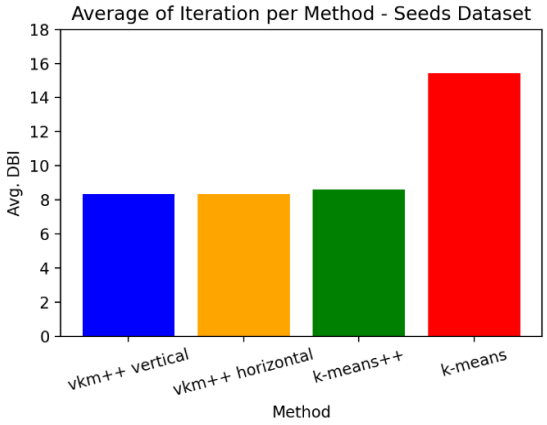

Figure 10. The average of iterations for Seeds dataset

Table 1. The average values of DBI

Method	Datasets	Avg. DBI	Avg. Iteration
VKM++ vertical	Iris	0.756	5.158
VKM++ horizontal	Iris	0.771	5.474
k-means++	Iris	0.830	8.842
k-means	Iris	0.802	9.000

Table 2. The average iterations

Method	Datasets	Avg. DBI	Avg. Iteration
VKM++ vertical	Seeds	0.612	6.421
VKM++ horizontal	Seeds	0.635	6.789
k-means++	Seeds	0.648	9.105
k-means	Seeds	0.701	8.947

4.3. Discussion

Figure 3 displays DBI values for the Iris dataset, which is initially balanced at k=2. As k increases, the proposed technique achieves a decreased DBI value at k=20. A lower DBI value suggests that the generated clusters are more cohesive. The experiment found no overlapping clusters, as all DBI values were below 1.00. The average DBI value in Figure 4 indicates that the proposed method has a lower average. In comparison to k-means++ and traditional k-means, variance-k-means++ demonstrates better cohesion. Figure 5 illustrates the dynamics of values for the number of iterations with the Iris dataset. However, the proposed method is marginally more efficient than both baseline methods. Figure 6 also illustrates that, on average, variance-k-means++ iterations, both vertical and horizontal, are more efficient than k-means++ and k-means.

When utilizing Seeds, its performance exhibits similarities to the Iris dataset. Figures 7 and 8 depict the performance using Seeds dataset. In this experiment, the performance of the proposed method is almost similar to k-means++, while k-means is not as good as others. The number of iterations of the proposed method outperforms the baseline methods, as shown in Figures 9 and 10. From the experiment result, variance-k-means++ shows better performance in the number of iterations compared to k-means++ and k-means. The result reflects variance-k-means++, which this method is created to stabilize the number of iterations. By controlling the initial centroid selection, the cluster becomes stable, and the number of iterations is minimized. However, the research is conducted only for two-dimensional data. It is proven that for low-dimensional data, the method has worked well. The challenge of this study is when it faces high-dimensional data. Therefore, future study is very important to investigate the high-dimensional data on this variance-k-means++.

VKM++ offers enhanced clustering performance due to its deterministic and dispersion-aware initialization approach. In contrast to k-means and k-means++, which depend on random or probabilistic seeding, VKM++ calculates the global mean and positions pseudo-centroids at half-variance. This method guarantees that the initial Seeds are spread across areas that reflect the natural distribution of the data instead of being randomly allocated.

The DBI evaluates cluster quality by considering intra-cluster cohesion and inter-cluster separation. By situating centroids near the extremes of dispersion, VKM++ minimizes cluster overlap and enhances the separation between clusters. As a result, the clusters become more compact internally and more distinct from one another, leading to lower DBI values when compared to k-means and k-means++.

The impact on convergence is associated with the optimal positioning of centroids. Since VKM++ begins with centroids that are closer to their ideal locations, fewer modifications are required during the iterative updates. This reduces oscillations in the assignments of data points and speeds up convergence. Empirical evidence demonstrates that VKM++ necessitates considerably fewer iterations than traditional methods, thereby decreasing computational time while preserving or enhancing clustering quality.

A systematic comparison of the Iris and Seeds datasets reveals consistent performance trends for the proposed method, particularly in terms of clustering quality and efficiency. In all datasets, variance-k-means++ consistently achieves lower DBI values compared to random initialization and standard k-means++, which signifies more compact clusters and improved inter-cluster separation. Furthermore, the proposed method displays faster convergence, requiring fewer iterations to reach stable solutions. However, the level of improvement varies across datasets. These cross-dataset trends indicate that the proposed method generalizes well across various data characteristics while also revealing dataset-dependent behaviors that affect the relative performance gains.

The results of our study are consistent with and further develop prior research on centroid initialization strategies within k-means clustering. Probabilistic seeding techniques, such as k-means++, have illustrated that distance-aware initialization significantly bolsters clustering stability by encouraging well-separated initial centroids. Similarly, the literature on distance-based and deterministic initialization methods stresses the significance of structured centroid placement to lessen sensitivity to random initializations and poor local minima.

While the results are promising, this study has several limitations that need to be acknowledged. The proposed method has been evaluated mainly on low-dimensional datasets, where distance metrics and variance estimates are highly interpretable and can be visually verified. In higher-dimensional spaces, data often exhibit sparsity and strong correlations among features, which can compromise the effectiveness of variance-based centroid placement. In these scenarios, the direct application of marginal variances may not adequately capture the true geometry of the data. Therefore, preprocessing techniques such as PCA may be necessary to ensure stable and meaningful initialization. These considerations will be viewed as important directions for future research to validate the robustness and scalability of the proposed approach in more complex data environments.

5. CONCLUSION

This study proposed a new algorithm to determine initial centroid in k-means. Our investigation is focused on determining the first of two centroids and ensuring that the initial centroid is not selected randomly. The features are limited to low-dimensional data for initial study. The starting centroid is chosen based on the midpoint of the data, and the distance is measured both vertically and horizontally using the variance values. The clustering process then follows the k-means++ algorithm. We evaluated the results using the DBI and the number of iterations for convergence analysis. The aim is to reduce the DBI and minimize the number of iterations. The findings demonstrate that both the DBI and the number of iterations, on average, exhibit a decline, indicating that the variance-k-means++ algorithm runs as expected. For Iris dataset, the DBI value is 0.756 for vertical VKM++ and 0.771 for horizontal, while k-means is 0.802 and k-means++ is 0.830. Whereas the average of the iteration number decreases significantly. Vertical VKM++ has iteration average of 5.158, horizontal VKM++ as 5.474. They are better than k-means 9.00 and k-means++ 8.842. Better performance can lead to this method in several real-world applications, such as anomaly detection, image clustering, and document embedding clustering.

However, the current study has limitations, such as handling multi-dimensional data and the lack of evaluation on streaming data. Future research may focus on investigating multi-dimensional data and applying this method to text data. Additionally, further evaluation could be conducted using the elbow method and the silhouette coefficient.

ACKNOWLEDGMENTS

The authors express their heartfelt thanks to the Faculty of Engineering, Universitas Negeri Jakarta, for providing research funding under the Faculty of Engineering Grant for Fundamental Research.

FUNDING INFORMATION

This research is supported for granted by Faculty of Engineering, Universitas Negeri Jakarta.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Widodo	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓		
Jiel Vayyad Ramadhan	✓		✓		✓	✓	✓	✓			✓			
Muhammad Ficky Duskarnaen	✓	✓		✓	✓					✓		✓		
Via Tuhamah Fauziastuti				✓		✓		✓		✓			✓	✓
Chelsea Zaomi Pondayu		✓	✓		✓		✓				✓		✓	
Mada Rekadarma Septianda			✓			✓	✓				✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] F. Nie, Z. Li, R. Wang, and X. Li, "An effective and efficient algorithm for k-means clustering with new formulation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3433–3443, 2023, doi: 10.1109/TKDE.2022.3155450.

[2] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002, doi: 10.1109/TPAMI.2002.1017616.

[3] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics (Switzerland)*, vol. 9, no. 8, pp. 1–12, 2020, doi: 10.3390/electronics9081295.

[4] P. Wang, X. Yang, W. Ding, J. Zhan, and Y. Yao, "Three-way clustering: Foundations, survey and challenges," *Applied Soft Computing*, vol. 151, 2024, doi: 10.1016/j.asoc.2023.111131.

[5] R. Duwairi and M. Abu-Rahmeh, "A novel approach for initializing the spherical K-means clustering algorithm," *Simulation Modelling Practice and Theory*, vol. 54, pp. 49–63, 2015, doi: 10.1016/j.simpat.2015.03.007.

[6] S. Bandyapadhyay, F. V Fomin, P. A. Golovach, W. Lochet, N. Purohit, and K. Simonov, "How to find a good explanation for clustering?," *Artificial Intelligence*, vol. 322, 2023, doi: 10.1016/j.artint.2023.103948.

[7] A. A. Khan, M. S. Bashir, A. Batool, M. S. Raza, and M. A. Bashir, "K-means centroids initialization based on differentiation between instances attributes," *International Journal of Intelligent Systems*, vol. 2024, no. 1, 2024, doi: 10.1155/2024/7086878.

[8] V. V Romanuke, "Random centroid initialization for improving centroid-based clustering," *Decision Making: Applications in Management and Engineering*, vol. 6, no. 2, pp. 734–746, 2023, doi: 10.31181/dmame622023742.

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

[10] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, vol. 07-09-January-2007, pp. 1027–1035.

[11] E. Hassan *et al.*, "A hybrid k-means++ and particle swarm optimization approach for enhanced document clustering," *IEEE Access*, vol. 13, pp. 48818–48840, 2025, doi: 10.1109/ACCESS.2025.3535226.

[12] V. A. Puligandla and S. Loncaric, "A supervoxel segmentation method with adaptive centroid initialization for pointclouds," *IEEE Access*, vol. 10, pp. 98525–98534, 2022, doi: 10.1109/ACCESS.2022.3206387.

[13] H. Wang and J. Mi, "Intuitive-K-prototypes: A mixed data clustering algorithm with intuitionistic distribution centroid," *Pattern Recognition*, vol. 158, 2025, doi: 10.1016/j.patcog.2024.111062.

[14] S. Parveen and M. S. Yang, "Lasso-based k-Means++ clustering," *Electronics (Switzerland)*, vol. 14, no. 7, 2025, doi: 10.3390/electronics14071429.

[15] Y. Ping, H. Li, B. Hao, and C. Guo, "Beyond k-Means++: Towards better cluster exploration with geometrical information," *Pattern Recognition*, vol. 146, 2024, doi: 10.1016/j.patcog.2023.110036.

[16] A. A. Abdulnassar and L. R. Nair, "Performance analysis of Kmeans with modified initial centroid selection algorithms and developed Kmeans9+ model," *Measurement: Sensors*, vol. 25, 2023, doi: 10.1016/j.measen.2023.100666.

[17] T. S. Priyadarshini and M. A. Hameed, "Developing heart stroke prediction model using deep learning with combination of fixed row initial centroid method with navie Bayes, decision tree, and artificial neural network," *Measurement: Sensors*, vol. 34, p. 101237, 2024, doi: 10.1016/j.measen.2024.101237.

[18] N. Bajpai, J. H. Paik, and S. Sarkar, "Balanced seed selection for K-means clustering with determinantal point process," *Pattern Recognition*, vol. 164, 2025, doi: 10.1016/j.patcog.2025.111548.

[19] Q. Zhou and B. Sun, "Adaptive K-means clustering based under-sampling methods to solve the class imbalance problem," *Data and Information Management*, vol. 8, no. 3, 2024, doi: 10.1016/j.dim.2023.100064.

[20] Y. Tian, X. Niu, and J. Chai, "Partial label learning via weighted centroid clustering disambiguation," *Neurocomputing*, vol. 604, 2024, doi: 10.1016/j.neucom.2024.128312.

[21] A. Karim, C. Loqman, Y. Hami, and J. Boumhidi, "Max stable set problem to found the initial centroids in clustering problem," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, pp. 569–579, 2022, doi: 10.11591/ijeecs.v25.i1.pp569-579.

[22] F. Faezy Razi, "A hybrid DEA-based K-means and invasive weed optimization for facility location problem," *Journal of Industrial Engineering International*, vol. 15, no. 3, pp. 499–511, 2019, doi: 10.1007/s40092-018-0283-5.

[23] W. Abdulelah Qasim and B. Ahmed Mitras, "A hybrid algorithm based on invasive weed optimization algorithm and grey wolf optimization algorithm," *International Journal of Artificial Intelligence & Applications*, vol. 11, no. 1, pp. 31–44, 2020, doi: 10.5121/ijaia.2020.11103.

[24] Y. Liu, J. C. Chai, X. Cui, W. Yan, N. Li, and L. Jin, "Multi-objective optimization of air dehumidification membrane module based on response surface method and genetic algorithm," *Energy Reports*, vol. 9, pp. 2201–2212, 2023, doi: 10.1016/j.egyr.2023.01.036.

[25] J. S. Keek, S. L. Loh, Y. C. Wong, X. J. Woo, and W. W. Lee, "Genetic algorithms and particle swarm optimization for interference minimization in mobile network channel assignment problem," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 276–288, 2021, doi: 10.22266/ijies2021.0831.25.

[26] N. L. G. P. Suwirmayanti, I. K. G. D. Putra, M. Sudarma, I. M. Sukarsa, and E. Setyaningsih, "IWOKM-GA hybrid method to improve clustering accuracy in banking data," *International Journal of Computing and Digital Systems*, vol. 18, no. 1, pp. 1–11, 2025, doi: 10.12785/ijcds/1571110934.

[27] L. Nigro and F. Cicirelli, "Improving clustering accuracy of k-means and random swap by an evolutionary technique based on careful seeding," *Algorithms*, vol. 16, no. 12, 2023, doi: 10.3390/a16120572.

[28] J. Liao, X. Wu, Y. Wu, and J. Shu, "K-NNDP: K-means algorithm based on nearest neighbor density peak optimization and outlier removal," *Knowledge-Based Systems*, vol. 294, 2024, doi: 10.1016/j.knosys.2024.111742.

[29] H. Kim, H. K. Kim, and S. Cho, "Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling," *Expert Systems with Applications*, vol. 150, 2020, doi: 10.1016/j.eswa.2020.113288.

[30] M. Zubair, M. D. A. Iqbal, A. Shil, M. J. M. Chowdhury, M. A. Moni, and I. H. Sarker, "An improved k-means clustering algorithm towards an efficient data-driven modeling," *Annals of Data Science*, vol. 11, no. 5, pp. 1525–1544, 2024, doi: 10.1007/s40745-022-00428-2.

[31] S. D. Saldarriaga-Zuluaga, J. M. López-Lezama, and N. Muñoz-Galeano, "Optimal coordination of over-current relays in microgrids using principal component analysis and k-means," *Applied Sciences (Switzerland)*, vol. 11, no. 17, 2021, doi: 10.3390/app11177963.

[32] S. A. Sheikh, M. I. Mir, O. A. Alamri, and J. G. Dar, "On variance and average moduli of zeros and critical points of polynomials," *Symmetry*, vol. 16, no. 3, 2024, doi: 10.3390/sym16030349.
[33] D. Cheng, J. Huang, S. Zhang, S. Xia, G. Wang, and J. Xie, "K-means clustering with natural density peaks for discovering arbitrary-shaped clusters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 11077–11090, 2024, doi: 10.1109/TNNLS.2023.3248064.
[34] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi: 10.1109/tpami.1979.4766909.
[35] F. Du, S. Wu, Z. Tian, F. Qiu, and C. Xu, "An improved prototype network and data augmentation algorithm for few-shot structural health monitoring using guided waves," *IEEE Sensors Journal*, vol. 23, no. 8, 2023, doi: 10.1109/JSEN.2023.3257366.
[36] S. Zhao, R. Yang, F. Meng, and L. Cai, "Optimizing cluster centroids with improved quadratic interpolation: an Adaptive K-means algorithm," *Journal of Computational and Applied Mathematics*, vol. 473, 2026, doi: 10.1016/j.cam.2025.116921.

BIOGRAPHIES OF AUTHORS

Widodo is an assistant professor at the Department of Informatics and Computer Engineering Education, Universitas Negeri Jakarta. He earned a PhD in computer science from the Faculty of Computer Science, Universitas Indonesia, in 2020, an MSc in computer science from Universitas Indonesia in 2004, and a BSc in information systems from Gunadarma University in 1999. His research interests include machine learning, natural language processing, and privacy-preserving data publishing. Currently, he is involved in machine learning and NLP research group and researching the optimization of clustering algorithms. He can be contacted via email at widodo@unj.ac.id.

Jiel Vayyad Ramadhan received his first degree in informatics and computer engineering education from Universitas Negeri Jakarta, Indonesia, in 2025. He graduated as one of the top students in his program and is currently working as a Python tutor at Algorithmics. His main research interests include machine learning, clustering, computer vision, data mining, and artificial intelligence. He can be contacted at email: jielvayad261102@gmail.com.

Muhammad Ficky Duskarnaen is an assistant professor at the Department of Informatics and Computer Engineering Education at Universitas Negeri Jakarta (UNJ). He holds a master's degree in computer science and is actively involved in academic leadership, curriculum development, and student supervision, particularly in final projects and internship seminars. His research interests include network infrastructure, wireless communication, and educational technology, as reflected in his guidance of student projects applying methodologies like the network development life cycle (NDLC). He can be contacted at duskarnaen@unj.ac.id.

Via Tuhamah Fauziastuti earned her Master of Education from the University of Pendidikan Sultan Idris Malaysia and her bachelor's degree from the Department of Information Systems at Universitas Islam Negeri (UIN) Syarif Hidayatullah Jakarta. Currently, she is an assistant professor at the Department of Informatics and Computer Engineering Education. Her research interests are information technology education and software engineering. She can be contacted at viatuhamah@unj.ac.id.

Chelsea Zaomi Pondayu is a Bachelor of Education in computer science from Informatics and Computer Engineering Education at Universitas Negeri Jakarta (2025), with a focus on AI, web development and educational technology. Her passion for AI innovation led her to become a finalist in the *2023 Outstanding Student Competition* at FT-UNJ, proposing an AI-based cancer diagnosis tool. Her undergraduate research implements convolutional neural networks (CNN) to classify dyslexia traits in handwriting, bridging AI and inclusive education. She can be contacted at chelseazaomi9@gmail.com.

Mada Rekadarma Septianda is a Bachelor of Education in computer science from the Department of Informatics and Computer Engineering Education at Universitas Negeri Jakarta. His research focuses on text processing, especially on text clustering and topic modeling. Currently, He is a member of machine learning and NLP Research Group. He can be contacted at mada.septianda@gmail.com.