The Academic
International Journal of Multidisciplinary Research

An Online Peer Reviewed / Refereed Journal
Volume 4 | Special Issue 3 | April 2026
ISSN: 2583-973X (Online)
Website: www.theacademic.in

An Efficient Single-Pass Clustering Method Using Distance Thresholding for Real-Time Data Grouping

Binayak Adhikari

M.Tech Scholar Dr. K. N. Modi University Newai, Rajasthan, India binayakadhikari067@gmail.com

Jyoti Sehrawat

Assistant Professor Dr. K. N. Modi University Newai, Rajasthan, India jyotisehrawat.cse@dknmu.org

DOI : https://doi.org/10.5281/zenodo.20079227

ARTICLE DETAILS

Research Paper

Accepted: 03-04-2026

Published: 25-04-2026

Keywords:

Clustering, Single-Pass Algorithm, Distance Thresholding, Real-Time Processing, Unsupervised Learning.

ABSTRACT

Clustering is a widely used technique for grouping similar data points in many real-world applications, such as customer analysis, pattern recognition, and real-time monitoring systems. Most traditional clustering methods, including K-means, rely on iterative processes and repeated updates of cluster centers, which increase computational cost and limit their usability in time-sensitive environments. In this work, we present a simple and efficient single-pass clustering method based on distance thresholding. The proposed approach group's data points by comparing their distances to existing clusters and assigning them based on a predefined threshold, eliminating the need for multiple iterations or complex optimization steps. This makes the method lightweight and suitable for scenarios where quick decisions are required. The performance of the proposed method is evaluated using a standard numerical dataset for customer segmentation. The results show that the method is able to form meaningful clusters with significantly reduced processing time compared to traditional approaches. Although the method depends on an appropriate threshold selection, it offers a practical balance between simplicity and performance. Overall, the proposed approach provides a fast and easy-to-implement alternative for real-time clustering tasks, especially in

systems with limited computational resources

INTRODUCTION

Clustering is one of the most important techniques in data analysis and machine learning. It is used to group similar data points together when there is no labeled information available. In simple terms, clustering helps in understanding the structure of data by dividing it into meaningful groups. This technique is widely used in many real-world applications such as customer segmentation, fraud detection, medical data analysis, image processing, and recommendation systems [1] [2].

In recent years, the amount of data generated in different systems has increased rapidly. Because of this, there is a growing need for fast and efficient data processing methods. Clustering plays an important role in handling such data, especially when we need to find patterns quickly. However, many traditional clustering algorithms are not designed for real-time environments, where speed and efficiency are very important [3]. One of the most popular clustering algorithms is K-means. It is simple, easy to implement, and works well in many situations. The algorithm divides the data into a fixed number of clusters and updates the cluster centers iteratively until convergence [4].

Although K-means is widely used, it has some clear drawbacks. It requires the number of clusters to be known in advance, which is not always possible in real world problems. Also, it depends on multiple iterations to improve the clustering result, which increases computation time and makes it less suitable for large datasets or real-time applications [5].

Another important issue with iterative clustering methods is that they consume more computational resources. In systems such as Internet of Things (IoT), UAV networks, and edge computing, devices often have limited processing power. In such cases, running complex and iterative algorithms can cause delays and reduce system performance. Therefore, there is a strong need for a clustering method that is simple, fast, and does not depend on repeated calculations [6].

To address these challenges, this paper proposes a single pass clustering method based on distance thresholding. The proposed method processes each data point only once and makes clustering decisions immediately. The idea is straight forward: when a new data point arrives, its distance from existing clusters is calculated. If the distance is smaller than a predefined threshold, the point is added to that cluster. Otherwise, a new cluster is created. This process continues until all data points are assigned. One of the key advantages of this approach is that it removes the need for iterative centroid updates. This

significantly reduces execution time and computational cost. In addition, the proposed method does not require the number of clusters to be predefined. Instead, the number of clusters is automatically determined based on the distance relationships among data points. This makes the method more flexible and suitable for different types of datasets. The proposed method is especially useful in applications where fast decision-making is required. For example, in real time monitoring systems, online data processing, and UAV based networks, quick grouping of data is necessary for efficient operation [7].

The simplicity of the proposed method also makes it easy to implement and deploy in practical systems. To evaluate the performance of the proposed approach, experiments are conducted using a standard customer segmentation dataset. The results are compared with the K means algorithm using different performance metrics such as execution time, Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Score [8].

The experimental results show that the proposed method achieves much faster execution while maintaining competitive clustering quality.

The main contributions of this work can be summarized as follows:

• A simple and efficient single-pass clustering method without iterative processing

• A distance threshold-based approach for automatic cluster formation

• A computationally lightweight solution suitable for real time applications

• A detailed comparison with K-means showing improved execution speed and competitive clustering performance

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the proposed methodology. Section 4 discusses the experimental results and analysis. Finally, Section 5 concludes the paper and suggests future work.

RELATED WORK

Clustering has been widely studied in the field of machine learning and data analysis. Over the years, many clustering techniques have been proposed, each with its own advantages and limitations. Among these methods, partition-based, hierarchical, and density-based clustering approaches are the most commonly used [1] [2].

One of the earliest and most popular clustering algorithms is K-means, introduced by MacQueen [3]. It partitions data into a predefined number of clusters by minimizing the distance between data points and their corresponding cluster centroids. Due to its simplicity and efficiency, K-means has been widely applied in many domains. However, it requires the number of clusters to be specified in advance and depends on iterative updates, which can increase computational cost and affect performance in large-scale datasets [4].

Hierarchical clustering is another well-known approach, which builds a tree-like structure of clusters using either agglomerative or divisive methods [5]. Although it provides better interpretability and does not require a fixed number of clusters initially, it suffers from high computational complexity and is not suitable for large datasets. Density-based methods such as DBSCAN have also been proposed to address some limitations of traditional clustering algorithms [6] [9].

These methods group data points based on density and can identify arbitrary-shaped clusters as well as noise points. However, their performance depends heavily on parameter selection, such as neighborhood radius and minimum points, which can be difficult to tune for different datasets. In recent years, several studies have focused on improving clustering efficiency for real-time and large-scale applications. Lightweight clustering techniques and approximate methods have been proposed to reduce computational overhead and improve processing speed [7].

These approaches aim to mini mize the number of iterations or simplify distance calculations, making them more suitable for dynamic environments [10]. Despite these advancements, most existing clustering algorithms still rely on iterative optimization or complex parameter tuning. This makes them less suitable for applications that require fast decision-making and low computational cost. In contrast, the method proposed in this paper focuses on a single-pass clustering approach, where data points are processed sequentially without iterative refinement. This helps in achieving faster execution while maintaining meaningful clustering performance. The proposed approach differs from traditional methods by eliminating the need for predefined cluster count and iterative updates. Instead, it uses a distance threshold-based mechanism to form clusters dynamically. This makes the method simple, efficient, and suitable for real-time applications, addressing the limitations of existing clustering techniques.

PROPOSED METHODOLOGY

This paper presents a fast and efficient single-pass clustering framework based on distance thresholding. The proposed approach is designed to eliminate iterative optimization while maintaining effective

grouping of data points. Unlike conventional clustering methods such as K-means, which require repeated centroid updates and convergence checks, the proposed method processes each data point only once in a sequential manner. This significantly reduces computational overhead and enables its applicability in real-time and resource-constrained environments.

Let the dataset be represented as:

$$X = \{x_1, x_2, x_3, ..., x_n\} \quad (1)$$

Where each data point is defined as:

$$x_i = (x_{i1}, x_{i2}, ..., x_{im}) \quad (2)$$

With n representing the total number of data points and m denoting the number of features in the dataset. To ensure uniform contribution of all features during distance computation, Min-Max normalization is applied to transform the data into a common scale:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3)$$

This transformation ensures that each feature contributes equally to the distance measure and prevents bias caused by varying feature magnitudes. Following normalization, the similarity between data points is computed using the Euclidean distance metric:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \quad (4)$$

This distance function provides a quantitative measure of similarity, where smaller values indicate higher similarity between data points. To guide the clustering process, a global distance threshold is defined as:

$$T = \alpha \cdot \text{mean}(D) \quad (5)$$

Where D represents the set of all pairwise distances in the dataset and α is a scaling factor that controls cluster compactness. The value of α plays an important role in determining the number and size of clusters. Lower values of α result in tighter clusters, while higher values allow broader grouping.

The clustering process is performed in a sequential manner. Initially, the first data point forms the first cluster. For each subsequent data point xi, its distance to the centroid of each existing cluster Cj is computed as:

$$d(x_i, C_j) = \|x_i - \mu_j\| \tag{6}$$

Where μj denotes the centroid of cluster Cj, defined as the mean of all data points assigned to that cluster. The centroid is updated incrementally as new points are added, ensuring that it reflects the current cluster structure.

The minimum distance across all clusters is then determined as:

$$d_{min} = \min_j d(x_i, C_j) \tag{7}$$

If dmin ≤ T, the data point is assigned to the nearest cluster. Otherwise, a new cluster is created and initialized with xi. This decision rule enables adaptive and data-driven cluster formation without requiring prior knowledge of the number of clusters. Since each data point is processed only once and no iterative refinement is performed, the proposed framework avoids repeated computations. This significantly improves execution speed while maintaining meaningful clustering performance. The absence of iterative updates also makes the method more stable and predictable in dynamic environments.

The computational complexity of the proposed method can be expressed as:

$$O(n \cdot k) \tag{8}$$

Where n is the number of data points and k is the number of clusters formed during execution. This complexity arises from comparing each data point with the centroids of existing clusters

A) **Algorithm**

Algorithm 1 Single-Pass Distance-Based Clustering

Require: Dataset $X = \{x_1, x_2, ..., x_n\}$, Threshold T
Ensure: Cluster labels

```
Normalize dataset X
Initialize empty cluster set C ← ∅
for each data point x_i ∈ X do
    if C is empty then
        Create new cluster C_1
        Assign x_i → C_1
    else
        for each cluster C_j ∈ C do
            Compute distance d(x_i, μ_j)
        end for
        d_min ← min_j d(x_i, μ_j)
        if d_min ≤ T then
            Assign x_i → C_j
            Update centroid μ_j
        else
            Create new cluster C_new
            Assign x_i → C_new
        end if
    end if
end for
return cluster labels
```

RESULT AND DISCUSSION

A) Simulation Setup

The performance of the proposed single-pass clustering method is evaluated using the Mall Customer Segmentation dataset. The dataset consists of customer records with features such as Annual Income and Spending Score, which are commonly used for clustering-based analysis. Prior to clustering, the dataset is normalized using Min-Max scaling to ensure uniform contribution of features during distance computation.

All experiments are conducted using Python with standard scientific libraries. The proposed method is compared against the K-means clustering algorithm under the same conditions to ensure a fair evaluation. Both methods are applied to the same feature space, and the number of clusters in K-means is set equal to the number of clusters generated by the proposed method.

The evaluation is performed using multiple performance metrics, including execution time, Silhouette Score, Davies Bouldin Index (DB Index), and Calinski-Harabasz (CH) Score. These metrics provide a comprehensive assessment of clustering quality in terms of separation, compactness, and overall structure.

B) Quantitative Results

Table I presents the comparative results of the proposed method and K-means.

TABLE I
PERFORMANCE COMPARISON OF PROPOSED METHOD AND K-MEANS

Metric	Proposed	K-means
Clusters	7	7
Execution Time (s)	0.005935	0.024453
Silhouette Score	0.4917	0.4808
DB Index (↓)	0.5443	0.7718
CH Score (↑)	177.68	222.56

The results show that the proposed method achieves significantly lower execution time compared to K-means. Specifically, the proposed approach requires only 0.005935 seconds, while K-means takes 0.024453 seconds. This demonstrates approximately four times faster performance, which is mainly due to the elimination of iterative centroid updates.

In terms of clustering quality, the proposed method achieves a higher Silhouette Score of 0.4917 compared to 0.4808 obtained by K-means. This indicates that the clusters formed by the proposed method have better separation between groups. Additionally, the Davies-Bouldin Index of the proposed method is significantly lower (0.5443) than that of K means (0.7718), suggesting that the clusters are more compact and well-defined.

Although K-means achieves a higher Calinski-Harabasz score, this is expected due to its iterative optimization mechanism, which refines cluster centroids over multiple iterations. In contrast, the proposed method achieves competitive performance without requiring such iterative refinement.

C) Graphical Analysis

Fig. 1 illustrates the clustering results obtained using both methods. The proposed method produces clearly separated clusters with minimal overlap, indicating effective grouping of data points. K-means also forms distinct clusters; however, slight variations in cluster boundaries can be observed due to centroid optimization.

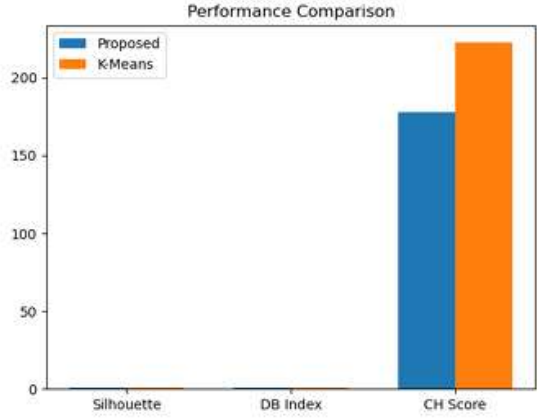

Fig. 1. Clustering results of Proposed Method and K-means

Fig. 2 presents the execution time comparison. It is evident that the proposed method significantly outperforms K-means in terms of speed. The absence of iterative updates allows the proposed method to process data in a single pass, making it highly efficient.

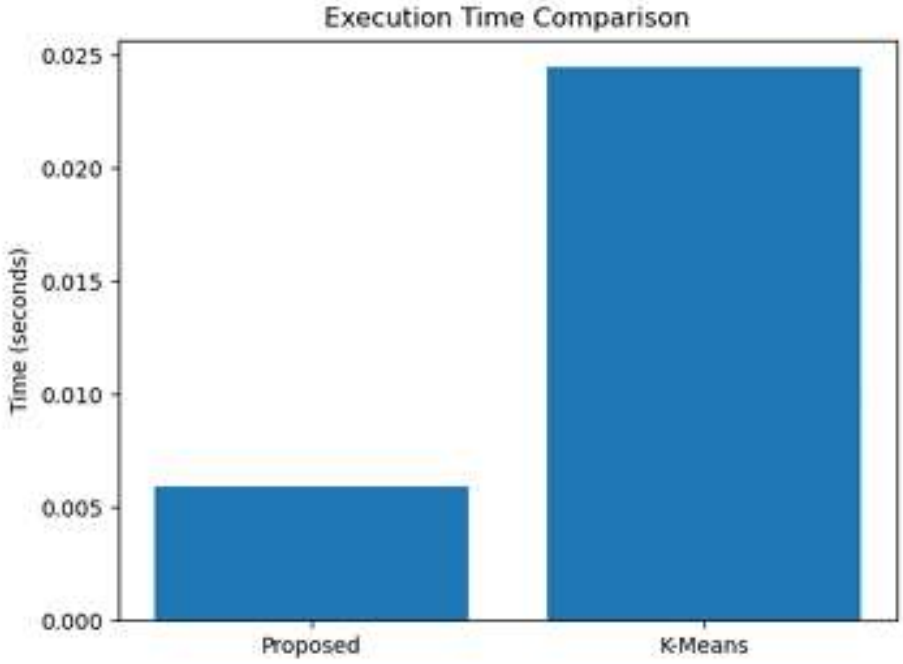

Fig. 2. Execution Time Comparison

Fig. 3 shows the comparison of clustering performance metrics. The proposed method achieves better Silhouette Score and lower DB Index, indicating improved cluster separation and compactness. While K-means achieves a higher CH Score, the overall performance of the proposed method remains competitive.

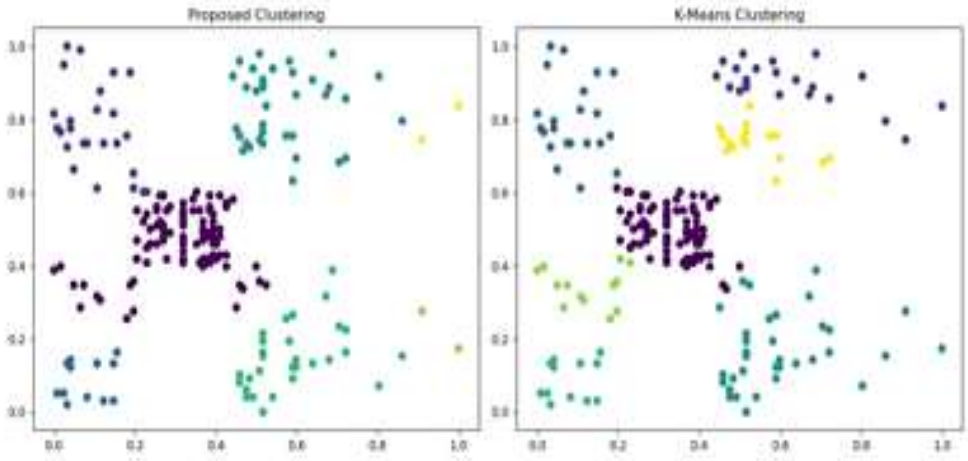

Fig. 3. Performance Metrics Comparison

D) Discussion

The experimental results demonstrate that the proposed single-pass clustering method provides a strong balance between computational efficiency and clustering quality. The elimination of iterative processing significantly reduces execution time, making the method suitable for real-time applications. Furthermore, the proposed approach achieves better separation and compactness of clusters, as indicated by Silhouette Score and DB Index. This suggests that the distance threshold based mechanism effectively captures the inherent structure of the data. Although K-means performs better in terms of CH Score, its reliance on iterative refinement increases computational cost. In contrast, the proposed method achieves competitive results without iteration, which highlights its efficiency and practicality Overall, the results confirm that the proposed method is a viable alternative to traditional clustering algorithms, particularly in scenarios where fast processing and low computational overhead are critical.

CONCLUSION AND FUTURE WORK

This paper presented a fast and efficient single-pass clustering framework based on distance thresholding. The proposed method eliminates the need for iterative optimization by processing each data point only once, thereby significantly reducing computational overhead. Unlike traditional clustering algorithms such as K-means, which rely on repeated centroid updates, the proposed approach performs clustering in a sequential manner while dynamically determining the number of clusters. The experimental results demonstrate that the proposed method achieves substantially lower execution time compared to K-means, making it highly suitable for real-time and resource-constrained applications. In addition, the proposed approach provides competitive clustering performance, achieving higher Silhouette Score and lower Davies-Bouldin Index, which indicate better cluster separation and compact ness. Although K-means achieves a higher Calinski-Harabasz score due to its iterative refinement process, the proposed method maintains a strong balance between efficiency and clustering quality.

Overall, the results confirm that the proposed framework is an effective alternative to conventional clustering techniques, particularly in scenarios where fast processing and low computational cost are critical. Despite its advantages, the proposed method has certain limitations. The clustering performance is influenced by the selection of the distance threshold parameter, which may require tuning for different datasets. Additionally, the current approach is primarily evaluated on low-dimensional data and may face challenges in handling high-dimensional feature spaces.

Future work can focus on developing adaptive threshold selection mechanisms to further improve clustering robustness. The extension of the proposed method to high-dimensional data and large-scale

datasets is another important direction. Furthermore, integrating the proposed framework with real time systems such as IoT and UAV networks can enhance its practical applicability. Incorporating advanced distance measures or hybrid techniques may also improve clustering performance in complex data scenarios.

REFERENCES:

- Z. Xing and W. Zhao, "K-means clustering: A review of the past 70 years," Available at SSRN 5842722, 2025.
- P. Koukaras and C. Tjortjis, "Data preprocessing and feature engineering for data mining: Techniques, tools, and best practices." AI, vol. 6, no. 10, 2025.
- A. Mukherji, A. Mondal, R. Banerjee, and S. Mallik, "Recent landscape of deep learning intervention and consecutive clustering on biomedical diagnosis," in Artificial Intelligence and Applications, vol. 3, no. 4, 2025, pp. 359–377.
- K. Backhaus, B. Erichson, S. Gensler, R. Weiber, T. Weiber et al., Multivariate analysis. Springer, 2025.
- C. Ang, S. Kim, and M. Pilanci, "Optimal scalar quantization for matrix multiplication: Closed-form density and phase transition," arXiv preprint arXiv: 2603.19559, 2026.
- M. Shahnawaz and M. Kumar, "A comprehensive survey on big data an alytics: Characteristics, tools and techniques," ACM Computing Surveys, vol. 57, no. 8, pp. 1–33, 2025.
- L. A. L. Da Costa, R. C. De Lamare, R. Kunst, and E. P. De Freitas, "Cluster-based machine learning-driven routing for uav networks in 6g environment," IEEE Access, 2025.
- P. W. Hodges, R. Sanchez, S. Pritchard, A. Turnbull, A. Hahne, and J. Ford, "Toward validation of clinical measures to discriminate between nociceptive, neuropathic, and nociplastic pain: cluster analysis of a cohort with chronic musculoskeletal pain," The Clinical Journal of Pain, vol. 41, no. 5, p. e1281, 2025.
- L. Duponchel, R. Rocha de Oliveira, and V. Motto-Ros, "Large language models (such as chatgpt) as tools for machine learning-based data insights in analytical chemistry," Analytical Chemistry, vol. 97, no. 13, pp. 6956–6961, 2025.
- Z. Jamalpour, S. Ghaderi, and M. Fathian-Kolahkaj, "High-risk patient profiles for ovarian cancer: A new approach using cluster analysis of tumor markers," Journal of Gynecology Obstetrics and Human Reproduction, vol. 54, no. 2, p. 102888, 2025