iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67721

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Enhanced Grid Based K-Medoids Clustering Technique for Customer Segmentation

M. Deepthi[1], Dr. K. Venkataramana[2]
[1, 2]*Department of MCA, KMMIPS, Tirupati, A.P, India*

Abstract: ***Among the existing clustering algorithms, the K-Medoids algorithm is widely used due to its robustness and effectiveness. However, its performance can be affected by the selection of initial cluster centers and sensitivity to noise. To address these issues, this paper proposes an improved K-Medoids clustering algorithm. A density-based noise removal technique is introduced through Grid to filter out outliers and obtain regional density. Grid-based clustering with K-Medoids improves efficiency by reducing the number of distance calculations through space partitioning. It is robust to noise and outliers since K-Medoids selects actual data points as representatives. The approach reduces computational complexity by clustering grid cells instead of all data points. Additionally, the influence of density estimation errors on centre selection is reduced through granularity-based*** *adjustments****, minimizing manual intervention in the peak density algorithm. This approach enhances clustering accuracy and stability.***

Keywords: Machine learning – Clustering - K-Means-K-Medoids - Grid-based clustering- granularity computing.

I. INTRODUCTION

Machine learning is a part of computer science that emanated from the study of pattern recognition and computational learning theory all in artificial intelligence. Algorithms are used to make predictions on data. Before now the field of machine learning was mainly algorithms and theory of optimization but recently machine learning covers several other disciplines which includes statistics, information theory, theory of algorithms, probability and functional analysis[1] . Machine learning and computational statistics are always closely related because of their specialty in prediction making and mathematical optimi-zation which brings about methods, theories and application to the field. In machine learning, strictly static pro-gram instructions are not followed, rather, algorithms are used to build a model from input which are used to make data-driven prediction or decisions.

Clustering is an unsupervised learning technique used to categorize patterns (observations, data points, or feature vectors) into distinct groups (clusters) based on similarity. It is widely applied in various domains as a fundamental step in exploratory data analysis. Despite its broad applicability, clustering remains a challenging combinatorial problem. Variations in assumptions, methodologies, and application contexts across different disciplines have slowed the transfer of universal clustering concepts and techniques [2]. Clustering is a widely used analytical technique for grouping unlabelled data to extract meaningful insights. Since no single clustering algorithm can address al clustering problems, various algorithms have been developed for diverse applications. It is defined as the process of grouping objects when there is little or no prior knowledge about their relationships in the given dataset. Clustering also aims to uncover the underlying patterns or classes present within the data. Additionally, it serves as a method for organizing unlabelled data into distinct groups with minimal or no supervision.[3].

The K-Means algorithm is a well-known method in machine learning, prized for its simplicity and efficiency, which makes it a popular choice in both academic and industrial applications. However, there are two significant drawbacks associated with the traditional K-Means clustering approach: The starting points for the clusters are chosen randomly, which significantly influences the clustering outcome. Since the next cluster centre is based on the previous one, an initial poor selection can result in slow convergence or convergence to a suboptimal solution, leading to increased computation time and less reliable results. K-Means calculates cluster centers by averaging the points in each cluster [4]. As a result, it is highly sensitive to noise or outliers. If an outlier is mistakenly assigned to a cluster, it can distort the center of the cluster, pulling it away from the actual centre of the group and negatively affecting the clustering performance.These limitations underscore the challenges of using K-Means in real-world scenarios where data may not be perfectly clean or evenly distributed[4].

K-Medoids is a clustering algorithm similar to K-Means, but instead of using the mean of data points as the cluster center, it uses actual data points (medoids) as centers. It works by selecting a set of medoids, assigning data points to the nearest medoid, and iteratively updating the medoids to minimize the sum of dissimilarities. K-Medoids is less sensitive to outliers compared to K-Means because it uses actual data points as centers.

©IJRASET: All Rights are Reserved

It is well-suited for datasets where the data points are non-spherical or when the distance metric is not Euclidean. K-Medoids is commonly used in situations where robustness to noise and outliers is important. Algorithms like Partitioning Around Medoids (PAM) are often used to efficiently find medoids in large datasets[5].

II. RELATED WORK

A cluster is a set of points where each point is closer or more similar to every other point in the cluster than to any point outside the cluster. Sometimes, a threshold is used to ensure all points in a cluster are sufficiently close or similar. However, in many datasets, a point on the edge of a cluster may be closer or more similar to points in another cluster than to points within its own cluster. Some of the techniques of type K-Means Clustering which is a centroid-based algorithm where clusters are defined by the mean of the points in the cluster[6]. It's efficient but sensitive to initial conditions and outliers.Mean Shift Clustering works by gathering each data point towards the mode (highest density of data points) iteratively, forming clusters based on the density of data points[6].

Various data analysis approaches have been documented in the literature, with the K-Means algorithm standing out as the most popular and straightforward clustering method. Its widespread use in numerous clustering applications is due to its simplicity and low computational complexity. However, the K-Means algorithm faces several challenges that impact its clustering performance. During initialization, users must specify the number of clusters in advance, and the initial cluster centers are chosen randomly[7]. This random selection can affect the algorithm's performance, especially for large datasets, making it difficult to determine the optimal number of clusters. Additionally, the algorithm's greedy nature can lead to minimal local convergence. Another limitation is the use of the Euclidean distance metric to measure similarity, which restricts the algorithm's ability to detect various cluster shapes and makes it challenging to identify overlapping clusters[7].

K-Medoids clustering is an effective technique for grouping similar data points in large datasets, especially when background knowledge is limited. Unlike K-Means, K-Medoids selects actual data points as cluster centers, making it more robust to outliers and noise. It minimizes the sum of dissimilarities, leading to better cluster head selection and reduced space complexity in overlapping clusters. K-Medoids also demonstrates improved execution time compared to K-Means. Overall, it is a more reliable choice for handling real-world, large-scale data[8].

K-Medoids clustering partitions data into meaningful groups by selecting representative medoids, making it robust to outliers. However, the conventional K-Medoids algorithm has limitations, such as requiring prior knowledge of cluster count and sensitivity to initial medoid selection [9]. To address these issues, an improved density-based K-Medoids algorithm is proposed, enhancing clustering accuracy for spatial data. This approach outperforms DBSCAN in handling circularly distributed and overlapping clusters. The refined algorithm ensures better partitioning and efficiency in large-scale data analysis [9].

III. K-MEANS ALGORITHM

The K-Means algorithm is one of the ten classical machine learning algorithms and a widely used hard clustering method. It is a prototype-based clustering approach that optimizes an objective function based on the distance between data points and cluster centroids. The algorithm iteratively updates cluster assignments by minimizing the evaluation index JJJ, which is derived from the sum of squared errors. K-Means relies on Euclidean distance as a similarity measure and aims to find the optimal partitioning of data given an initial set of cluster centers. When the dataset is densely structured, and the differences between clusters are distinct, K-Means produces highly effective clustering results[10].

1) Step 1: Read and Initialize the Dataset Given a dataset $X=\{x_1,x_2,...,x_n\}\subset R^d$ where xi represents a *d*-dimensional data point.
2) Step 2: Grid the Dataset Partition the data space into a set of non-overlapping grids $G=\{G_1,G_2,...,G_m\}$ based on a predefined resolution r*r*.
3) Step 3: Compute the Density of Each Grid and Classify It Define the density of a grid *Gi* as: D(Gi)=|Gi| where |Gi| is the number of data points in Gi, and V(Gi)) is the volume of the grid. Set a density threshold Dmin and classify grids with D(Gi)<Dmin as noise.
4) Step 4: Compute the Granularity Density For a central grid Gc, define its granularity density *GD:* D(Gj)$GD(Gc)=\sum Gj\in N(Gc)$ $D(Gj)$ where N(Gc) denotes the set of neighbouring grids of Gc.
5) Step 5: Initialize Cluster Centers Based on Granularity Density For each granularity:
    - Step 5.1: Select the grid Gmax with the highest granularity density and set its centroid as an initial cluster center: $ci=1|Gmax|\sum xj\in Gmax\ xjci=|Gmax|1\sum xj\in G$
    - Step 5.2: If the number of selected centers k′ reaches the predefined number of clusters k*k*, proceed to Step 6.

- Step 5.3: Otherwise:
  - Mark Gmax as processed.
  - Set the density D(Gmax)=0
  - Recalculate granularity density GD(Gc).
  - Return to Step 5.1.

6) Step 6: Apply K-Means Clustering Use the selected initial points C={c1,c2,...,ck}} to perform K-means clustering:
1. Assignment Step: Assign each data point xi*xi* to the nearest cluster: Cj={xi∈X|‖xi−cj‖2≤‖xi−cl‖2,∀l≠j}
2. Update Step: Recalculate the centroid of each cluster: cj=1|Cj|∑xi∈Cjxi
3. Repeat until convergence.

7) Step 7: Output the Results Return the final cluster assignments {C1,C2,...,Ck} and cluster centroids {c1,c2,...,ck}.

IV. PROPOSED ENHANCED K-MEDIODS ALGORITHM

Grid-based clustering is a method that partitions the data space into a finite number of grid cells and performs clustering based on the density distribution of data points. The K-Medoids algorithm, a more robust alternative to K-Means, is often integrated with grid-based clustering to improve clustering stability and reduce sensitivity to noise and outliers.

A. Steps

Grid-Based K-Medoids Clustering Algorithm

1) Step 1: Read and Initialize the DatasetDefine the dataset as: D = { d_1, d_2, …., d_n } where each data item did i has a feature vector: di=(fi1,fi2,…,fim)∈Rm d_i = (f{i1}, f{i2}, ……, f{im}) \in {R}^{m} for i=1,2,…,n_i = 1, 2, …., n, and mm is the number of features.

2) Step 2: Grid the DatasetDivide the data space into a 4×44 \times 4 grid, forming a set of grid cells: G={G_1,G_2,…,G_p},p=4×4=16 G = { G_1, G_2,….., Gp}, p = 4 x 4 = 16 Each grid cell GjGj contains a subset of data points: Gj={di|di falls within the boundaries of Gj}Gj = for j=1,2,…,16.

3) Step 3: Calculate Grid Density and Remove NoiseThe density of each grid cell Gj is defined as: D(Gj)=|Gj| where |Gj| is the number of data points in Gj. Define a density threshold Dmin and classify grids as noise if: D(Gj)<Dmin Remove such noisy grids from further processing.

4) Step 4: Calculate Granular Density for Each Grid Area For each grid cell Gc, compute its granular density by summing the densities of itself and its adjacent grids: GD(Gc)=∑Gj ∈ N(Gc), D(Gj) where N(Gc) is the set of adjacent grids to Gc.

5) Step 5: Select Initial Medoids Based on Grid Density

Step 5.1: Identify the Grid with the Highest Density

Select the grid G_{max} with the highest granularity density: Gmax=arg max$_{Gc∈G}$ GD(Gc)

Define its midpoint as the initial medoid: $Mi=\frac{1}{\|Gmax\|}\sum_{dj∈Gmax} dj$

Step 5.2: Check if the Required Number of Medoids is Reached

If the number of selected medoids k′ reaches the predefined K: k′=K then proceed to Step 6.

Step 5.3: Otherwise, Process the Grid and Recalculate

Mark the selected grid and its adjacent grids as processed. Set their density to zero: D(Gj)=0, ∀Gj∈{Gmax∪N(Gmax)}= 0,Gj Recalculate the granular density and return to Step 5.1.

6) Step 6: Apply K-Medoids Clustering

Use the selected medoids M={M_1,M_2,…,M_K} as initial medoids in K-Medoids Clustering.

Step 6.1: Assign Each Data Point to the Nearest Medoid

Each data point d_i is assigned to the cluster of the closest medoid: Cj={di∈D|Medoid(di)=Mj, j=arg minl dist(di,M_l)} where dist(di,Ml) is the chosen distance metric (e.g., Manhattan, Euclidean).

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Step 6.2: Update Medoids

For each cluster Cj, update the medoid Mj by selecting the point that minimizes the total distance: $M_j=\arg\min_{d\in C_j}\sum_{di\in C_j} dist(di,d)$

Repeat Step 6.1 and Step 6.2 until medoids do not change (i.e., convergence).

7) Step 7: Output the Results

Return:

- The final cluster assignments $C_1,C_2,\ldots,C_K$.
- The final medoid locations $M_1,M_2,\ldots,M_K$.

V. RESULT AND ANALYSIS

The Grid-Based K-Medoids algorithm effectively clusters data by leveraging both grid-based density estimation and the robustness of K-Medoids. By selecting initial medoids based on density peaks, the algorithm overcomes the sensitivity issues of traditional K-Medoids, leading to more stable and accurate clustering results.

TABLE-1
CUSTOMER VISIT DATASET

Cust id	Monthly visits	Avg spend	cluster
1	7	4651	Medium
2	20	3898	Medium
3	29	1375	High
4	15	1116	High
5	11	437	High
6	8	978	High
7	29	1176	High
8	21	4987	Medium
9	7	4093	Medium
10	26	4959	Medium

A. Data Analysis

- Accuracy Improvement: The use of high-density grids for initialization helps avoid poor medoid selection, improving the overall accuracy of clustering.
- Noise Reduction: Low-density regions are filtered out, reducing the impact of noise and improving cluster purity.
- Efficiency Gains: Since clustering operates on grid cells rather than individual data points, computational efficiency is improved, making the algorithm scalable for large datasets.
- Frequent Visitors with Low Spending (High Cluster): Customers like #3 (29 visits, $1375 spending) and #7 (29 visits, $1176 spending) visit often but spend relatively less per visit. These could be budget-conscious customers or those who make small, frequent purchases.
- Moderate Visitors with Medium Spending (Medium Cluster): Customers like #1 (20 visits, $3898 spending) and #8 (21 visits, $4987 spending) show a balance between visits and spending. This group represents a stable middle-tier segment, which can be influenced through personalized promotions.
- Customers with Lower Visits but High Spending (Likely Low Cluster, not shown in this sample): This cluster may contain high-value customers who make fewer visits but spend significantly per transaction. These customers can be targeted with premium offers or loyalty rewards.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

B. Key Observations

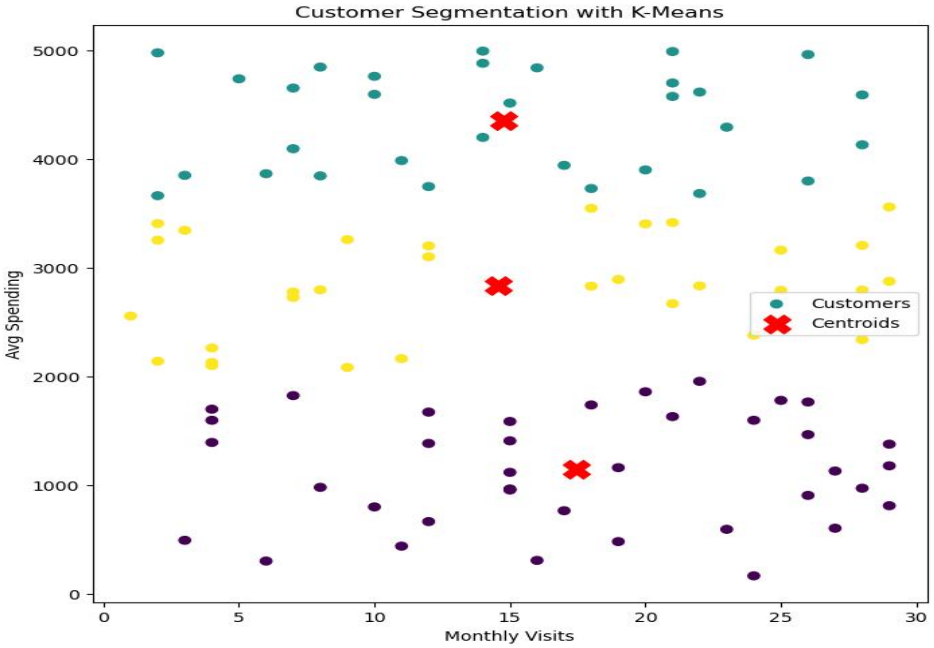


Fig.2. clusters with k-means(red)

1) The graph represents Customer Segmentation using the K-Means clustering algorithm, based on Monthly Visits (X-axis) and Average Spending (Y-axis).
2) Different colors indicate the three distinct clusters of customer spending behavior: Low, Medium, and High.
3) The red X markers represent the centroids of the clusters, calculated by K-Means based on minimizing variance.
4) Unlike K-Medoids, K-Means selects centroids based on mean values, which may be influenced by outliers.
5) The clusters appear well-separated, but some points are spread near cluster boundaries, which may affect segmentation accuracy.

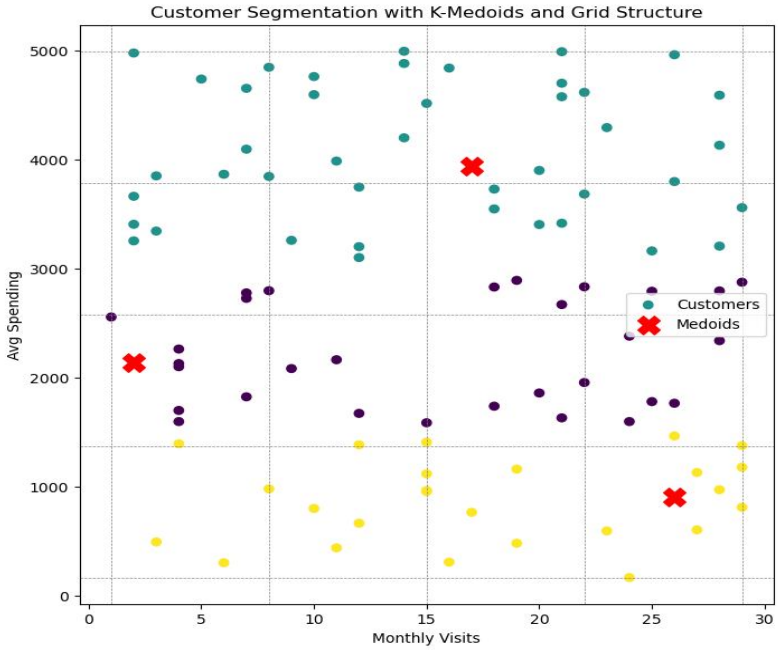


Fig.1 Clusters with Grid Based K-Medoids (red)

©IJRASET: All Rights are Reserved

The scatter plot represents customer segmentation using the Grid-Based K-Medoids Clustering Algorithm. The key components of the graph are:

1) X-Axis (Monthly Visits) – Represents the number of times a customer visits in a month.
2) Y-Axis (Avg Spending) – Represents the average amount a customer spends per visit.
3) Colored Dots (Customers) – Each dot represents a customer, categorized into three different clusters.
    - Different colors indicate different spending clusters (High, Medium, Low).
4) Red 'X' Markers (Medoids) – These represent the central points (medoids) of each cluster, selected using the K-Medoids algorithm.
5) Grid Structure – The dataset is divided into grid cells to determine density and improve the selection of medoids.

- The yellow cluster (low spending customers) is concentrated in the lower half of the graph, indicating customers who make small purchases per visit.
- The purple cluster (medium spending customers) is scattered in the mid-range, balancing visits and spending.
- The teal cluster (high spending customers) is located in the upper part of the graph, indicating customers who tend to spend more on average.
- The red X markers (medoids) are positioned at key central points of the clusters, showing optimal representatives for each group.

C. *Advantages of Grid-Based K-Medoids Clustering*

1) Better Handling of Noise & Outliers: The grid structure helps in filtering noisy or sparsely populated regions, improving clustering accuracy.
2) Density-Aware Initial Medoid Selection: Unlike random initialization in traditional K-Medoids, this approach ensures that medoids are chosen based on dense areas, improving cluster stability.
3) Efficient Computation: Grid-based preprocessing reduces the number of distance computations, making the clustering process more efficient than traditional methods.

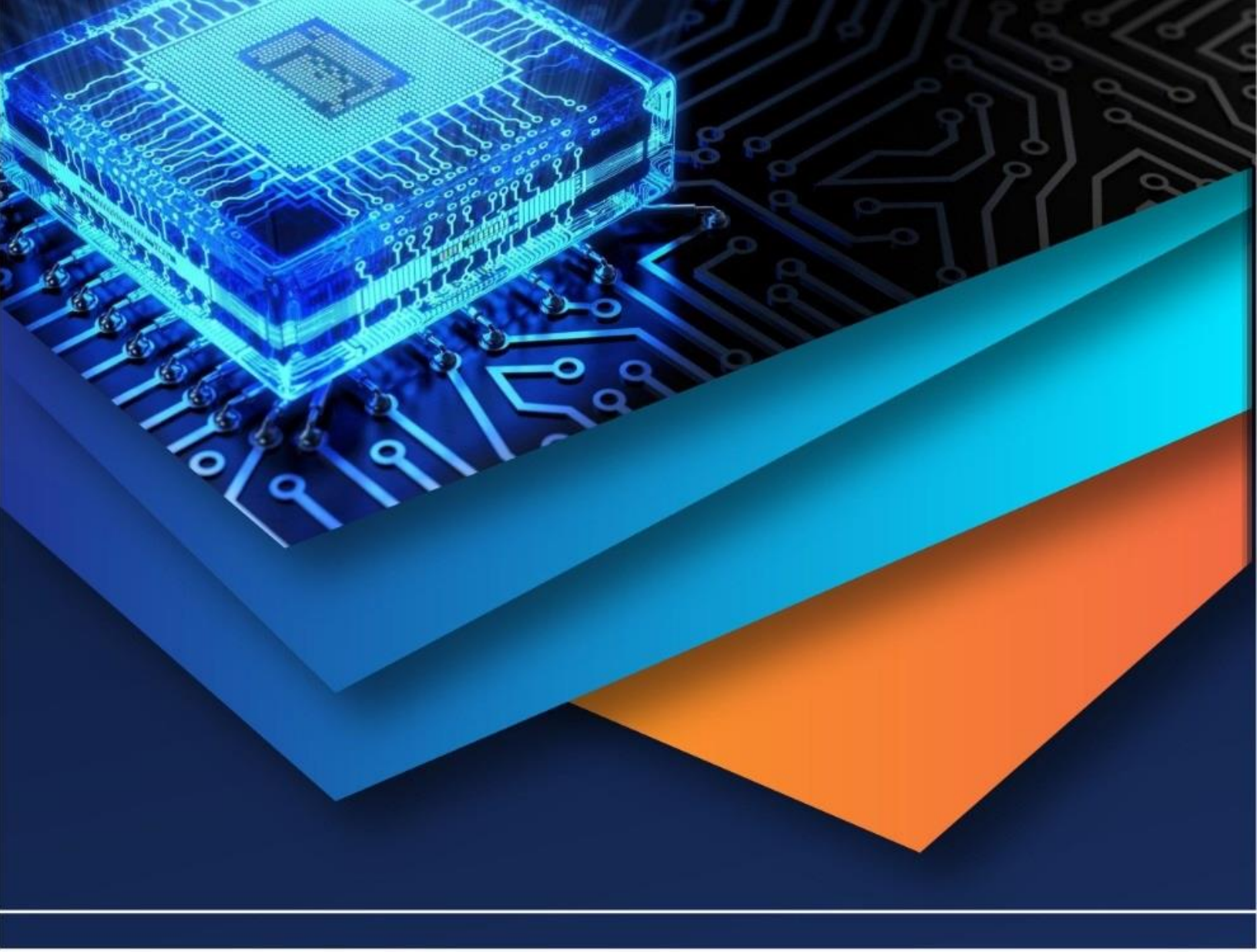
K-Means and K-Medoids differ primarily in how they define cluster centers. K-Means uses the mean of all points in a cluster as the centroid, making it sensitive to outliers since extreme values can shift the mean significantly. In contrast, k-medoids selects actual data points as cluster representatives, making it more robust to noise and outliers. K-means generally relies on Euclidean distance, while K-Medoids can work with various distance metrics, offering greater flexibility.

VI. CONCLUSION

Grid-based clustering algorithms are efficient and scalable, making them well-suited for large datasets. By partitioning the data space into grids, they reduce the computational complexity of clustering. These methods are robust to noise and outliers since clustering is performed at the grid level before refining clusters. They handle arbitrary-shaped clusters better than traditional methods like K-Means. The ability to parallelize grid-based clustering enhances processing speed for big data applications. Combining it with algorithms like K-Medoids further improves accuracy by selecting representative data points. In this paper grid density-based K-Mediods is applied to customer visiting and speinding and generated clusters with various categories and results are given. However, the choice of grid size significantly impacts clustering results and requires careful tuning. Overall, grid-based clustering is a powerful approach for high-dimensional and large-scale clustering tasks.

REFERENCES

[1] Christopher, Isonkobong. (2020). Machine Learning:A Review. Semiconductor Science and Information Devices. 2.10.30564/ssid.v2i2.1931.
[2] Harwath, D., Torralba, A., Glass, J. Unsupervised learning of spoken language with visual context. In Advances in Neural Information Processing Systems, 2016: 1858-1866.
[3] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data clustering: a review. ACM Comput. Surv. 31, 3 (Sept. 1999), 264–323.
[4] Oyewole, G.J., Thopil, G.A. Data clustering: application and trends. Artif Intell Rev 56, 6439–6475 (2023),Springer.
[5] Park, H. S., & Jun, C. H. (2009**).** A simple and fast algorithm for k-medoids clustering. Expert Systems with Applications, 36(2), 3336-3341.
[6] Preeti Arora, Deepali, Shipra Varshney, Analysis of K-Means and K-Medoids Algorithm For Big Data,Procedia Computer Science,Volume 78,2016,Pages 507-512,ISSN 1877-0509,
[7] Raghuvira Pratap A, K Suvarna Vani, J Rama Devi and Dr.K Nageswara Rao, "An Efficient Density based Improved K- Medoids Clustering algorithm" International Journal of Advanced Computer Science and Applications(IJACSA), 2(6), 2011.
[8] Gupta, Er & Gupta, Er & Mishra, Amit. (2012). RESEARCH PAPER ON CLUSTER TECHNIQUES OF DATA VARIATIONS. International Journal of Advance Technology & Engineering Research (IJATER),2012
[9] Jain, A. K... *Data clustering: 50 years beyond K-means.* Pattern Recognition Letters, 31(8), 651-666.2010
[10] Abiodun M. Ikotun et.al..K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, Information Sciences,Volume 622,2023,Pages 178-210,ISSN 0020-0255.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)