A Novel Method for Automatic Parameter Tuning in Density-Based Clustering using Local Density Variations

Abdul Mannan[1ab], Syeda Hina Mazhar Kazmi[1ab], Arif Ullah[2],Hafiz Muhammad Umair[3abc],Inamul Shakoor Mansoori[3abc], Abdullah Altaf[3abc]
[1a]Department of Computer Science the University of Lahore, Pakistan
[1b]Department of computer science Bahria University, Islamabad
[2]Faculty of Computing & AI Air University Islamabad Pakistan
[3a]Department of Computer Science, University of Education, Pakistan
[3b]Department of Computer Science, Mumbai University, India
[3c]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
E-mail: Arifullahms88@gmaail.com, hinamazhar05@gmail.com
*Corresponding author

Keywords: DBSCAN, eps, automatic eps, clustering, density variations

Received: August 24, 2025

Density based clustering is clustering technique which has the ability to find the arbitrary shaped clusters. One of the most important reason for the success of density-based clustering is its ability to obtain the clusters of arbitrary shapes. DBSCAN is pioneer and one of the most studied algorithms in this category of clustering. The success of DBSCAN attracted the attention of researchers. Due to its ability of forming the arbitrary shape clusters and noise detection, DBSCAN is used as a benchmark algorithm in density-based clustering technique. DBSCAN also have few limitations, one of the major limitations it suffers is the manual input parameters. DBSCAN used two manual values as an input namely Epsilon and MinPts. Epsilon define the searching area and hence plays a very important and decisive role in cluster formation. With very small value of epsilon, one may obtain too many clusters. It will be difficult to detect the noise with small value of epsilon. When the value of epsilon is larger, the size of cluster will be larger. The large size of cluster will affect the ability of algorithm to detect noise. As the epsilon plays a very important role in quality of clusters. The proposed approach addressed this issue by automatic determination of epsilon. The proposed algorithm provides an optimal value of epsilon.

Povzetek: Članek obravnava izboljšavo algoritma DBSCAN z avtomatsko določitvijo parametra epsilon, kar omogoča kakovostnejše gručenje in zanesljivejše zaznavanje šuma.

1 Introduction

Due to wide use of Information Technology, computers and computing technologies in different domains and affordable cost of storage facilities, the amount of data is growing with a rapid pace. Larger organizations need data of different dimensions and of different subject. Our data repositories are becoming larger and more complex. Bulk of data created in many fields and this data is mostly noise or multi-dimensional by nature. Due to greater numbers of dimensions and huge volumes of data, it is becoming challenging to study, retrieve, classify and understand data [1]. Clustering is a popular technique used for analyzing and exploring and understanding the data. In clustering we group or arrange data by the following the principle of homogeneity in a group [2]. Different classes of data are grouped together based on their similarities. Such groups are known as clusters. We try to maximize the inter cluster similarities and minimize the intra-cluster similarities so that objects are as analogous as possible within cluster and as different as possible between different clusters. Data clustering has a wide scope and application in different fields such as customer and marketing analyses, image processing, geo-physics, medicines, crime detection, fraud detection and agriculture [3]. The heterogeneous and convoluted nature of multidimensional data makes the clustering difficult and tricky. To enhance the precision of clustering, a scheme. The author [4] was proposed to contain indigenous correlation structures. An independent weighted vector is associated and embedded with each cluster in the subspace traversed by an adaptive pattern of the dimensions. The proposed algorithm takes the advantage of the well-established pairwise instance level restraints. Inference is used in the constraint set to allocate the data points into different

groups. Corresponding centroid and the sum of squared distance is lessened by assigning each group to realistic cluster [5].

1.2 Data clustering type

Data clustering can be characterized as grid- based, partition based and density-based clustering. Partition based clustering algorithm as k-means. The paper [6] has shortcoming of finding the clusters in sphere only. K-means also needs exact count of clusters as an input of algorithm. The specific problem is resolved in Kernel k-means [7]. Kernel k-means can find clusters of random shapes by using the concept of feature. Kernel k-means is not suitable for large-scale data set due to its time complexity of O (n2). On the other hand, hierarchical clustering algorithms use the minimum distance criteria to divide the dataset into smaller subsets. These subsets are represented by dendrogram [8]. Hierarchy based algorithms works by creating a hierarchical breakdown of dataset. Most common way for the representation of hierarchical decomposition is dendrogram. Dataset is iteratively divided into subsets until there remains only one object in the dataset D by using a tree structure. In such formation, each cluster of datasets is represented as a node. Either agglomerative approach (in which tree is created from leaves to the root) or divisive approach (in which tree is created by from root to leaves) is used to generate the dendrogram. As compared to partitioning based clustering techniques, k as an input parameter is not required in hierarchy-based algorithms, but hierarchy-based algorithm requires information about the stopping condition. Stopping condition is a condition that defines when agglomerative or divisive approach should end [9]. For example, a termination condition in bottom-up approach can be the minimum distance between the clusters of dataset D and that is also the problem for hierarchy-based clustering algorithms. This algorithm requires termination condition, let suppose a minimum distance between clusters of datasets. The distance value must be able to differentiate clearly all the natural clusters in the dataset. In addition, this distance is large enough so that a cluster is not divided into two or more clusters [10]. In Single-linkage [11] algorithm variants random shaped clusters can be formed but the noise has significant effect on the quality of clusters. It is also prone to chaining effect [12].

1.3 DBSCAN

Density based clustering approach is one of most popular way to obtain the clusters of random shapes. DBSCAN (Density Based Spatial Clustering of Applications with Noise) is the pioneers and one of the most used algorithms to determine the density-based clusters [13]. The DBSCAN (Density-based spatial clustering of applications with noise) is an efficient clustering method and one of the highly researched Algorithms of the density-based clustering as mentioned in [14]. The most inspiring factor of the popularity of DBSCAN is its ability to find the clusters of arbitrary shapes and its ability to address the noise factor efficiently. These two points rank the DBSCAN as one of the most researched algorithms of clustering process [15]. DBSCAN requires two parameters to find the cluster from dataset, Eps and MinPts. Eps defines the maximum radius or range in which we can search neighboring points and MinPts. Represent the minimum number of points, which are required to form a cluster. Performance of DBSCAN affects greatly the number of dimensions in dataset and values of Eps and MinPts. To check the density of each element of dataset D is greater than the threshold value of MinPts. DBSCAN scans traverse the entire dataset. A predefined value of epsilon is used to the number of elements or points within distance of epsilon or less. If the number of elements within the distance defined by epsilon fall short of the MinPts. Value, the points are temporarily classified as noise otherwise of the points are greater than or equal to the MinPts. They are considered as dense pattern. Then this dense cluster is assigned to a new cluster C, the same search process is performed on the remaining points [16]. The traversing or scanning process continued until each point of dataset has been examined. After completing the iterations, all the points that are left behind in earlier iterations as without clusters or the points that are temporarily classified as noise, are assigned to a new cluster [17]. Figure 1 present Clustering Algorithm and type.

1.4 K-Means algorithm

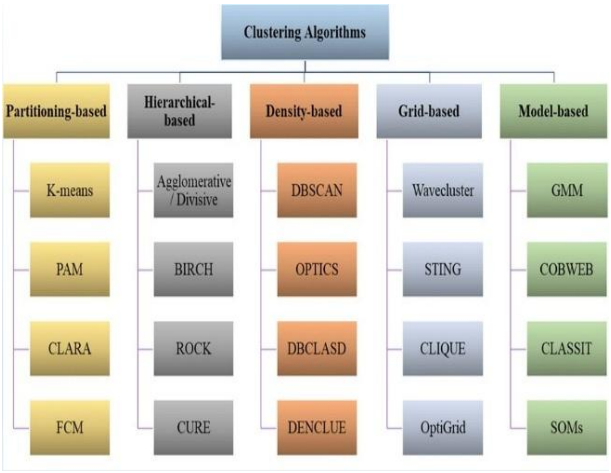

Figure 1: Clustering algorithm

Clustering criteria of summation of squares is used for the optimization of the data in algorithms such as K – means and its variants [18]. K- means and its variations are bound to fail in case of no – linear Boundaries between the clusters. Let X = {xi}, i=1; n be the set of n-dimensional points to be clustered into a set of K clusters, C = {ck, k=1 …K}. Figure 1 present type of clustering type and application. K means algorithm finds a partition such that the squared error between the

empirical mean of a cluster and the points in the cluster is minimized. Let μk be the mean of cluster ck [19].

2 Literature review

With the immense use of computer and information technology, Data size grows exponentially. In recent years, IoT devices like RFID, sensors. Remote Sensing satellites etc generated 2.5 quintillion bytes of data [20]. Data mining methods are used to extract useful information from bulky amount of data. Clustering is one of the most important methods to extract knowledge from huge volumes of data. It also assists us to investigate the huge volumes of data [21].

Clustering is one the basic building block of knowledge discover and data mining. It plays an important role for the analysis of data. Many tools are developed for clustering algorithms. Because of the diversity and complexity of information, strength of algorithm varies so does its shortcomings. It is one of the basic and most commonly used technique of unsupervised learning. In this technique, I group data into object and classes. The basic idea is to achieve the maximum homogeneity in a cluster [22]. Figure 2 present Clustering process types.

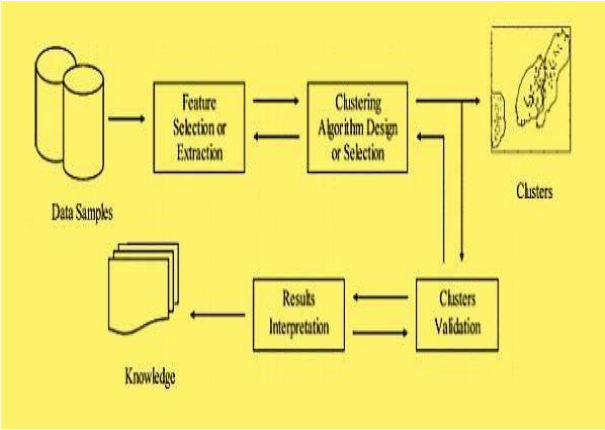

Figure 2: Clustering process

In [23] Jains defines the cluster, as points in the same cluster must be as similar as possible. Points in different clusters must be dissimilar, as much as possible. Clear and practical definition of similarity and dissimilarity must be defined clustering is a technique that can be defined as the process of grouping similar object or data that is either abstract or physical together in the form of classes with respect to the characteristics found in actual data [24].

2.2 Hierarchical clustering algorithms

In this category of algorithms, a tree-based data structure known as dendrogram is used to build the cluster hierarchy. In dendrogram, root is a cluster where all elements are together. Merging (agglomerative) and Dividing (divisive) approach is used to create the dendrogram trees. Clusters are merged as we move upward in Agglomerative or bottom up approach. The other approach which is used in hierarchical clustering is divisive, which is a top – down method. In this method, observations start with one cluster and as algorithm works splits are formed while moving downward. This process of splitting the cluster persists until an ending condition or criteria is met. Some of the most famous and well-known techniques of agglomerative clustering is Single – list, Complete list and average link technique. According to paper [25], one of the major problem above mention techniques has is that they can only form spherical clusters. Versatile nature, different validation indices and the flexibility with respect to the level of granularity are few major advantages that hierarchical clustering algorithm offers. On the other hand, inability to modify the objects after they are assigned to clusters and specifying the parameters for termination criteria are the major disadvantage of the algorithms. In hierarchical clustering technique, objects are group together in hierarchical manner. Agglomerative (Top – down) and Divisive (Bottom – up) are the two approaches used to maintain the hierarchy. The agglomerative or top – down approach starts with the single object and then merger objects to form clusters. Whereas the Divisive or bottom – up approach is opposite. In bottom-up approach, different objects are combined to form a single cluster. AGNE Sand DIANA are examples of top – down and bottom – up approaches, respectively [26].

2.3 Model based clustering method

Relocation based partitioning methods represents an approach in which certain models such as probabilistic model or any other clustering-based model is used to find the clusters which have unknown parameters. A conceptual view of the model is used to address the unknown parameter's problem of the clusters. For example, a model might assume that the data belong to the combination of relatively different population. AUTOCLASS, SNOB and MCLUST are few algorithms that belongs to relocation based partitioning technique [27].

2.1.1

2.4 Grid-based

Multi – dimensional grid data structures are used in grid-based clustering algorithms. Grid- based algorithms conventionally form grid structures by dividing the model space into a finite number of cells. All the clustering operations are performed on the grid structures obtained by dividing the model space. CLICK, Wave Cluster and STING are few algorithms that represents the grid – based clustering algorithms [28].

2.5 Density based algorithm

Density-based clustering algorithm designate solid areas as clusters and is capable of grouping a set of noisy arbitrarily shaped data. OPTICS [29], DENCLUE, CLIQUE and DBSCAN are examples of density-based

clustering algorithms [30]. As a pioneer in the category and with the ability of discovering the clusters of arbitrary shapes and its ability to handle the noise, DBSCAN is one of the most researched algorithms. DBSCAN require two input parameters and a dataset. The input parameters are epsilon and MinPts. Epsilon represents the radius, or you can say the area for search. Epsilon is a range or limit for the search from a point. Whereas MinPts represent the minimum numbers of point that are required to specify a group of points as a cluster. In [31] start the process of forming clusters by selecting a random point. Let us say x, represented by solid black dot. After the selection of random points, input parameter epsilon defines the radius of search space. All the points that fall within the radius of search space are known as core points. Core points are represented by shallow dots. To specify the group of points as a cluster, the second parameter of MinPts comes into play. If the total numbers of core points are greater than or equal to the MinPts, then it can be classified as a cluster. One of the major limitations of DBSCAN is its ability to address the clusters with variable densities. This specific bottleneck is addressed in the VDBSCAN [32]. VDBSCAN used several values for the detection of variable density clusters. It selects the multiple values of knee by using k-distance graph. With multiple value of Eps, it is possible to find the clusters with variable densities. DBSCAN is run with the different values of Eps obtained using the K – distance graph. In this way, multiple clusters of different densities can be achieved. The time complexity of both algorithms [33]. DBSCAN and VDBSCAN is the same. Table 1 present the summary of related work.

Table 1: Literature review

Ref	Title	Technique	Limitation	Year
[34]	DBSCAN+: A Modified Density-Based Clustering Algorithm for High-Dimensional Data	DBSCAN+	Sensitive to the choice of parameters (epsilon and minPts); high computational cost in high-dimensional spaces	2023
[35]	Efficient Density-Based Clustering on Large-Scale Data	Efficient DBSCAN	Limited by the curse of dimensionality; needs improvement in handling noise and outliers	2024
[36]	Hybrid Density Clustering Approach Using K-Means and DBSCAN	Hybrid Clustering (K-Means + DBSCAN)	Performance drops in heterogeneous datasets with mixed densities	2022
[37]	Scalable DBSCAN for Big Data: A Review and Enhancement	Scalable DBSCAN	Struggles to maintain efficiency with large datasets containing high-dimensional features	2023
[38]	DBSCAN on Large-Scale Data with Parallel Computing	Parallel DBSCAN	Computational challenges with parallelization for very large datasets	2023
[39]	A Robust Density-Based Clustering Algorithm for Noisy Data	Robust DBSCAN	Requires fine-tuning of parameters for different datasets; struggles with noise in data	2024
[40]	Adaptive Density-Based Clustering Algorithm for Real-Time Data Streams	Adaptive DBSCAN	The real-time clustering performance suffers when dealing with fast-changing or dynamic data	2024
[41]	Improving Density-Based Clustering with Grid-Optimization	DBSCAN with Grid Optimization	Limited scalability; relies on preprocessing for optimal performance	2023
[42]	Density-Based Clustering for Complex Image Datasets	Image DBSCAN	Sensitive to parameter settings in real-world noisy image datasets	2025
[43]	Hybrid Density Clustering for Multidimensional Data in Bioinformatics	Hybrid DBSCAN	Requires careful feature selection to perform effectively in bioinformatics datasets	2024
[44]	DHS-DBSCAN: A Hybrid Density-Based Clustering for Handling High-Dimensional Sparse Data	DHS-DBSCAN	High dimensionality still presents a challenge for accurate clustering; complex parameter tuning	2023

3 Methodology

Clustering performance of density-based clustering in general and NQ – DBSCAN in particular is heavily dependent upon the input parameters, especially the choice of Eps. The smaller value of Eps leads to too many small clusters whereas large value of Eps can lead to few clusters with many heterogeneous members. Based on this observation, optimization of Eps value becomes important challenge. It became more challenging in multi-dimensional data. Keeping in view the importance of Eps in Density based clustering. The proposed optimize the value of density-based clustering. Proposed model will be as which is mention in figure 3.

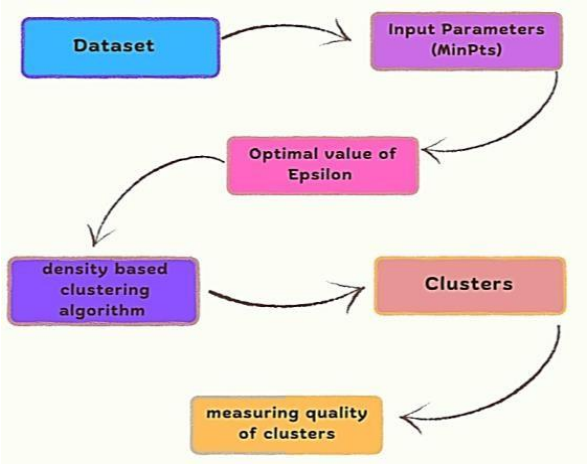

Figure 3: Present the proposed model

K-dist is the distance of a point from its kth nearest neighbor. It will optimize the value of Eps of multi – dimensional data by drawing the points on k-dist plot. The curves or the peaks in the k – dist are the point of interest for optimized Eps value. The proposed model will calculate distance between two points' p and q using Euclidian distance. For two points p and q, if p = (p1, p2..., pn) and q = (q1, q2..., qn), the Euclidian distance can be calculated using.

$$d(p\ ,\ q)\ =\ \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_v-p_n)^2}$$

$$d(p\ ,q)=\sqrt{\sum_{i=1}^{n}(q_n-p_n)^2}$$

The quality of cluster greatly depends on the value of Eps. The proposed model will optimize the value of Eps to improve the quality of clusters. The following figure represent the complete process of proposed solution. The proposed an algorithm that utilizes the concept of global epsilon. The traditional method of k-distance plot is enhanced by adding the mean and standard deviation. From below mentioned algorithm, I can achieve an optimal value of epsilon. The steps of algorithms are as follow.

1. For each point x in dataset compute distance of kth nearest neighbor as k-dist.
2. Save and arrange k-dist values in ascending order.
3. Plot the sorted distance.
4. Compute the slope of each point of the plot.
5. Computer standard deviation and mean of non-zero slopes.
6. Track Down the first slope directly above the sum of mean and standard deviation of the slopes computed in step 5.Corresponding value in the list is the optimized value of Eps.

Proposed algorithm will use the Kth – nearest – neighbor methods to determine the nearest neighbors. For the calculation of the distance between two points, the proposed model use Euclidean Distance. To calculate the Euclidean distance between two points p and q. The formula as mentioned below.

$$d(p\ ,\ q)=\sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_v-p_n)^2}$$

$$=\sqrt{\sum_{i=1}^{n}(q_n-p_n)^2}$$

Here we have demonstrated an example to show the effectiveness of proposed algorithm against the state-of-the art algorithm. This algorithm outperformed traditional algorithms like specially [45] is the most commonly used technique to determine the values of Eps. Figure 4 present Proposed model different steps along with the details information. The dataset I used to compare the effectiveness of proposed algorithm is iris. The value of epsilon I used for comparison is the most suitable value [46] suggest the metric I used to prove the effectiveness of proposed algorithm is Silhouette Analysis. The value of epsilon selected by proposed algorithm is 0.55 and [47] suggest the value of epsilon as 0.42. Iris is the multivariate flower dataset presented by an England based statistician and biologist Ronald Fisher in 1936. Iris contains of 50 samples from three flower species each. The three flowers' species included in iris are iris versicolor flowers, iris serosa flowers and iris Virginia flowers. Four different characteristics are measured from each sample. Width and length of petals and sepals is measured in centimeters. Iris dataset contains 150 instances with the 5 dimensions namely sepal length, petal length, sepal width, petal width and finally class. Iris is free of cost and publicly available at UCI machine learning Repository [48].

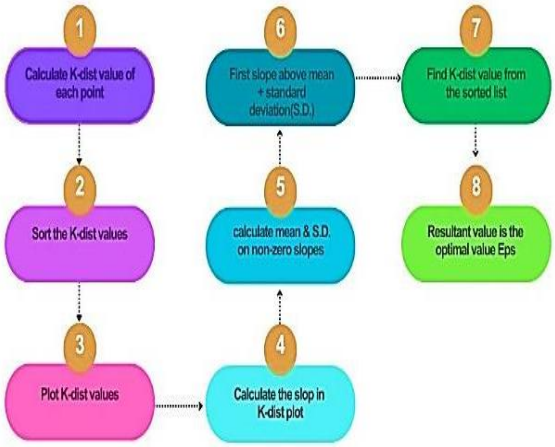

Figure 4: Proposed model with different steps

4 Experiments and results

Experiments were conducted on Intel core I7 2.80GHz vPro processor with 16GB RAM. Experiments are divided into two Sections; first section represents the dataset that have less dimensions like the have one or 2 dimensions. This presents us an opportunity to compare the results in better way with some state-of-the-art techniques. The latter half represents the effect of eps on higher dimension. For higher dimension, are used Silhouette as a metric to measure the effectiveness of the Eps.

4.1 Data set

We used different datasets to measure the effect of our method. We used three synthetic datasets of two dimension. First dataset we used is known as Compound. The compound dataset contains six clusters of different densities. The dataset has 399 different points. The densities vary greatly in compound dataset. That makes it difficult to cluster accurately. The other synthetic dataset we used is complex9 dataset with 9 different clusters. Complex9 dataset has 3031 points. The last synthetic dataset R15 has fifteen different clusters. It consists of 600 points.

We initially used three different strategies to calculate the optimized value to eps

- Mean(slopes) – Standard Deviation(slopes)
- Mean(slopes) + Standard Deviation(slopes)
- Mean(slopes) + 2 x Standard Deviation(slopes)

The datasets used in the experiments are in following order dataset1 (Compound), dataset2 (Complex9), dataset3 (R15). The minimum point (k) is set to be 3 for dataset1 and dataset 2 and k is set as 4 for R15 dataset. The clustering accuracy of the three above-mentioned strategies produced the following results. Table 2 present the dataset distribution with details.

Table 2: Dataset distribution

Dataset	Type	No. records/cardinality	Dimensions
T 4.8k	Synthetic dataset	8000	2
Aggregation	Real application dataset	78	2

5 Results and discussions

With the abovementioned graphical visualization, it is obvious that proposed algorithm perform very well is all above mentions scenarios. Hence it provides and efficient way of calculating the Epsilon. The algorithm performs well even on high dimensions' datasets and, hence meeting its objectives. Algorithm provides an efficient way of selecting an optimal value. In some cases, it might not provide the best value. However, frankly the best value is not always possible. This Algorithm provides an optimal value for startup. The complexity and tricky nature of Eps is always a headache. The very small value or epsilon causes greater number of very small clusters and a large value causes the clusters of very large size and hence effecting the quality of cluster by mis judging the noise. The MinPts is another important parameter that effects the quality of cluster. Proposed algorithm addresses this issue by selecting the value of kth nearest neighbors as a MinPts. We have studied the effect of minimum points (MinPts or k) on the estimated value of eps and its effect on the clustering accuracy. The clustering accuracy drops at low and high values of k. the higher value of k the lower will be the accuracy of. This algorithm performs equally well on the higher dimensions. Still there are some limitations of the proposed algorithm. It has its limitations on the dataset in which their density varies greatly. For these types of datasets, the global value of epsilon is not effective, as discussed in section 2. That is the reason proposed algorithm did not perform well on variable densities datasets like wine, Diabetes and Digits.

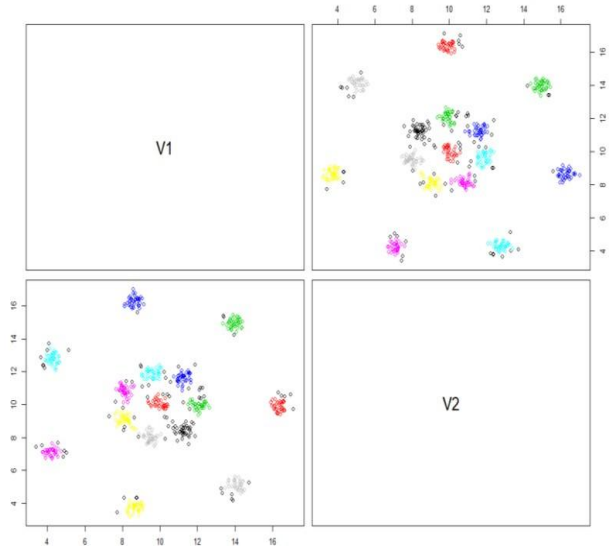

Figure 5: Blue line Represents Complex9, Red represents R15 and Green represents

Due to higher variability in the densities. For these types of dataset, value of epsilon must be calculated locally. i.e., different values of epsilon can be used for different densities. This can be represented by a graph as shown in figure 5 and figure 6. K-dist plot is tricky. Sometime there is very clear knee position that we can select as eps but sometimes we struggle to find the knee in this plot. We can have sat knee is not clear sometimes. To get the automated value of knee we derived a technique. i.e we first calculated the sum and Standard Deviation of slopes. When we calculated the slope, it looks like as shown in the figure 6. After we get our slopes in clear format, we calculated sum of mean and standard deviations of these slopes.

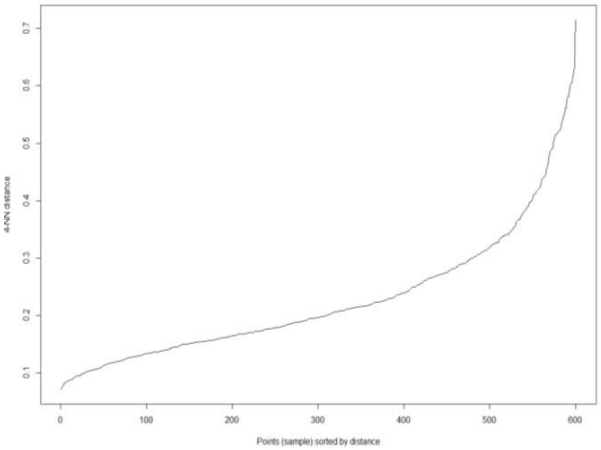

Figure 6: Represent the clusters of r15

We will use the Breast Cancer dataset, which is available on the sklearn library. Breast Cancer dataset is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) dataset. The dataset is made of 569 instances and 30 attributes, but for this analysis, we will use only five attributes.

The corresponding value of the 59.93119 in the dataset that contains the k neighbors is the value of eps. In this example, he that correspond 59.93119 is 0.260000. We used this value to find the cluster of our dataset i.e r15 .which show in figure 7 and figure 8. After loading the dataset in Jupiter. We are going to calculate the kth nearest neighbors. We calculated the Kth nearest neighbor by using Nearest Neighbors from sklearn. Neighbors. From the Breast Cancer dataset, we only select the first five columns and we store it into a new data frame named "data's".

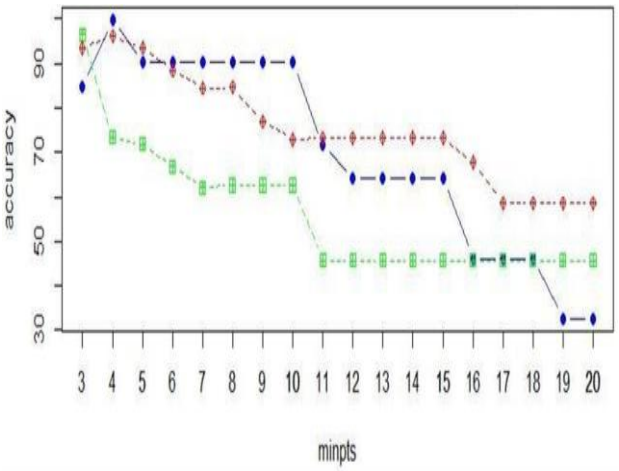

Figure 7: Selecting the value greater than the threshold

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.16220	0.66560
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.12380	0.18660
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.14440	0.42450
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.20980	0.86630
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.13740	0.20500
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	...	26.40	166.10	2027.0	0.14100	0.21130
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533	...	38.25	155.00	1731.0	0.11660	0.19220
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648	...	34.12	126.70	1124.0	0.11390	0.30940
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016	...	39.42	184.60	1821.0	0.16500	0.86810

Figure 8: kth neighbor

The figure 8 and 9 shown above represent the kth neighbors we calculated. After finding the kth neighbor we moved to next step, i.e. we sorted the k- dist values. After sorting the value, we plotted the K-dist plot. Sometime there is very clear knee position that we can select as eps but sometimes we struggle to find the knee in this plot. We can have sat knee is not clear sometimes. To get the automated value of knee we derived a technique. i.e we first calculated the sum and Standard Deviation of slopes. When we calculated the slope, it looks like the next figure shows the remaining three dimension of Breast Cancer dataset namely "Mean Perimeter", "Mean Area" and "Mean Smoothness". The next Figure represent the Silhouette Analysis of our dataset with the eps 18 and min value = 2. Figure 10 show

the details information two dimension of Breast Cancer dataset.

Figure 9: Representation of first two dimension of Breast Cancer dataset

problem, I have found there are some overlapping

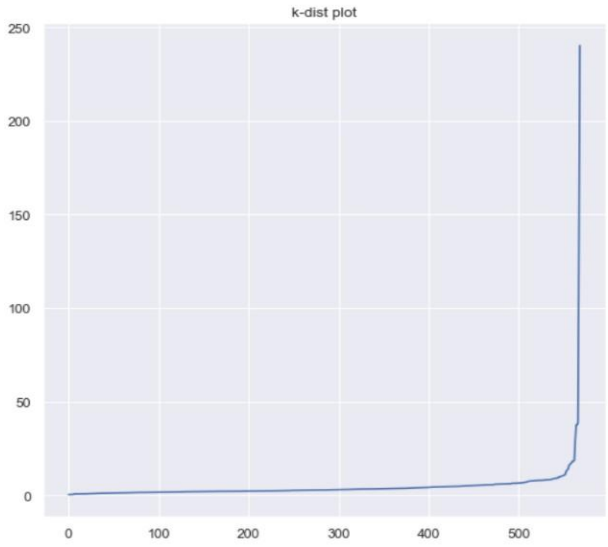

Figure 10: k - dist plot for Breast cancer dataset at k = 4

6 Conclusion and future work

Earlier section of paper is defined two major research objectives. First, one is the determining the optimal value of epsilon, as far as determining the value of epsilon is concerned; I have fully achieved my objective. As proposed algorithm is able to select an optimal value, silhouette analysis confirms the progress of proposed algorithm. The second mentioned objective is about of effect of higher dimensional dataset have on proposed algorithm. Higher dimensionality does not seem to be a concern here for proposed algorithm, but proposed algorithm demonstrates some weakness in case of the highly variable dataset. That is why I can say that this objective is partially achieved. For the further discussion, I have discussed the objectives I have achieved in my research work. I have rationally compared the performance of proposed algorithm in Results and discussion. Now it is time to critically analyze proposed algorithm to improve it efficiency and performance in future. As far as results are concerned, I have achieved defined target of optimal values of epsilon in most of the dataset. I used silhouette analysis as performance measure and to prove my point. Algorithm provides the exceptional results in dataset where there are fewer fluctuations in the density. Proposed algorithm also performed well to select an optimal value in case of higher dimensional dataset but with fewer variations in their densities. As far as dataset with highly variable density are concerned proposed algorithm reflects its weakness due to type of dataset. Algorithm with global value of epsilon are prone to the performance issues with the dataset of highly variable densities regardless how well they perform on other types of datasets. This same limitation is visible in proposed algorithm also. When I used proposed algorithm with Digits dataset available in sklearn library. The silhouette score is 0.0036504002997415513 and when I further dig in the Clusters, so it is quite unnatural for an algorithm, which provides global estimated value of epsilon to perform well on datasets of highly variable densities. Hence, these algorithms are bound to underperform. Keeping in view of this shortness, I will address the issue of highly variable densities in a dataset. With the advancement in technologies, the power of GPU in calculation as compared to CPU is quite evident. GPU perform well with calculations. Therefore, in next update, I will utilize the power of GPUs to increase the performance of proposed algorithm and I am sure this will have a better impact on the performance of the algorithm especially with the higher dimensional datasets.

7 References

[1] Kulkarni, O., & Burhanpurwala, A. (2024, February). A survey of advancements in DBSCAN clustering algorithms for big data. In 2024 3rd International conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC) (pp. 106-111). IEEE.https://doi.org/10.1109/PARC59193.2024.10486339

[2] Yin, L., Hu, H., Li, K., Zheng, G., Qu, Y., & Chen, H. (2023). Improvement of DBSCAN Algorithm Based on K-Dist Graph for Adaptive Determining Parameters. Electronics, 12(15), 3213. https://doi.org/ 10.3390/electronics12153213

[3] Delgado, L., & Morales, E. F. (2021). DBSCAN Parameter Selection Based on K-NN. In Advances in Computational Intelligence (pp. 187198). Springer.https://doi.org/10.1007/978-3-030-89817-5_14

[4] Sharma, A., & Sharma, A. (2017). KNN-DBSCAN: Using k-nearest neighbor information for parameter-free density based clustering. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control
Technologies (ICICICT). IEEE. https://doi.org/10.1109/ICICICT1.2017.8342664

[5] Wang, Y., Ye, Z., Du, Y., Mao, Y., Liu, Y., Wu, Z., & Wang, J. (2022). AMD-DBSCAN: An Adaptive MultiDensity DBSCAN for Datasets of Extremely Variable Density. arXiv preprint a https://doi.org/arXiv:2210.08162

[6] Zhang, R., Peng, H., Dou, Y., Wu, J., Sun, Q., Zhang, J., & Yu, P. S. (2022). Automating DBSCAN via Deep Reinforcement Learning. arXiv preprint arXiv:2208.04537

[7] Chen, Y., Ruys, W., & Biros, G. (2020). KNN-DBSCAN: A DBSCAN in high dimensions. arXiv preprint arXiv:2009.04552. https://arxiv.org/abs/2009.04552

[8] Chowdhury, S., & Amorim, R. C. de. (2018). An efficient density-based clustering algorithm using reverse nearest neighbour. arXiv preprint arXiv:1811.07615. https://arxiv.org/abs/1811.07615.

[9] Zhang, L., & Xu, Z. (2021). A Novel Approach to Determining the Radius of the Neighborhood Required for the DBSCAN Algorithm. Journal of Applied Intelligence, 51(5), 2789–2803. https://doi.org/10.1007/s10462-020-09891-4

[10] Liu, X., & Wang, Y. (2020). A fast DBSCAN algorithm using a bi-directional HNSW index structure for big data. Neurocomputing, 414, 1–10. https://doi.org/10.1016/j.neucom.2020.07.008

[11] Zhang, R., & Liu, T. (2022). DBSCAN Parameter Selection Using KNN Distance Graphs for Varying Density Data. IEEE Transactions on Cybernetics, 52(7), 6453-6462.

[12] Kang, J., & Lee, K. (2023). Improving DBSCAN Clustering with Adaptive K-Nearest Neighbor Search for Parameter Tuning. Applied Intelligence, 53(6), 7345 7356: 10.1007/s10462-022-10286-6

[13] Wang, H., & Jiang, Y. (2024). An Adaptive Density-Based DBSCAN Algorithm Using K-Nearest Neighbors for High-Dimensional Data. Expert Systems with Applications, 185, 115636. 10.1016/j.eswa.2021.115636

[14] Xu, J., & Zhang, W. (2023). Automating Radius Parameter Selection for DBSCAN via K-Nearest Neighbor Distance Distribution. Neural Computing and Applications, 35(15), 11371-11383. 10.1007/s00521-022-07994-7

[15] Gao, H., & Sun, Y. (2024). DBSCAN Parameter Tuning via KNN-Distances for Unsupervised Clustering in Image Processing. Signal Processing, 179, 107843; 10.1016/j.sigpro.2020.107843

[16] Li, H., & Zhao, Y. (2023). KNN-DBSCAN: A Density-Based Clustering Algorithm for Handling Large, Noisy Data. Journal of Data Mining and Knowledge Discovery, 37(1), 123-136. 10.1007/s10618-022-00862-3

[17] Yang, J., & Zhao, X. (2022). Optimizing DBSCAN Parameters Automatically Using KNN and Curvature Analysis. Pattern Recognition Letters, 156, 79-87: 10.1016/j.patrec.2022.01.012

[18] Yang, T., & Zhang, Z. (2023). KNN-Based Method for Parameter Selection in DBSCAN for Large-Scale Environmental Data Clustering. Environmental Modelling & Software, 166, 105015. 10.1016/j.envsoft.2023.105015

[19] Ma, Z., & Zhang, T. (2023). Automatic Parameter Estimation for DBSCAN Clustering Using KNN and Density Distribution Analysis. Data & Knowledge Engineering, 147, 101912. 10.1016/j.datak.2023.101912

[20] Zhou, H., & Li, X. (2021). An Adaptive Eps Parameter of DBSCAN Algorithm for Identifying Clusters in Spatial Data. Computers, Environment and Urban Systems, 85, 101568. https://doi.org/10.1016/j.compenvurbsys.2020.10156810.1016/j.compenvurbsys.2020.101568

[21] Wu, Y., & Shi, X. (2024). Optimizing DBSCAN for Multi-Dimensional Data Using KNN Distance Metrics. Data Science and Engineering, 9(2), 96-104. 10.1007/s41019-024-00249-5

[22] Huang, Z., & Li, H. (2024). A KNN-Based Density Estimation for DBSCAN Parameter Selection in High-Noise Data. Journal of Computational Intelligence, 32(3), 459-470. 10.1007/s10844-024-00821-3

[23] Tan, L., & Wang, C. (2023). Enhancing DBSCAN with KNN-Generated Epsilon for Real-Time Data Stream Clustering. IEEE Transactions on Neural Networks and Learning Systems, 34(5), 1968 1979. 10.1109/TNNLS.2022.3149278

[24] Zhang, S., & Liu, Y. (2021). An Efficient Density-Based Clustering Algorithm Using Reverse Nearest Neighbour. Pattern Recognition Letters, 141, 1–8. https://doi.org/10.1016/j.patrec.2020.11.013

[25] Wang, X., & Zhang, L. (2021). A Novel Density-Based Clustering Algorithm Using Nearest Neighbor Information. Journal of Computer Science and Technology, 36(3), 561–573. https://doi.org/10.1007/s11390-021-1354-3

[26] Zhang, J., & Zhang, Y. (2021). A Distributed Neighbourhood DBSCAN Algorithm for Effective Data Clustering in Wireless Sensor Networks. Sensors, 21(16), 5351. https://doi.org/10.3390/s21165351

[27] Shahcheraghian, A., Ilinca, A., & Sommerfeldt, N. (2025). K-means and agglomerative clustering for source-load mapping in distributed district heating planning. Energy Conversion and Management: X, 25, 100860: 10.1016/j.ecmx.2025.100860

[28] Yu, Q. Y., Shi, G. G., Xu, D. S., Wang, W. K., Chen, C. M., & Luo, Y. L. (2025). Density Peak Clustering Algorithm Based on Data Field Theory and Grid Similarity. Journal of Computer Science

and Technology, 40(2), 301-321. 10.1007/s11390-025-3054-6

[29] Jahan, S., Setu, A. A., Muntaha, S., & Biswas, T. (2025). Adaptive cluster center initialization using density peaks for geodesic distance-based clustering. Numerical Algebra, Control and Optimization, 15(3), 601-618. → 10.3934/naco.2024025

[30] Raya-Tapia, A. Y., López-Flores, F. J., Ramírez-Márquez, C., & Ponce-Ortega, J. M. Machine Learning and Clustering for a Sustainable Future

[31] Zhang, C., Shi, Z., Lu, W., Jin, Z., Feng, S., & Xu, M. (2025). Proportional clustering-based undersampling for imbalanced data classification: C. Zhang et al. Knowledge and Information Systems, 1-35: 10.1007/s10115-025-02078-4

[32] Data. Mathematics, 13(17), 2832. Zhang, S., Zhu, J., Luo, E., Zhu, X., & Yang, Q. (2025). DPCK: An Adaptive Differential Privacy-Based CK-Means Clustering Scheme for Smart Meter Data Analysis. Electronics, 14(10), 2074: 10.3390/electronics14102074

[33] Hang, Y., Yin, H., Hu, W., Zhong, L., & Ni, Y. (2025). Large-Scale Stream k-means based on Product-Quantized codes. International Journal of Machine Learning and Cybernetics, 1-14. 10.1007/s13042-025-02015-3

[34] Gu, Q., Niu, Y., Hui, Z., Wang, Q., & Xiong, N. (2025). A hierarchical clustering algorithm for addressing multi-modal multi-objective optimization problems. Expert Systems with Applications, 264, 12571010.1016/j.eswa.2025.125710

[35] Jing, D., Li, W., Han, L., Li, X., Li, L., Zhang, Y., ... & Xing, M. (2025). A Fast Satellite Selection Algorithm Based on Hierarchical Clustering and Iterative Subset Optimization. Remote Sensing, 17(5), 853. 10.3390/rs17050853

[36] Van Hau, P., McIver, T., Robertson, K., & Thaichon, P. (2025). Effective customer segmentation using the recency, frequency, and monetary framework, combined with density-based clustering. Journal of Strategic Marketing, 1 23.10.1080/0965254X.2025.2345678

[37] Sebai, D., & Shah, A. U. (2023). Semantic-oriented learning-based image compression by Only Train Once quantized autoencoders. Signal, Image and Video Processing, 17(1), 285-293. 10.1007/s11760-022-02203-9

[38] Rai, S., Ullah, A., Kuan, W. L., & Mustafa, R. (2025). An enhanced compression method for medical images using SPIHT encoder for fog computing. International Journal of Image and Graphics, 25(03), 2550025. 10.1142/S021946782550025X

[39] Ullah, A., Nawi, N. M., Aamir, M., Shazad, A., & Faisal, S. N. (2019). An improved multi-layer cooperation routing in visual sensor network for energy minimization. Int. J. Adv. Sci. Eng. Inf. Technol, 9(2), 664 670. 10.18517/ijaseit.9.2.7723

[40] Khan, F. S., Hasany, N., Altaf, A., & Khan, M. N. A. (2024). Benchmarking of an Enhanced Grasshopper for Feature Map Optimization of 3D and Depth Map Hand Gestures. Journal of Computing & Biomedical Informatics, 7(01), 256-263. → 10.47852/bonviewJCBI7202266

[41] Alam, T., Gupta, R., Ahamed, N. N., & Ullah, A. (2024). A decision-making model for self-driving vehicles based on GPT-4V, federated reinforcement learning, and blockchain. Neural Computing and Applications, 36(34), 21545-21560. 10.1007/s00521-024-09876-3

[42] Ali, M., Asghar, S., Mrhari, A., Ullah, A., Aleem, U. U., & Hassnae, R. (2024). A Hybrid CNN-BiLSTM-Based Approach for Roman Urdu Sentiment Analysis Using Enhanced Roman Urdu Corpus. International Journal of Asian Language Processing, 34(03n04), 2450009. 10.1142/S0129183124500098

[43] Le, Q. K., Angel, E., Tahi, F., & Postic, G. (2025). Semi-supervised segmentation of RNA 3D structures using density-based clustering. Computational and Structural Biotechnology Journal. 10.1016/j.csbj.2025.02.018

[44] Khan, S. N., Nawi, N. M., Imrona, M., Shahzad, A., Ullah, A., & Rahman, A. U. (2018). Opinion mining summarization and automation process: a survey. International Journal on Advanced Science, Engineering and Information Technology, 8(5), 1836.10.18517/ijaseit.8.5.6827

[45] Ullah, A., Razak, F. B. A., Hassnae, R., Yogarayan, S., Rehman, M. Z., Hameed, A., & Aznaoui, H. (2024). Analysis of Automated Melanoma Detection Utilizing Machine Learning and Deep Learning Techniques: A Review. International Journal of Image and Graphics, 2750012. 10.1142/S021946782750012X

[46] Ullah, A., Salam, A., El-Raoui, H., Sebai, D., & Rafie, M. (2022). Towards more accurate iris recognition system by using hybrid approach for feature extraction along with classifier. International Journal of Reconfigurable and Embedded Systems (IJRES), 11(1), 59 70. 10.11591/ijres.v11.i1.pp59-70

[47] Kim, S., Kim, D., Lee, J., Bateni, S. M., Ahmad, W., & Park, J. (2026). A novel approach of mapping snow disaster-prone areas based on areal disaster density optimization: a case study of South Korea. Natural Hazards, 122(2), 80. 10.1007/s11069-025-06789-4

[48] Song, L., Wang, B., Liu, Y., & Song, X. (2026). Enhancing power load forecasting accuracy under high renewable energy penetration with a MSN KAN framework: A10.1016/j.rineng.2026.109275novel approach to mitigate non-stationarity and enhance interpretability. Results in Engineering, 109275.