Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A HEURISTIC NOVEL APPROACH FOR DETERMINATION OF OPTIMAL EPSILON FOR DBSCAN CLUSTERING ALGORITHM

HASSAN SAYED RAMADAN[1], HUDA AMIN MAGHAWRY[2], MOHAMMED EL-ELEAMY[2], KHALED EL-BAHNASY[2]

[1,2]Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

E-mail: [1]hassanramadan@cis.asu.edu.eg, [2]huda_amin@cis.asu.edu.eg, [2]m.hamdy@cis.asu.edu.eg, [2]khaled.bahnasy@cis.asu.edu.eg

ABSTRACT

Clustering algorithms for identifying, and defining patterns between data elements. There are various types of clustering, and each type has different clustering algorithms. Each clustering algorithm has its applications, and requires parameter(s) to start the algorithm. These parameters are like a challenge to researchers to find the optimal values of parameters, or even close enough to get satisfying clustering results. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of these clustering algorithms. Epsilon is one of its parameters, and most of researchers choose it randomly. Therefore, the objective of this work is to propose a heuristic approach to find the optimal epsilon for DBSCAN clustering algorithm. The concept of this approach is to repeat DBSCAN many times, and each time it calculates a different epsilon value till it finds the optimal epsilon. Finding the optimal epsilon depends on evaluating clusters each time, and of course optimal epsilon has the best evaluation scores. Proposed approach uses the root mean square standard deviation (RMSSTD), and the R-squared (RS) to evaluate clusters. We run the proposed approach on three benchmark different dimensional datasets. Also, we used Silhouette index to validate the clustering results of the proposed approach. The proposed approach was successfully able to find the optimal epsilon for all three datasets.

Keywords: *DBSCAN, Optimal Epsilon, Clustering, RMSSTD, RS, Silhouette*

1. INTRODUCTION

Clustering is the process of dividing data points, or the population into a number of groups such that every group has data points more similar to other data points in the same group than those in other groups [1]. In simple words, the aim is to segregate groups with similar traits and assign them into clusters. Clustering has many types such as hierarchical clustering, density-based clustering, partitioning methods, fuzzy Clustering, and others [2]. For each type of clustering, there are many types of clustering algorithms. Each clustering algorithm has its advantages, drawbacks, and application areas. Also, each clustering algorithm requires parameter(s). Mostly, it is hard for researchers to determine the optimal values of the parameters. In this study, we will discuss a specific clustering algorithm. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is the pioneer algorithm [3] for density-based clustering, which contains large amount of data, which has outliers, and noise. DBSCAN has two parameters: epsilon (Eps: defines the radius of neighborhood around a point x), and minimum points (MinPts: the minimum number of neighbors within “eps” radius). Despite DBSCAN discovers clusters with arbitrary shape [4] and outliers [5], but it has certain limitations, that is not easy to determine proper initial values of Eps. Also, conventional DBSCAN cannot produce optimal Eps. Finding optimal Eps could be done manually that researchers keep trying different values till they got a result of clusters that satisfies them. This is possible, but with large datasets, it is harder to find the optimal Eps. Manually, it will cost long time, and big effort. So, in this paper, a heuristic approach is proposed to find the optimal Eps automatically. The aim of this study is to solve the challenge of finding the optimal Eps, which will save a lot of time, and will be more accurate than doing it manually. Also, this means getting the best division of data as clusters, and determination of the optimal number of clusters, which could be a parameter for other clustering algorithms. Novelty of our approach is shown in the usage of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

evaluation criteria RMSSTD, and RS in DBSCAN to make a decision to determine the value of the next Eps till finding the optimal Eps. Our validation depends on Silhouette index. We used Silhouette index as an evaluation of which objects lie well inside the cluster, and which do not. The silhouette score falls within the range [-1, 1]. The silhouette score of 1 means that the clusters are very dense, and nicely separated. The score of 0 means that clusters are overlapping. The score of less than 0 means that data belonging to clusters may be wrong/incorrect.

In this paper we will cover the previous studies of finding the optimal Eps in section 2. We provide detailed information about used datasets, and the mechanism of the proposed approach in section 3, followed by a discussion on the experimental results after applying the proposed approach, and its validation in section 4. We provide a conclusion of this article in section 5. Finally, we explain the ideas of the future work in section 6.

2. RALTED WORK

There many studies in the literature of cluster analysis. Many novel clustering algorithms, techniques, and approaches got proposed every year. Most studies focus on solving a specific problem facing researchers in clustering. Xu et al. [6] introduced a new clustering algorithm DBCLASD (Distribution-Based Clustering of LArge Spatial Databases) to discover clusters of points belonging to a spatial point process arises. This problem is common in many applications, and now it is covered with DBCLASD. Corizzo et al. [7] proposed the DENCAST system, which is a novel distributed algorithm. It performs density-based clustering and exploits the identified clusters to solve both single- and multi-target regression tasks. Also, there are several approaches in literature for finding suitable epsilon for DBSCAN. Giri et al. [8] proposed a new approach to determine an optimal epsilon (Eps) related to DBSCAN using empty circles in computational geometry. They extracted all the empty circles, then the collection of radii of empty circles are sorted into increasing order, and they have selected the knee / elbow value of those sorted radii as the value of epsilon parameter of DBSCAN. Karami et al. [9] proposed BDE-DBSCAN, which is a hybrid method with a combination of analytical DBSCAN and tournament selection method. This includes the concept of Binary Differential Evolution (BDE) and traditional DBSCAN clustering that helps to choose the parameters. Ren et al. [10] proposed another approach of modified DBSCAN named as Density Based Clustering Algorithm with Mahalanobis Metric (DBCAMM) where instead of Euclidean distance the Mahalanobis distance is used. Lai et al. [11] proposed a new method where they presented an optimization technique, multiverse optimizer algorithm (MVO) in which the parameters of DBSCAN algorithm are selected through iterative variable updating. The traditional computation process of most important parameter epsilon is calculated using k^{th} nearest neighbor's algorithm (hereafter KNN), where the knee value of the k^{th} nearest distances is considered as the value of epsilon parameter. On the other hand, the proposed approach is able to find the optimal Eps in DBSCAN clustering algorithm, and detect the optimal numbers of clusters heuristically using RMSSTD, and RS for evaluating clusters.

3. MATERIALS AND METHODS

3.1 Experimental Data

To validate the proposed approach, we need clustering datasets that the proposed approach should cluster optimally, and finds the optimal Eps for each one. Also, datasets should be in different dimensions, which will add more validation to the proposed approach, so the clustering datasets used in this study are artificial benchmark clustering datasets. They were derived from a GitHub project [12] that contains a collection of clustering problems that are popular and used many times in the literature. We used three clustering benchmark datasets. they have different sizes, and different attributes. First dataset is Hepta [13], which consists of 212 records, and each record has 3 attributes. Second dataset is Spherical_4_3 [14], which consists of 400 records, and each record has 3 attributes. Third dataset is Twenty [15], which consists of 1000 records, and each record has 2 attributes.

3.2 The Proposed Approach

The proposed approach is to repeat DBSCAN many times with evaluation criteria of clusters till it finds the optimal Eps. It starts with a random Eps, then it evaluates clusters using the evaluation criteria of RMSSTD, and RS [16]. For RMSSTD, a lower value indicates a better separation of clusters, and for RS, a higher value indicates a higher similarity between objects in the same cluster. After evaluation, it doubles the value of Eps, and calculates the evaluation score, then it gets the half value of Eps, and calculates the evaluation score. According to these three scores it decides the best direction to move in. Either the doubled value Eps, or the half value Eps, and it keeps repeating this process till it notice lower values of evaluation scores. This means it already found the optimal Eps, which has the lowest RMSSTD value, and highest RS value. Pseudo code of the proposed approach is shown in Algorithm 1.

Input:
1. A set of N objects as $D = \{O1, O2, . . ., On\}$
2. Eps
3. MinPts

Output: Optimal clustering of D

```
Begin
    Run DBSCAN on D with random Eps
    Run DBSCAN on D with Eps: Eps * 2
    Run DBSCAN on D with Eps: Eps / 2
    Calculate RMSSTD, and RS for each run
    NoProgress = 0
    WHILE NoProgress < 3 DO
        Choose the lowest RMSSTD
        IF Eps: Eps * 2 THEN
            Run DBSCAN with Eps: Eps * 2
        ELSE IF Eps: Eps /2 THEN
            Run DBSCAN with Eps: Eps / 2
        ELSE IF Eps: Eps THEN
            NoProgress = NoProgress + 1
            Run DBSCAN with Eps: Eps / 2
        ELSE
            Choose the highest RS
            IF Eps: Eps * 2 THEN
                Run DBSCAN with Eps: Eps * 2
            ELSE
                Run DBSCAN with Eps: Eps / 2
            END IF
        END IF
    END WHILE
End
```

Algorithm 1: Proposed Approach Of DBSCAN

We developed a tool of proposed approach in Python language. We run our tool on computer capabilities: Inter(R) Core(TM) i7-2600K CPU @3.40GHz, and 32.0 GB RAM.

4. EVALUATIONS AND DISCUSSIONS

We have applied the proposed approach on three clustering datasets.

The purpose of this test is to verify the ability of the proposed approach to find the optimal Eps, as well as to compare the performance of it with that of the improved Multi-Verse Optimizer algorithms, IMVO1, and IMVO2 [11].

MVO is a special variable updating method with excellent optimization performance. It has been improved for the optimization of the parameters of DBSCAN. MVO is able to find out the highest clustering accuracy of DBSCAN quickly. Also, it can find the interval of Eps corresponding to the highest accuracy. IMVO1 is denoted as an improvement of the optimization algorithm based on MVO, and IMVO2 is denoted as an improvement of the optimization algorithm based on IMVO1, and both algorithms can find the interval of Eps [Eps_{min}, Eps_{max}] [11].

4.1 Hepta

Hepta is an artificial dataset clustered to 7-clusters. Figure 1 shows a visualization of clusters of Hepta dataset.

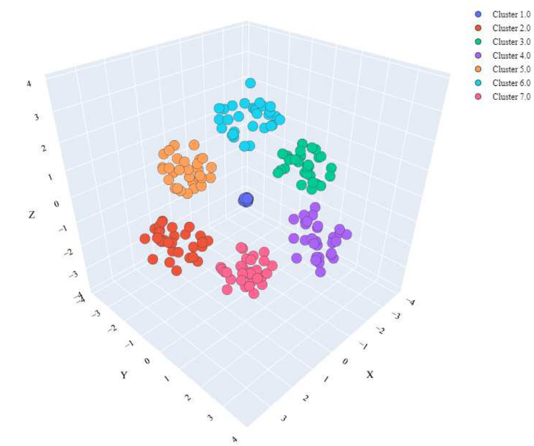


Figure 1: Visualization Of Clusters Of Hepta Dataset

The proposed approach started with a random Eps: 8, and MinPts: 4.

Then Figures 2(A), 2(B), 2(C), 2(D), 2(E), 2(F), 2(G), 2(H) show the visualization of clusters of iterations 1, 2, 3, 4, 5, 6, 7, and 8, respectively.

The Eps values of iterations 1,2,3,4 ,5,6,7, and 8 are 8, 16, 4, 2, 1, 0.5, 0.25, and 0.125, respectively.

Table 1 shows the evaluation score of each iteration of iterations of Hepta dataset.

It is observed in iterations 1, 2, and 3 that all clusters have the same value of RMSSTD: 1.6491, RS: 0, and also dataset was clustered in just 1-cluster. This means that Eps value is over the highest distance in the dataset, so the proposed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

approach cannot move higher by incrementing the Eps. This means it has just one way to find the optimal Eps by move lower by decrementing the Eps.

In iteration 4 the proposed approach already found the optimal Eps: 2 with the highest Silhouette score: 0.7019. You can observe that RMSSTD value of Eps is the lowest, and RS value is the highest. Also, it has to keep moving lower. According to the approach, it has to keep getting values of RMSSTD higher than the lowest RMSSTD the proposed approach got, to make sure it found the optimal Eps.

In iteration 5, the proposed approach got the same values in Eps: 1, so it kept moving lower. In iteration 6, the proposed approach got a higher RMSSTD value with Eps: 0.5 (with noise).

In iteration 7, the proposed approach got a higher RMSSTD value with Eps: 0.25 (with noise) than Eps: 0.5.

After three consecutive iterations, the proposed approach still keeps getting higher RMSSTD value with Eps: 0.125 (with noise) than Eps: 0.25. So, it stopped moving lower, cause in this line all RMSSTD values will keep getting higher, or will get fixed in one value higher than the lowest RMSSTD value in Eps: 1, and 2.

Table 1: Evaluation Of Clusters Of Hepta Dataset Iterations

#Iteration	Eps	RMSSTD	RS	Silhouette score	#Cluster	Noise
1	8	1.6491	0	0	1	No
2	16	1.6491	0	0	1	No
3	4	1.6491	0	0	1	No
4	2	0.4154	0.9383	0.7019	7	No
5	1	0.4154	0.9383	0.7019	7	No
6	0.5	0.9171	0.7098	0.3109	13	Yes
7	0.25	1.6405	0.0198	-0.1106	2	Yes
8	0.125	1.653	0.0001	-0.0697	1	Yes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

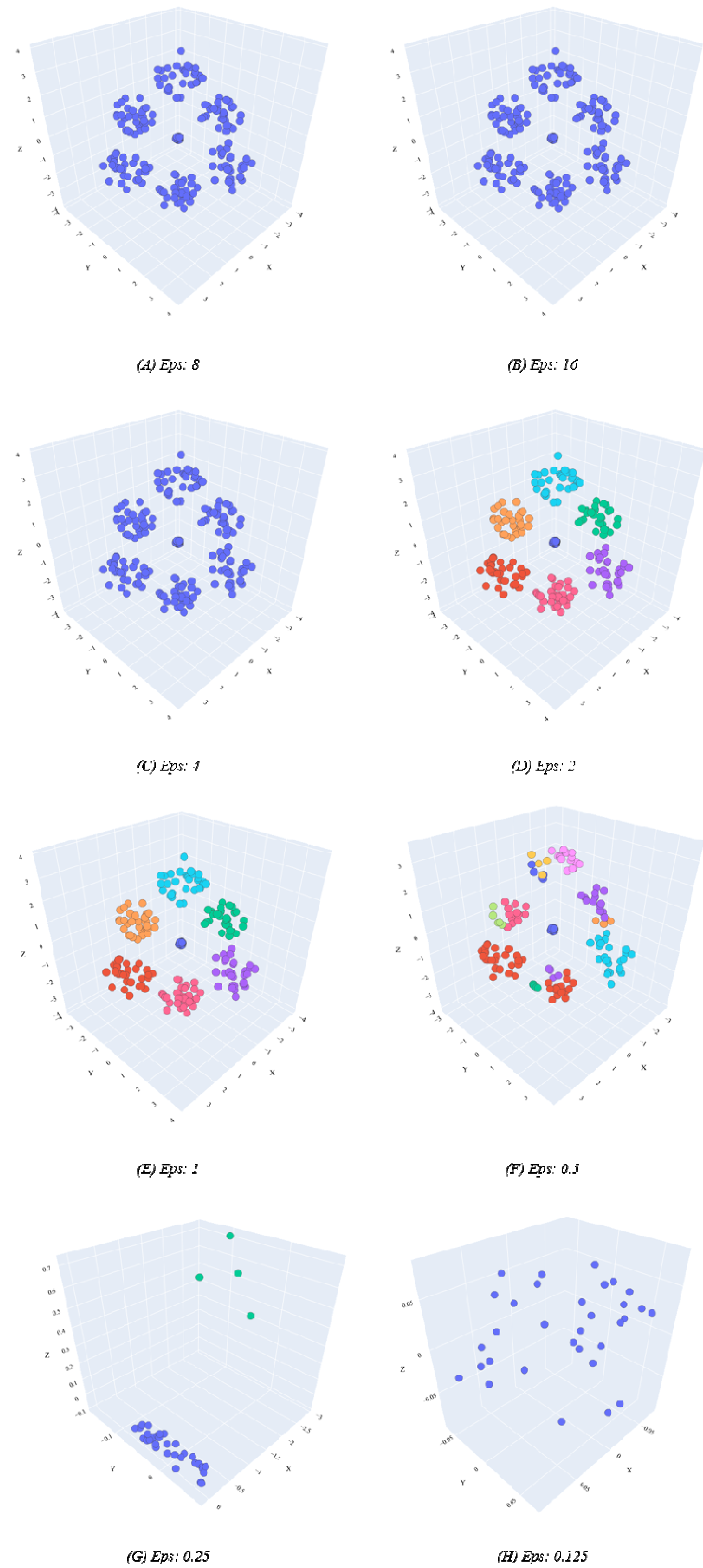


Figure 2: Visualization Of Iterations Of Hepta Dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 3, it is observed that at the optimal Eps: 1, and 2, RMSSTD is the lowest value in the blue line, and RS is the highest value in the red line.

This proof the efficiency of the proposed approach, and values represents the concept of the approach.

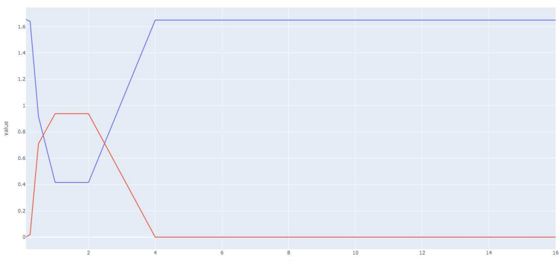


Figure 3: Chart Of Values Of RMSSTD, And RS Of Iterations Of Hepta Dataset

The proposed approach was able to meet the research objectives with Hepta dataset. It was able to find the optimal Eps in just 8 iterations heuristically, and accurately.

4.2 Spherical_4_3

Spherical_4_3 is an artificial dataset clustered to 4-clusters. Figure 4 shows a visualization of clusters of Spherical_4_3 dataset.

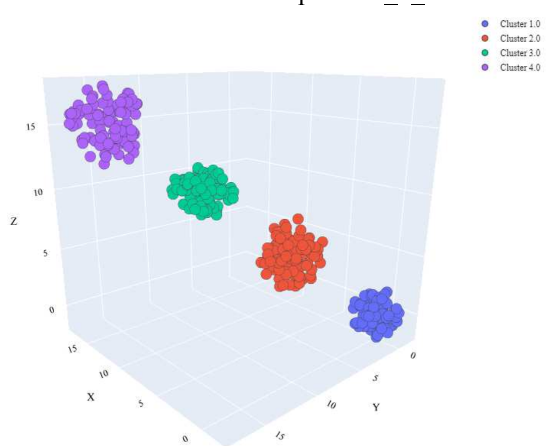


Figure 4: Visualization Of Clusters Of Spherical_4_3 Dataset

The proposed approach started with a random Eps: 20, and MinPts: 4.

Then Figures 5(A), 5(B), 5(C), 5(D), 5(E), 5(F), 5(G), 5(H) show the visualization of clusters of iterations 1, 2, 3, 4, 5, 6, 7, and 8, respectively.

The Eps values of iterations 1, 2, 3, 4, 5, 6, 7,and 8 are 20, 40, 10, 5, 2.5, 1.25, 0.625, and 0.3125, respectively.

Table 2 shows the evaluation score of clusters of each iteration of Spherical_4_3 dataset.

It is observed in iterations 1, 2, 3, and 4 that all clusters have the same value of RMSSTD: 5.7128, RS: 0, and also dataset was clustered in just 1-cluster. As mentioned before, the proposed approach cannot move higher by incrementing the Eps. it will move lower by decrementing the Eps to find the optimal Eps.

In iteration 5, the proposed approach already found the optimal Eps: 2.5 with the highest Silhouette score: 0.6894. It is observed that RMSSTD value of Eps is the lowest, and RS value is the highest. Also, it has to keep moving lower. According to the approach, it has to keep getting values of RMSSTD higher than the lowest RMSSTD the proposed approach got, to make sure it found the optimal Eps.

In iteration 6, the proposed approach got a higher RMSSTD value with Eps: 1.25 (with noise).

In iteration 7, the proposed approach got a higher RMSSTD value with Eps: 0.625 (with noise) than Eps: 1.25.

After three consecutive iterations, the proposed approach still keeps getting a higher RMSSTD value with Eps: 0.3125 (with noise) than Eps: 0.625. So, it stopped moving lower, cause in this line all RMSSTD values will keep getting higher, or will get fixed in one value higher than the lowest RMSSTD value in Eps: 2.5.

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Evaluation Of Clusters Of Spherical_4_3 Dataset Iterations

#Iteration	Eps	RMSSTD	RS	Silhouette score	#Cluster	Noise
1	20	5.7128	0	0	1	No
2	40	5.7128	0	0	1	No
3	10	5.7128	0	0	1	No
4	5	5.7128	0	0	1	No
5	2.5	1.1402	0.9605	0.6894	4	No
6	1.25	1.2385	0.9535	0.6442	4	Yes
7	0.625	4.2778	0.473	-0.3022	24	Yes
8	0.3125	5.7184	0.0031	-0.48	2	Yes

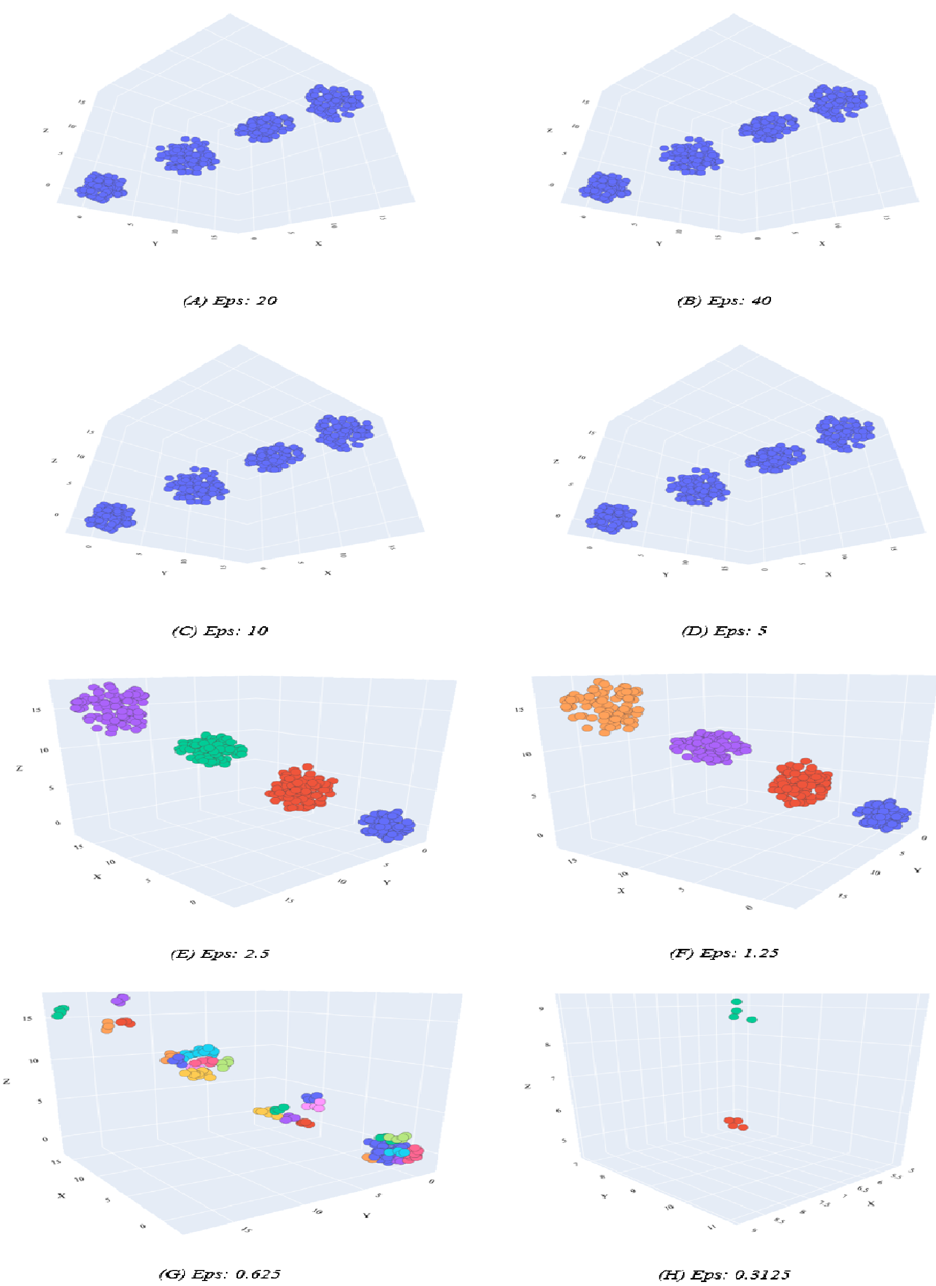


Figure 5: Visualization Of Iterations Of Spherical_4_3 Dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 6, it is observed that at the optimal Eps: 2.5, RMSSTD is the lowest value in the blue line, and RS is the highest value in the red line.

This proof the efficiency of the proposed approach, and values represents the concept of the approach.

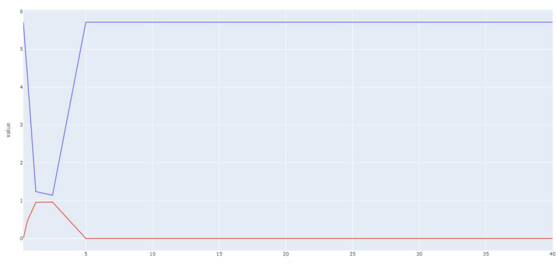


Figure 6: Chart Of Values Of RMSSTD, And RS Of Iterations Of Spherical_4_3 Dataset

The proposed approach was able to meet the research objectives with Spherical_4_3 dataset. It was able to find the optimal Eps in just 8 iterations heuristically, and accurately.

4.3 Twenty

Twenty is an artificial dataset clustered to 20-clusters. Figure 7 shows a visualization of clusters of Twenty dataset.

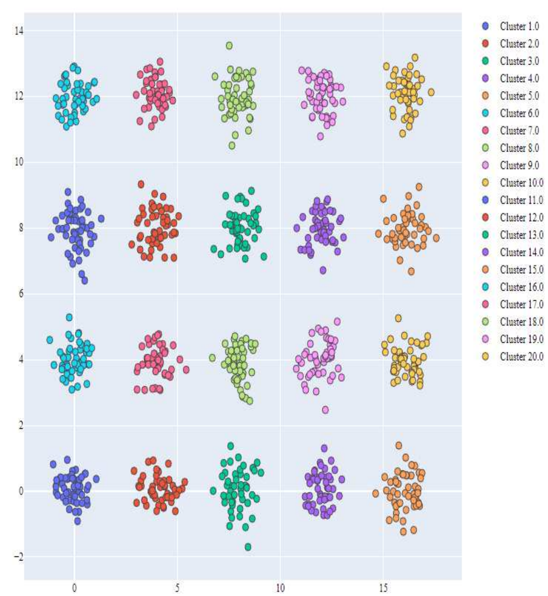


Figure 7: Visualization Of Clusters Of Twenty Dataset

The proposed approach started with a random Eps: 4, and MinPts: 4.

Then Figures 8(A), 8(B), 8(C), 8(D), 8(E), 8(F), 8(G) show the visualization of clusters of iterations 1, 2, 3, 4, 5, 6, and 7, respectively.

The Eps values of iterations 1, 2, 3, 4, 5, 6, and 7 are 4, 8, 2, 1, 0.5, 0.25, and 0.125, respectively.

Table 3 shows the evaluation score of clusters of each iteration of Twenty dataset.

It is observed in iterations 1, and 2 that all clusters have the same value of RMSSTD: 5.1368, RS: 0, and also dataset was clustered in just 1-cluster. As mentioned before, the proposed approach cannot move higher by incrementing the Eps. it will move lower by decrementing the Eps to find the optimal Eps.

In iteration 3, the proposed approach got a lower RMSSTD value with Eps: 2. It getting a better clustering.

In iteration 4, the proposed approach already found the optimal Eps: 1 with the highest Silhouette score: 0.749. You can observe that RMSSTD value of Eps is the lowest, and RS value is the highest. Also, it has to keep moving lower. According to the approach, it has to keep getting values of RMSSTD higher than the lowest RMSSTD the proposed approach got, to make sure it found the optimal Eps.

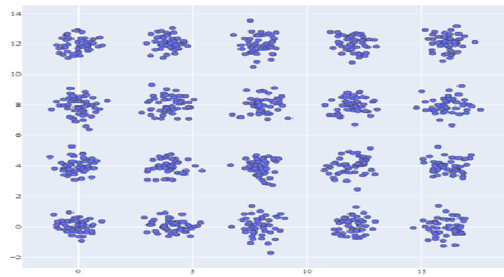
In iteration 5, the proposed approach got a higher RMSSTD value with Eps: 0.5 (with noise).

In iteration 6, the proposed approach got a higher RMSSTD value with Eps: 0.25 (with noise) than Eps: 0.5.

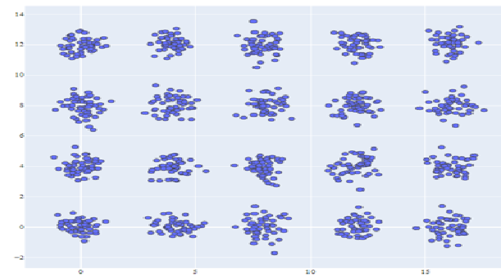
After three consecutive iterations, the proposed approach still keeps getting a higher RMSSTD value with Eps: 0.125 (with noise) than Eps: 0.25. So, it stopped moving lower, cause in this line all RMSSTD values will keep getting higher, or will get fixed in one value higher than the lowest RMSSTD value in Eps: 1.

Table 3: Evaluation Of Clusters Of Twenty Dataset Iterations

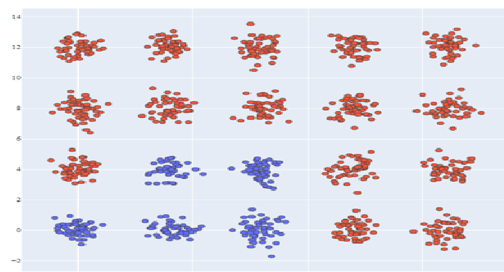
#Iteration	Eps	RMSSTD	RS	Silhouette score	#Cluster	Noise
1	4	5.1368	0	0	1	No
2	8	5.1368	0	0	1	No
3	2	4.6316	0.1878	0.2014	2	No
4	1	0.495	0.9909	0.749	20	No
5	0.5	0.9902	0.9636	0.6893	21	Yes
6	0.25	2.9813	0.6817	0.0436	55	Yes
7	0.125	4.8683	0.1234	-0.6463	24	Yes



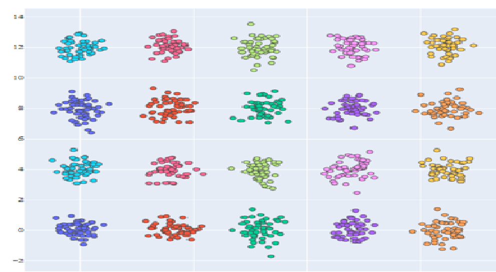
(A) Eps: 4



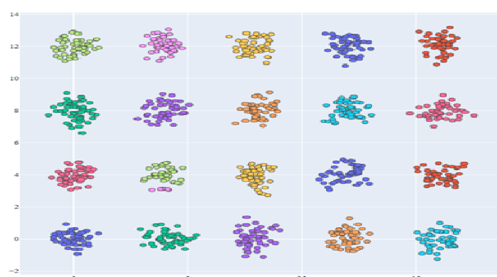
(B) Eps: 8



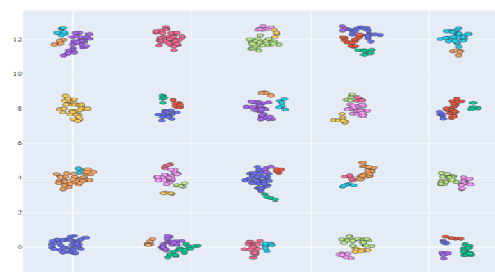
(C) Eps: 2



(D) Eps: 1



(E) Eps: 0.5



(F) Eps: 0.25

(G) Eps: 0.125

Figure 8: Visualization Of Iterations Of Twenty Dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 9, it is observed that at the optimal Eps: 1, RMSSTD is the lowest value in the blue line, and RS is the highest value in red the line. This proof the efficiency of the proposed approach, and values represents the concept of the approach.

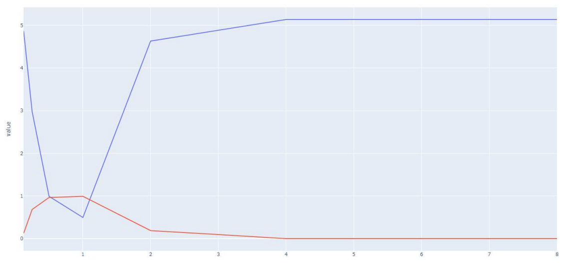


Figure 9: Chart Of Values Of RMSSTD, And RS Of Iterations Of Twenty Dataset

The proposed approach was able to meet the research objectives with Twenty dataset. It was able to find the optimal Eps in just 7 iterations heuristically, and accurately.

As noticed, the proposed approach was successfully able to find the optimal Eps, and divide all datasets into the optimal clusters. So, it is clear how the proposed approach efficient, rapid, and reliable.

Next, another validation of the performance of the proposed approach. This will be by comparing its Eps results with IMVO1, and IMVO2 results. For IMVO1, and IMVO2, we let value of MinPts=4 as a constant value for each dataset. Then, we run IMVO1 for 400 iterations, and IMVO2 for 200 iterations to find the value range of Eps corresponding to the highest clustering accuracy of DBSCAN for each dataset.

The comparison of Eps results of each dataset with every algorithm are shown in Table 4.

IMVO1, and IMVO2 intervals ranges contain the optimal Eps value of the proposed approach. IMVO1, and IMVO2 intervals ranges contain the optimal Eps value of the proposed approach.

IMVO1, and IMVO2 were able to find the optimal Eps value, or a near value to the optimal Eps, which confirms the efficiency, and accuracy of the proposed approach.

Also, the proposed approach was able to find the optimal Eps in 7, or 8 iterations not 200, nor 400 iterations. This is another validation for the high performance of the proposed approach

Table 4: Comparison of Eps results correspond to the optimal clustering accuracy

Algorithm	Hepta	Spherical_4_3	Twenty
Proposed Approach	1, or 2	2.5	1
IMVO1 ([Epsmin, Epsmax])	[0.535874, 2.85785]	[1.72716, 2.88271]	[0.562737, 1.68126]
IMVO2 ([Epsmin, Epsmax])	[0.517368, 2.85867]	[1.71547, 2.88833]	[0.559487, 1.70573]

5. CONCLUSION AND FUTURE WORK

DBSCAN is one the most efficient clustering algorithms, but to get accurate clustering you need to find the right values of its parameters. This creates great difficulties for manually setting parameters. For example, in dataset D1, when MinPts=4, the optimal Eps 1, which is a challenging to search manually specially with large datasets.

So, the determination of the optimal epsilon for DBSCAN is the aim of this study. We proposed a heuristic novel approach that can find the optimal epsilon for DBSCAN rapidly, and accurately. The concept of this approach is to repeat DBSCAN several times until it finds the optimal epsilon. Each time the proposed approach repeats DBSCAN, it uses the root mean square standard deviation (RMSSTD) and R-Squared (RS) to evaluate cluster. The experimental results show that the proposed approach not only can find the optimal Eps quickly, and effectively, but also is validated as a more efficient approach than previous studies.

For the previously mentioned example, if we run our approach with a random Eps: 1024, the proposed approach will find the optimal Eps in just 15 iterations.

As noticed the proposed approach is accurate, heuristic, and rapid with various datasets with different dimensions.

In future work, we will focus on biological scope by applying the proposed approach for gene expression data, and clustering genes, and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

conditions with DBSCAN. Our aim will be to proof the ability of the proposed approach to find the optimal epsilon efficiently and rapidly using many different datasets from different scopes. Also, we will develop a python package for the proposed approach for researchers.

REFERENCES:

[1] Clustering in machine learning [Internet]. GeeksforGeeks. 2021 [cited 2021Dec18]. Available from: https://www.geeksforgeeks.org/clustering-in-machine-learning

[2] Prasad Pby S, Prasad S, 45 PV, 45. Types of clustering algorithms in machine learning with examples [Internet]. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. 2021 [cited 2021Dec18]. Available from: https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms/

[3] Mahesh Kumar K, Rama Mohan Reddy A. A fast DBSCAN clustering algorithm by accelerating neighbor searching using groups method. Pattern Recognition. 2016;58:39–48.

[4] Chen Z, Li YF. Anomaly detection based on enhanced DBSCAN algorithm. Procedia Engineering. 2011;15:178–82.

[5] Shah GH. An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets. 2012 Nirma University International Conference on Engineering (NUiCONE). 2012;

[6] Xiaowei Xu, Ester M, Kriegel H-P, Sander J. A distribution-based clustering algorithm for mining in large spatial databases. Proceedings 14th International Conference on Data Engineering.

[7] Corizzo R, Pio G, Ceci M, Malerba D. DENCAST: Distributed density-based clustering for multi-target regression. Journal of Big Data. 2019;6(1).

[8] Giri K, Biswas T K. Determining Optimal Epsilon (eps) on DBSCAN using Empty Circles, AISE 2020. 2021.

[9] Karami A, Johansson R. Choosing DBSCAN parameters automatically using differential evolution. International Journal of Computer Applications. 2014;91(7):1–11.

[10] Ren Y, Liu X, Liu W. DBCAMM: A novel density based clustering algorithm via using the Mahalanobis metric. Applied Soft Computing. 2012;12(5):1542–54.

[11] Lai W, Zhou M, Hu F, Bian K, Song Q. A new DBSCAN parameters determination method based on improved MVO. IEEE Access. 2019;7:104085–95.

[12] Deric. Deric/clustering-benchmark [Internet]. GitHub. [cited 2021Dec18]. Available from: https://github.com/deric/clustering-benchmark

[13] Deric. Clustering-benchmark/hepta.arff at master · Deric/clustering-benchmark [Internet]. GitHub. 2015 [cited 2021Dec18]. Available from: https://github.com/deric/clustering-benchmark/blob/master/src/main/resources/datasets/artificial/hepta.arff

[14] Deric. Clustering-benchmark/SPHERICAL_4_3.ARFF at master · Deric/clustering-benchmark [Internet]. GitHub. 2015 [cited 2021Dec18]. Available from: https://github.com/deric/clustering-benchmark/blob/master/src/main/resources/datasets/artificial/spherical_4_3.arff

[15] Deric. Clustering-benchmark/twenty.arff at master · Deric/clustering-benchmark [Internet]. GitHub. [cited 2021Dec18]. Available from: https://github.com/deric/clustering-benchmark/blob/master/src/main/resources/datasets/artificial/twenty.arff

[16] Rujasiri P, Chomtee B. Comparison of Clustering Techniques for Cluster Analysis. Kasetsart J. (Nat. Sci.). 2009;43;378-388.