

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/354622722>

Develop a dynamic DBSCAN algorithm for solving initial parameter selection problem of the DBSCAN algorithm

Article in Indonesian Journal of Electrical Engineering and Computer Science · September 2021

DOI: 10.11591/ijeecs.v23.i3.pp1602-1610

CITATION

1

READS

401

5 authors, including:



Md Zakir Hossain

Dhaka University of Engineering & Technology

7 PUBLICATIONS 96 CITATIONS

[SEE PROFILE](#)



Md. Waliur Rahman Miah

Dhaka University of Engineering & Technology

13 PUBLICATIONS 38 CITATIONS

[SEE PROFILE](#)



Jahid Hasan Rony

Dhaka University of Engineering & Technology

7 PUBLICATIONS 20 CITATIONS

[SEE PROFILE](#)



Momotaz Begum

Dhaka University of Engineering & Technology

37 PUBLICATIONS 75 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Data mining techniques for noisy and imbalanced data [View project](#)

Develop a dynamic DBSCAN algorithm for solving initial parameter selection problem of the DBSCAN algorithm

Md. Zakir Hossain, Md. Jakirul Islam, Md. Waliur Rahman Miah, Jahid Hasan Rony,
Momotaz Begum

Department of Computer Science and Engineering, Dhaka University of Engineering and Technology, Gazipur,
Bangladesh

Article Info

Article history:

Received Oct 14, 2020
Revised Jul 11, 2021
Accepted Jul 18, 2021

Keywords:

Arbitrary shape
DBSCAN
Eps
Initial parameter
MinPts
Outlier

ABSTRACT

The amount of data has been increasing exponentially in every sector such as banking securities, healthcare, education, manufacturing, consumer-trade, transportation, and energy. Most of these data are noise, different in shapes, and outliers. In such cases, it is challenging to find the desired data clusters using conventional clustering algorithms. DBSCAN is a popular clustering algorithm which is widely used for noisy, arbitrary shape, and outlier data. However, its performance highly depends on the proper selection of cluster radius (*Eps*) and the minimum number of points (*MinPts*) that are required for forming clusters for the given dataset. In the case of real-world clustering problems, it is a difficult task to select the exact value of *Eps* and *MinPts* to perform the clustering on unknown datasets. To address these, this paper proposes a dynamic DBSCAN algorithm that calculates the suitable value for *Eps* and *MinPts* dynamically by which the clustering quality of the given problem will be increased. This paper evaluates the performance of the dynamic DBSCAN algorithm over seven challenging datasets. The experimental results confirm the effectiveness of the dynamic DBSCAN algorithm over the well-known clustering algorithms.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Md. Jakirul Islam
Department of Computer Science and Engineering
Dhaka University of Engineering and Technology, Gazipur, Bangladesh
Email: jakirduet@duet.ac.bd

1. INTRODUCTION

In the last decades, computer and information technology have been developed rapidly. As a result, the data volume has been increased massively in science and engineering, and it will be increased constantly on a large scale [1]. The increase of data from terabytes to pet bytes changes science and engineering, transforms different organization from data-poor to increasingly data-rich. This cause to develop a significant number of algorithms in the field of data clustering.

Clustering is the key to find accurate data patterns of large datasets [2]. Clustering has been used in many applications, including marketing, networking, image processing, biology, geographic observation, web analysis, and medical-based applications [3]. Clustering plays a significant role in allowing automatic identification of unlabeled records by grouping them into clusters based on similarity measurements [4].

Recently, several researchers have been focusing on the development of different clustering algorithms, such as BIRCH [5], DBSCAN [6], k-nearest neighbor [7], mini-batch k-means [7], [8], k-means [7], [9], OPTICS [10], and GMM [11]. These clustering algorithms are recognized in different

categories: centroid-based clustering, graph-based clustering, partitioning clustering and density-based clustering [12].

In this research, we focus on the density-based clustering algorithm, which mainly depends on the notion of density. In the literature, the density-based clustering method describes clusters according to the high-density regions which are separated from the low-density regions [1]. The clusters continue growing until the density exceeds a certain threshold [13]. According to that, the density-based clustering algorithm has an advantage in creating groups with arbitrary shapes.

The DBSCAN is a popular density-based clustering algorithm that goal at identifying the high dense region to form clusters and low areas to separate them. The DBSCAN algorithm mainly focuses on minimizing the number of input parameters. It discovers clusters efficiently without using specifying the number of clusters [14]. It has been successfully applied in different fields including medical images, satellite images, anomaly detection in temperature data, and GPS data [15]. However, the performance of this algorithm is highly dependent on two user-defined parameters as shown in Figure 1, namely *Eps* and *MinPts*. For example, if the large value is chosen for *Eps*, then DBSCAN forms the clusters with dissimilar data. On the other hand, if a small value is chosen for *Eps*, then this technique forms the clusters having a small amount of similar data. In such cases, the selection of *Eps* and *MinPts* is a challenging task for performing clustering on real-world datasets, which is the main concern of this research. Although, many researchers have been focusing on improving the performance of the DBSCAN algorithm using either selecting the initial value of *Eps* or *MinPts* dynamically [16]-[21]. The literature shows that, there is no research works have been taken initiative to select the initial value of both *Eps* and *MinPts* dynamically. Therefore, this paper proposes a dynamic DBSCAN algorithm that selects the initial value of both *Eps* and *MinPts* dynamically, hoping that good quality clusters with better accuracy will be found at the end of the run.



Figure 1. Initial parameter *Eps* and *MinPts*

The organization of this paper is as follows. In Section 2, the detail of DBSCAN algorithm is presented. In Section 3, the works related to this research are described. In Section 4, the detail of the proposed dynamic DBSCAN algorithm is described. Section 5 demonstrates the experimental results provided by the proposed and the well-known data clustering methods. Section 6 describes the conclusion and future work of this research.

2. RELATED WORK

Clustering is considered for preparing an efficient unsupervised analysis through the data. In this regard, several clustering algorithms have developed in the last decades. In=t can be seen that, these algorithms have experienced difficulties when the datasets are unstructured, noise, different in shapes and sizes, and densities. The DBSCAN algorithm is recognized as a popular density-based clustering algorithms which can perform clustering on the datasets with different sizes and shapes [7]. However, its performance highly depends on the value of *Eps* and the value of *MinPts*.

Several researchers have been conducted for the improvement of DBSCAN algorithm. For example, in [16], a new heuristic is proposed as a distance measuring method to perform the clustering in multi-density datasets. In [17], an improved DBSCAN algorithm called soft DBSCAN where fuzzy set theory is applied to perform the clustering on such datasets. In [18], a new DBSACN clustering algorithm is proposed for normalized datasets which are proven to be an efficient clustering approach. However, this method unable to resolve the problem of selecting input parameters as in the DBSCAN algorithm. In [19], another DBSCAN algorithm based on graph-based index structure is proposed for performing clustering on high-dimensional. In [20], the DBSCAN algorithm is implemented in a distributed system. The distributed systems apply in big datasets to process the data in a very fast-growing way. Parallelization also uses in the distributed system. In [21], an improved DBSCAN algorithm is proposed to reduce the computational time of the existing DBSCAN algorithm. This algorithm can divide data space into gride and then optimizes the computational

cost by reducing unnecessary query operations of the DBSCAN algorithms. In [22], the partitioning technique is combined with DBSCAN to select the suitable value of the user-defined parameters of DBSCAN algorithm. However, this method has not evaluated against the datasets with different densities. The authors of [23] have incorporated the gaussian-means into the DBSCAN algorithm to improve the selection of the value of DBSCAN parameters. However, this method could not show better clustering performance against dense datasets. In [24], the binary differential evaluation concept is incorporated into the DBSCAN algorithm to effectively determine the value of DBSCAN parameters. In [25], an improved DBSCAN clustering algorithm namely DBSCAN-KNN-GA is proposed by adopting the k-nearest neighborhood concept (KNN) and genetic algorithm (GA) into DBSCAN algorithms which finds the value of Eps and $MinPts$ dynamically. It can be observed that all of these algorithms have difficulties to select the user-defined parameters of DBSCAN algorithm. Therefore, this study proposes a dynamic DBSCAN algorithm for non-spherical clustering. The proposed technique does not rely on any predefined parameters as the original DBSCAN technique. The proposed method calculates the best Eps and $MinPts$ dynamically, which improves the cluster quality.

3. THE DBSCAN CLUSTERING ALGORITHM

The DBSCAN algorithm aims to identify the high dense region clusters by separating them from low dense regions. It has two user-defined parameters namely Eps and $MinPts$ which are required to find the clusters of datasets. The main idea of the DBSCAN is that each point in the dataset is scanned to determine the nearest neighbors within a particular distance [17]. To determine a core-point, the number of neighbors should exceed the threshold. Otherwise, it might be a border-point or a noise. Before providing the idea of the DBSCAN algorithm, some notions need to be defined first. The notions are,

- **Directly density-reachable:** A point q is said to be directly density-reachable from a point p , if and only if the point q is in the Eps neighborhood of point p , as shown in Figure 2(a).
- **Density-reachable:** A point q is density-reachable from point p because there exists a sequence q_1, q_2, \dots, q_n with $q_1=p, q_2=p, \dots, q_n=q$. The point's q_{i+1} is directly density-reachable from q_i , as shown in Figure 2(b).
- **Density connected:** A point q is density connected to a point p with respect to Eps and $MinPts$ if there is a point f belongs to a dataset such that both q and p are density-reachable from f with respect to Eps and $MinPts$, as shown in Figure 2(c).
- **Core point:** A data point p is a core point if the number of points in its Eps neighborhood is greater than or equal to the $MinPts$, as shown in Figure 2(d).
- **Border point:** A point is called border point if it is a part of a cluster and not dense themselves, as shown in Figure 2(d).
- **Noise point:** A point is called noise point if it does not belong to any cluster, as shown in Figure 2(d).
- **Cluster:** A cluster C is a non-empty subset of the data set, as shown in Figure 2(d).

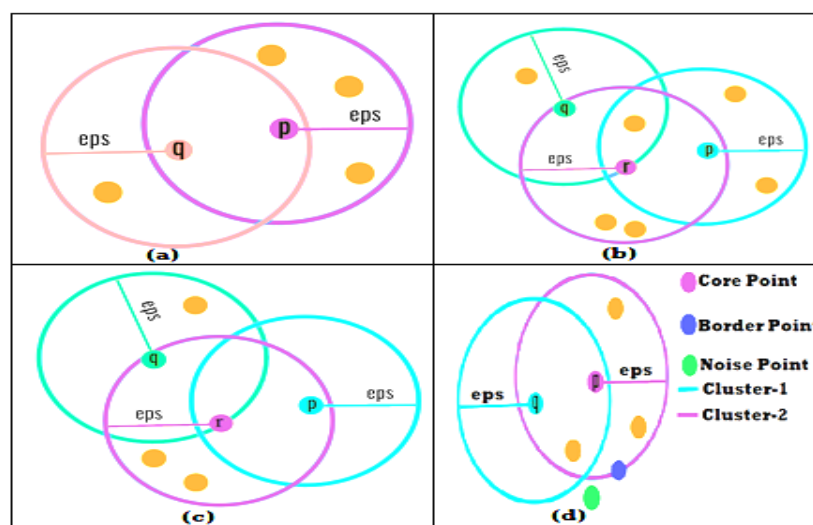


Figure 2. DBSCAN algorithm notions ($Eps = 2$, $MinPts = 4$), (a) directly density-reachable, (b) density reachable, (c) density connected, and (d) core, border, noise point and cluster

In DBSCAN algorithm, the core point is used to make primary clusters. If all core points are density-connected from one another, then all primary clusters merged to the same cluster. After formation of the primary clusters, the rest of the data point will be treated as either border points or noise points. The existing DBSCAN algorithm is demonstrated in Algorithm 1.

Algorithm 1 Psodocode of DBSCAN (X , eps , $minpts$)

Input: X -a set of unvisited points, eps -the distance threshold, and $minpts$ -the minimum number of points

Output: A set of clusters.

```

1.  foreach  $x$  in  $X$  do
2.     $x := \text{MarkedAsVisited}(\text{True})$ ;
3.     $N \leftarrow \text{ReturnNeighbors}(x, eps)$ ;
4.    if ( $|N| < minpts$ ) then
5.       $x := \text{MarkedAsNoise}(\text{True})$ ;
6.    Else
7.       $C \leftarrow \{x\}$ ;
8.      foreach  $x' \text{ in } N$  do
9.         $N := N \setminus x'$ ;
10.       if ( $\neg \text{Visited}(x')$ ) then
11.          $x := \text{MarkedAsVisited}(\text{True})$ ;
12.          $N' \leftarrow \text{ReturnNeighbors}(x', eps)$ ;
13.         if ( $|N'| \geq minpts$ ) then  $N \leftarrow N \cup N'$ ; end if
16.        if ( $x' \text{ not in } C$ ) then  $C \leftarrow C \cup \{x'\}$ ; end if
19.      end if
20.    end foreach
22.  end if
23. end foreach

```

4. THE PROPOSED ALGORITHM

The main purpose of the research is to determine the best value for Eps and $MinPts$ dynamically which will improve the cluster quality. For the explanation purpose, we consider the input data points are spread in a two-dimensional (x , y) search space with the uniform distribution. In the first stage, we calculate a distance matrix D using these points, as shown in Figure 3.

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \dots & d_{nn} \end{bmatrix}$$

Figure 3. Matrix calculated from assumed data points

Each element d_{jk} (euclidian distance from data point j to data point k) of this matrix can be calculated using (1).

$$d_{jk} = \sqrt{(P_i - P_{i+1})^2} \quad (1)$$

In the second stage, we calculate the mean distance M for each row of distance matrix D using the (2).

$$m_i = \sum_{j=1}^n d_{jk} \quad (2)$$

The mean distance M is calculated as follows,

$$M = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ \vdots \\ m_n \end{bmatrix}$$

In the third stage, we find the minimum mean value from M using the following (3). This found minimum mean value is considered to be the closest nearest neighborhood point of the i^{th} data point.

$$m_{\min} = \min(m_i) \text{ where } i \leq n \quad (3)$$

Here, the i^{th} data point is the highest density point which is denoted by $h_{(x,y)}$. In this paper, the highest density point $h_{(x,y)}$ is the key to select the best value of Eps and $MinPts$, respectively.

In the fourth stage, we calculate Manhattan distance from the highest density point h to other data point P by the following (4).

$$md_i = |h_i - P_i| \text{ where } i \leq n-1 \quad (4)$$

In the fifth stage, we set the value of $Eps = 0.5$ and then count the number of data points that lies within Eps . After that we increase the value of Eps by 0.5 and then again count the number of data points with in Eps . This process continues until a local maximum is reached. The whole idea is presented in Figure 4 where the value of Eps_1, Eps_2, Eps_3 and Eps_4 is 0.5, 1.0, 1.5, and 2.0, respectively. It can be observed that Eps_1, Eps_2, Eps_3 and Eps_4 contain three, four, five and three data points, respectively. Here, the Eps_3 is the local maximum, as shown in Figure 5. In the sixth stage, we select Eps_4 as the desired Eps and corresponding number of data points as the desired $MinPts$ of the proposed DBSCAN algorithm shows in Figure 5.

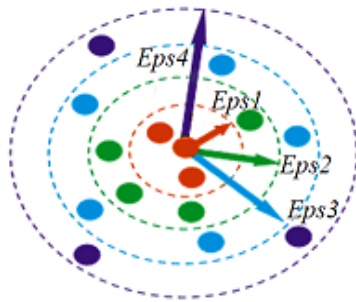


Figure 4. Data density in the different Eps

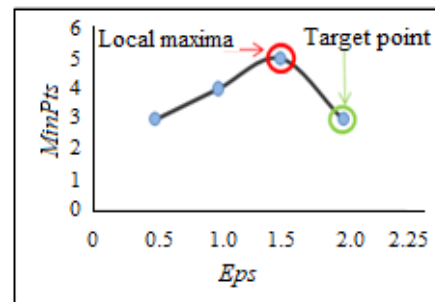


Figure 5. Relationship between Eps and $MinPts$

In the final stage, we calculate Manhattan distance from the high-density point h to other data points P of the dataset. After that we find the core points and boundary points based on the found Eps and $MinPts$. This process will continue for all the unvisited data points. The proposed algorithm is demonstrated in Algorithm 2.

5. RESULT AND ANALYSIS

5.1. Data sets and experimental setup

For the experimental purpose, seven 2D artificial datasets are considered. These datasets are half Ring, three spirals, corners, semicircular, half moon, half kernel, and aggregation. These datasets have different densities including clusters inside clusters, multi-density, connected clusters, and well-separated densities.

Algorithm 2 The Proposed DDBSCAN Clustering Algorithm

```

1. Calculate Euclidean distance matrix  $D$  from the dataset;
2. Calculate the mean of each row of  $D$ . Find the minimum mean value among all the
   mean of row of  $D$ . This minimum mean is the high-density point which is considered
   to the center point  $P_{center}$ ;
3. Set  $r = 0.5$ ;
4. Set  $P_{count}[0] = 0$ ;
5. for  $i$ : =1 to  $N$  do
6.   for  $j$ : = 1 to  $N$  do
7.     calculating the Manhattan distance  $d$  from  $P_{center}$  to  $P_i$ ;
8.     if ( $d < r$ ) then  $P_{count}[i] :=$  count the number of points within  $r$ ; end if
9.   end for
10.  if ( $P_{count}[i] > P_{count}[i-1]$ ) then  $r := r + 0.5$ ; continue;
11.  Else
12.    calculate  $diff_i := |P_{count}[i] - P_{count}[i-1]|$ ;
13.  end if
14.   $MinPts := diff_i$ ;
15.   $Eps := r$ ;
16. end for
17. Repeat
18.   for  $i$ : =1 to  $N$  do
19.     for  $j$ : =1 to  $N$  do
20.       calculate distance  $d_{ij}$ ;
21.       if ( $d_{ij} \leq Eps$ ) then
22.         if ( $d_{ij} = Eps$ ) then this is the boundary point and count  $C$ ;
23.       Else
24.         this is not the boundary point and count  $C$ ;
25.       end if
26.     end if end for
27.   end for
28.   if ( $MinPts \leq C$ ) then
29.     this is the core point  $P_{core}$ . Calculate another core point  $P_{core}$  and make cluster
       with
       boundary point and mark visited;
30.   Else
31.     continue;
32.   end if
33. until all the data points are not visited.

```

These datasets allow examine the performance of the dynamic DBSCAN algorithm over the well-known density-based clustering algorithms. The properties of these datasets are summarized in Table 1. All the experiments were performed on Intel Core i5 2.4GHz processor with 4GBRAM on the platform Microsoft Windows 10. We implement the proposed dynamic DBSCAN algorithm using C++ and artificial data sets using MATLAB.

Table 1. Summary of used datasets

<i>Data Set</i>	<i>Size</i>	<i>Classes</i>
Corners	1000	4
Three Spirals	312	3
Half Ring	373	2
Semi Circular	1000	2
Half Moon	1000	2
Half Kernel	1000	2
Aggregation	788	7

5.2. Performance metric

In this research, we evaluate the accuracy of the dynamic DBSCAN algorithm over the datasets in Table 1 using Purity criterion, as shown in (5). The purity metric evaluates the quality of formed clusters.

$$Purity = \frac{1}{n} \sum_{q=1}^k \max_{1 \leq j \leq l} n_q^j \quad (5)$$

Where n is the total number of samples; l is the number of categories, n_q^j is the number of samples in cluster q that belongs to the original class j ($1 \leq j \leq l$). A cluster quality is high if the purity close to 100%.

5.3. Result analysis

The result of this study is obtained to examine the performance of the proposed algorithm and other well-known algorithms such as DBSCAN [7], BDEDBSCAN [24], and DBSCAN-KNN-GA [25] over these datasets. We apply the Dynamic DBSCAN algorithm on the different shape datasets in order to visualize the shapes of the clusters. It is important to check if the proposed algorithm can produce clusters with arbitrary shapes since it is one of the primary requirements of density-based clustering algorithms. Figure 6 shows the cluster for the Corner dataset where the proposed algorithm selects $Eps = 2.5$ and $MinPts = 12$ dynamically.

After that, the performance of the proposed dynamic DBSCAN algorithm is evaluated over the three spirals dataset. For these datasets, the proposed algorithm selects $Eps = 1.5$ and $MinPts = 2$ dynamically and creates clusters, as shown in Figure 7. The formation of the clusters by the proposed algorithm for the rest of the datasets is shown in Figure 8. Experimental results of four different DBSCAN algorithms shown in Table 2.

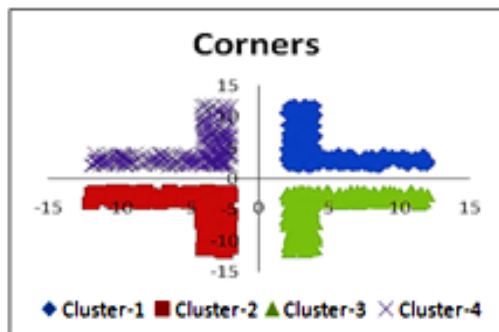


Figure 6. Corners dataset clustering result

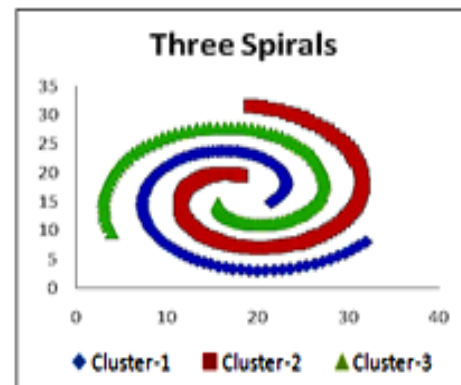


Figure 7. Three spirals dataset clustering result

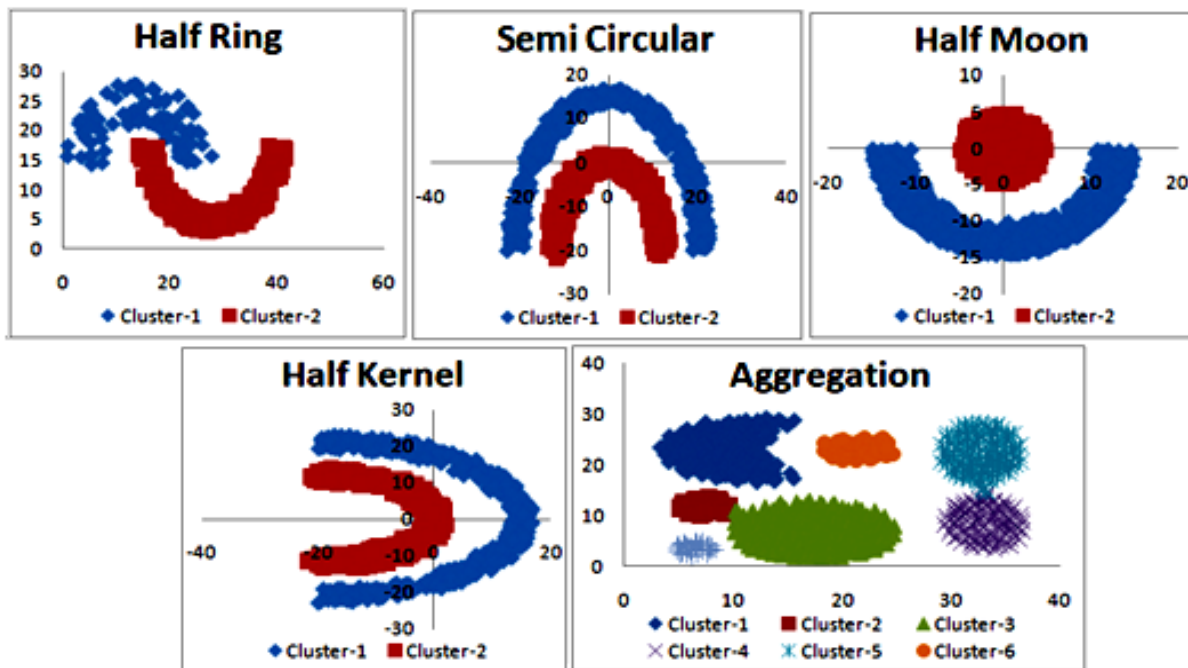


Figure 8. The results of performing clustering using the proposed DBSCAN algorithm on half ring, semi circular, half moon, half kernel, and aggregation datasets

Table 2. Experimental results of four different DBSCAN algorithms

Data Set	DDBSCAN Purity (%)	DBSCAN Purity (%)	BDEDBSCAN Purity (%)	DBSCAN-KNN-GA Purity (%)
Corners	100	91.33	100	100
Three Spirals	100	92.14	99.52	99.33
Half Ring	99.97	90.26	99.21	99.37
Semi Circular	100	90.04	99.10	99.61
Half Moon	99.98	91.46	99.75	99.75
Half Kernel	99.5	91.12	99.01	99.31
Aggregation	92.61	71.42	92.51	81.42

6. CONCLUSION

In this research, we are motivated to eliminate the drawbacks of the original DBSCAN algorithm. The selection of *Eps* and *MinPts* in the DBSCAN algorithm is a challenging task in the real world. If the DBSCAN has chosen a large value for *Eps* then DBSCAN forms the clusters having dissimilar data. On the other hand, if a small value is chosen for *Eps*, then this technique forms the clusters having a small amount of similar data. In this research, we address the initial value selection problem of the DBSCAN algorithm and propose a dynamic DBSCAN algorithm to remove initial parameter sensitivity. The proposed technique calculates the best *Eps* and *MinPts* dynamically, which improves the cluster quality. The experimental results evaluated the performance of the dynamic DBSCAN algorithm for various 2D data sets and compared it with the DBSCAN, BDEDBSCAN, and DBSCAN-KNN-GA algorithms. The result shows the better performance of the proposed Dynamic DBSCAN algorithm over the other existing algorithms on seven various datasets. Future research on DDBSCAN should consider the following issues. In the DDBSCAN algorithm, we use pair-wise Euclidian distance, so when the data set is large, its performance is reduced. If the inter-cluster distance is low, the DDBSCAN algorithm performance will be decreased.

REFERENCES

- [1] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: Parameter reduction and outlier detection," *Information Systems*, vol. 38, no. 3, pp. 317-330, 2013, doi: 10.1016/j.is.2012.09.001.
- [2] S. Sharma, J. Agrawal, S. Agarwal, and S. Sharma, "Machine learning techniques for data mining: A survey," in *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1-6, Dec. 2013, doi: 10.1109/ICCIC.2013.6724149.
- [3] S. Agarwal, "Data mining: Data mining concepts and techniques," in *2013 International Conference on Machine Intelligence and Research Advancement*, Dec. 2013, pp. 203-207, doi: 10.1109/ICMIRA.2013.45.
- [4] Q. Liu, M. Deng, Y. Shi, and J. Wang, "A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity," *Computers & Geosciences*, vol. 46, pp. 296-309, 2012, doi: 10.1016/j.cageo.2011.12.017.
- [5] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *ACM Sigmod Record*, vol. 25, no. 2, pp. 103-114, 1996, doi: 10.1145/235968.233324.
- [6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," ser. *KDD '96*, AAAI Press, 1996, pp. 226-231.
- [7] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*. Springer, 2006, pp. 25-71, doi: 10.1007/3-540-28349-8_2.
- [8] C. C. Liu, S. W. Chu, Y. K. Chan, and S. S. Yu, "A modified K-means algorithm - two-layer K-means algorithm," in *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2014, pp. 447-450.
- [9] M. Hossain, M. Akhtar, R. Ahmad, and M. Rahman, "A dynamic k-means clustering for data mining," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 13, no. 2, pp. 521-526, Feb. 2019, doi: 10.11591/ijeecs.v13.i2.pp521-526.
- [10] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web*, ser. *WWW '10*. New York, NY, USA: Association for Computing Machinery, 2010, pp. 1177-1178, doi: 10.1145/1772690.1772862.
- [11] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2481-2499, 2012, doi: 10.1109/TIP.2011.2176743.
- [12] V. W. Ajin and L. D. Kumar, "Big data and clustering algorithms," in *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, 2016, pp. 1-5.
- [13] A. Saxena et al., "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664-681, 2017, doi: 10.1016/j.neucom.2017.06.053.
- [14] M. Parimala, D. Lopez, and N. Senthilkumar, "A survey on density based clustering algorithms for mining large spatial databases," *International Journal of Advanced Science and Technology*, vol. 31, no. 1, pp. 59-66, 2011.

- [15] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gbscan and its applications," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 169-194, 1998, doi: 10.1023/A:1009745219419.
- [16] T. Cheng, "An improved DBSCAN clustering algorithm for multi-density datasets," in *Proceedings of the 2nd International Conference on Intelligent Information Processing*, 2017, pp. 1-5.
- [17] A. Smiti and Z. Eloudi, "Soft dbscan: Improving dbscan clustering method using fuzzy set theory," in *2013 6th International Conference on Human System Interactions (HSI)*, Aug. 2013, pp. 380-385, doi: 10.1109/HSI.2013.6577851.
- [18] Nidhi and K. A. Patel, "An efficient and scalable density-based clustering algorithm for normalize data," *Procedia Computer Science*, vol. 92, pp. 136-141, 2016, 2nd International Conference on Intelligent Computing, Communication Convergence, ICCCC 2016, 24-25 January 2016, Bhubaneswar, Odisha, India, doi: 10.1016/j.procs.2016.07.336.
- [19] K. Mahesh Kumar and A. Rama Mohan Reddy, "A fast dbscan clustering algorithm by accelerating neighbor searching using groups method," *Pattern Recognition*, vol. 58, pp. 39-48, 2016, doi: 10.1016/j.patcog.2016.03.008.
- [20] A. Merk, P. Cal, and M. Woźniak, "Distributed DBSCAN Algorithm—Concept and Experimental Evaluation," in *International Conference on Computer Recognition Systems*, 2017, pp. 472-480: Springer, doi: 10.1007/978-3-319-59162-9_49.
- [21] L. Meng'Ao, M. Dongxue, G. Songyuan, and L. Shufen, "Research and improvement of DBSCAN cluster algorithm," in *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, 2015, pp. 537-540: IEEE, doi: 10.1109/ITME.2015.100.
- [22] H. Darong and W. Peng, "Grid-based DBSCAN algorithm with referential parameters," *Physics Procedia*, vol. 24, pp. 1166-1170, 2012, doi: 10.1016/j.phpro.2012.02.174.
- [23] A. Smiti and Z. Elouedi, "Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques," in *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, 2012, pp. 573-578, doi: 10.1109/INES.2012.6249802.
- [24] A. Karami and R. Johansson, "Choosing DBSCAN parameters automatically using differential evolution," *International Journal of Computer Applications*, vol. 91, no. 7, pp. 1-11, 2014, doi: 10.5120/15890-5059.
- [25] B. Mu, M. Dai, and S. Yuan, "DBSCAN-KNN-GA: a multi density-level parameter-free clustering algorithm," in *IOP Conference Series: Materials Science and Engineering*, vol. 715, no. 1. IOP Publishing, 2020, pp. 012023.