Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Solving the Non-Deterministic Nature of the DBSCAN Algorithm

Samarjit Das[1], Atowar ul Islam[2], Ms Priyanka Sarma[3], Ms Sangeeta Borkakoty[4], Angshuman Sinha[5]
[1] Department of CSE & IT, The Assam Royal Global University, Guwahati, India

[2,3,4] Department of Computer Science and Electronics, University of Science and Technology,Meghalaya, Techno City, Kiling Road, Baridua, 9th Mile,
Ri-Bhoi, Meghalaya-793101, India

[5] Department of CS & IT, Cotton University, Guwahati, India.

Corresponding Author: Email- atowar91626@gmail.com[2]

Abstract:

Clustering Algorithms are important tools in data mining and an effective approach towards the formation of clusters from a huge dataset. One such algorithm is the DBSCAN algorithm which follows the density based notion of clusters to determine clusters as well as noise points in the dataset. However, this algorithm has certain disadvantages. In our previous work we addressed one such major disadvantage - the absence of a valid formulation for the input parameters on which DBSCAN mainly relies. We introduced a simple algorithm to analyse the values of these parameters, following the basic ideas of simple frequency distributions, mean deviations and first nearest distances. DBSCAN also suffers from a problem of non-determinism, which arises, when adjacent clusters share a common border point. Although, it's a rare situation and doesn't create an impact, it may disrupt the cluster quality, allowing the chances of noise points to be present near to the clusters. Here, we present a simple algorithm that addresses this demerit. We follow a simple idea of observing the behaviour of the adjacent clusters under consideration in absence of the common border point and determine which cluster has the greatest affinity towards it.

Keywords: DBSCAN, common border point, critical core point, critical border point.

1. INTRODUCTION

Data Mining techniques are being extensively used for extracting out unusual, implicit, potentially useful insights about the data, not discovered previously [1,2,3,4]. Clustering is one such technique. Hard clustering is based on crisp sets where a data strictly belongs to a particular cluster. With the advent of the concept of fuzzy sets [5] a new approach of clustering has been unlocked known as fuzzy clustering. In this approach data partially belonging to multiple clusters are dealt with. Fuzzy C-Means (FCM) clustering algorithm is one of the most popular algorithms following this approach[6]. The performance of FCM with three different distance functions had been tested and compared in Pattern Recognition [7]. An application of FCM had been shown in vehicular pollution [8]. Das and Baruah had put forward different algorithms to deal with the problem of random initialization of centroids in FCM [9,10,11,12,13,14]. The application of YFCM algorithm of Das and Baruah is reflected in finding patterns of crime against women [15].DBSCAN (Density Based Spatial Clustering For Application of Noise) is a clustering algorithm that follows a density based notion of clusters [16]. It is designed to discover arbitrary-shaped clusters in any Database, and at the same time can distinguish noise points. The main reason why DBSCAN recognizes the clusters is that within each cluster there is a typical density of points which is considerably higher than outside of the cluster. Furthermore, the density within the areas of noise is lower than the density in any of the clusters [16]. Numerous researches have been done in the field of DBSCAN [17,18,19,20,21]. One such is the ST-DBSCAN, that deals with spatial-temporal data[17]. The other research has been done on determining the epsilon value on peatland on DBSCAN Algorithm to clustering data on

peatland hotspots in Sumatera [18]. Another research has been done on obtaining best results for varied densities in DBSCAN [19].While DBSCAN is known for its efficient retrieval of clusters, it does have some major disadvantages. In our previous work we addressed one such major disadvantage - the absence of a valid formulation for the input parameters on which DBSCAN mainly relies. We introduced a simple algorithm to analyse the values of these parameters, following the basic ideas of simple frequency distributions, mean deviations and first nearest distances [22]. Another disadvantage which may be rare in case but can affect the clustering scenario is the non-deterministic nature of DBSCAN. This situation may arise if two or more than two adjacent clusters share a common border point. The original paper [16] follows the order of processing, i.e., the cluster to visit the point at first includes it. Although, this may seem to solve the problem but there is a possibility of disrupting the cluster quality, if the point is not assigned to the correct cluster. Through our proposed algorithm, we have attempted to solve this disadvantage. We follow a simple idea of observing the behavior of the adjacent clusters under consideration in absence of the common border point and determine which cluster has the greatest affinity towards it. The rest of the paper is organized as follows. Section-2 includes the basic concepts of DBSCAN algorithm. Section-3 discusses the problems existing with DBSCAN, focusing on the non-deterministic nature of DBSCAN. Section-4 discusses the general idea, our proposed algorithm and inferences. Section-5 includes Results and Analysis of our proposed algorithm. Finally, the conclusions are presented in Section-6.

2.**The Dbscan Algorithm**

2.1 **Basic Concepts**

Given a dataset of n objects/points(O_1,O_2,…,O_n) we at first discuss the following concepts as proposed in [16] **.**

Definition 1: Eps(Ɛ) neighbourhood: For any arbitrary object Oj ϵ D, the Ɛ- neighbourhood is the set of all the objects within its Ɛ radius. Here the distance measure used may be Manhattan Distance , Euclidean Distance Measure etc.. Mathematically, it's defined as:- $N_ε$ (Oj)={O ϵ D| distance(Oj, O) <= Ɛ}

Definition 2: Core Object: An Object Oj ϵ D is said to be a core object if its Ɛ neighbourhood contains points equal to or more than the number of points specified in the MinPts parameter. i.e. |$N_ε$ (Oj)|>=MinPts

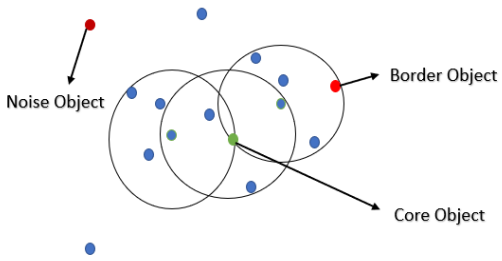

Fig 1: Core object, Border Object and Noise Object.

Definition 3: Directly Density Reachable: An object p is directly density reachable from an object q if the two conditions are satisfied:-

- p ϵ $N_ε$ (q); p, q ϵ D
- |$N_ε$ (q)|>=MinPts, i.e., q is a core object

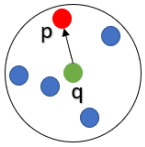

Fig 2: Directly Density Reachability

Definition 4: Density Reachable: An object p is density reachable from an object q if there exists a chain of objects O1, O2, O3, ……..,On, q=O1 and p=On such that Oi+1 is directly density reachable from Oi.

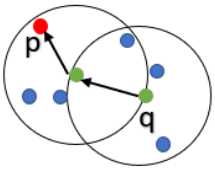

Fig 3: Density Reachability.

Definition 6: Density Connected: An object p is density connected to an object q if both p and q are density reachable from an object say o.

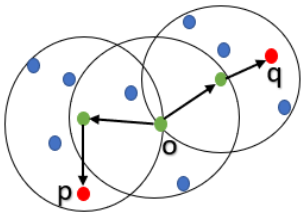

Fig 4: Density Connectivity.

Definition 7: Border Object**:** An object p is a border object if it is not a core object but density-reachable from another core object(see Fig 1).

Following the above concepts we shall establish the notion of clusters and noise objects.

Definition 8: Cluster: For the dataset D, a cluster C, wrt. Ɛ and MinPts, is a non-empty subset of D which, follows the following conditions:-

a. For all objects p,q: if p ϵ C and q is density reachable from p, then, q ϵ C. (Maximality).

b. For all p,q ϵ C: p is density connected to q wrt. Ɛ and MinPts. (Connectivity).

Definition 9: Noise: Let C1, C2, C3….Ck be the clusters of D wrt. Ɛ and MinPts. Then noise={O ϵ D| O ∉ Ci (i=1,2,..k)}(see Fig 1).

2.2. The Algorithm:

The Algorithm starts with a point p and retrieves all neighbours of point p within Eps(Ɛ) value. If the total number of these neighbours is greater than MinPts then, p is a core object thus, a new cluster is created. The point p and its neighbours are assigned into this new cluster. Then, it iteratively collects the neighbours within Eps(Ɛ) distance from the core points. The process is repeated until all of the points have been processed.

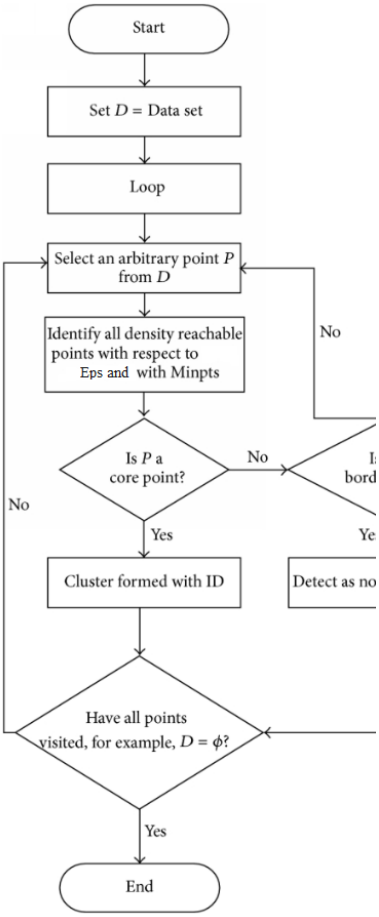

Fig 5:Flowchart of the DBSCAN Algorithm.

3. Problems Existing with DBSCAN

3.1 No valid formulation for input parameters

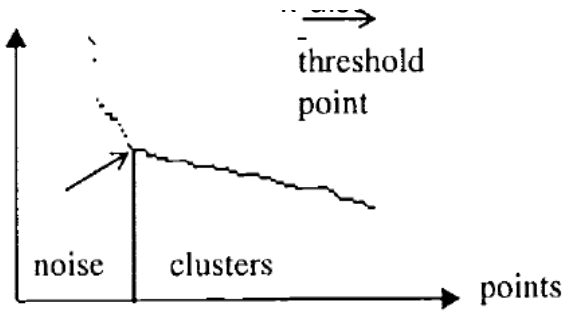

Fig 6: k-distance plot .

In the KDD proceedings, for determination of the input parameters, a heuristic approach has been provided [16]**.**The approach requires the determination of the k-distance plot for a given value of k and the first point in the first valley of the graph, referred to as the threshold point(see Fig 6), corresponds to the value of Ɛ (Eps) and the value of MinPts is set to the value of k.

Although this approach is an effective one, but there are a few problems associated with it.

First of all, there is no predefined formulation for the value of k. According to KDD, k=4 is proposed as the value for k>4 doesn't change the k-dist plot graph significantly. Therefore, the parameter MinPts is set to 4 for all databases (for 2-dimensional data). But, it totally depends on the data distribution or the density of the data. Moreover, this approach might not be true in case of high-dimensional data(for dimensions greater than 2-dimension).Furthermore, determination of the first valley in the k-dist plot graph might be difficult to be computed.

To capture the degree of density of a cluster, the density factor can be computed [17]. Based on this, density factor we can determine if the cluster is "loose" or a "tight" one.

3.2 Problem of identifying noise objects

Noise is a set of objects that doesn't belong to any of the clusters formed. Due to no valid formulation of the input parameters Ɛ (Eps) and MinPts and varying densities in the dataset, one of the major disadvantages of the DBSCAN algorithm is the problem of identifying noise objects [17].

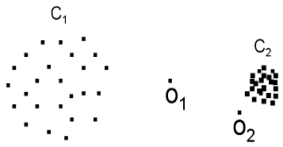

Fig 7: Example dataset which contain clusters of varying densities

Consider the above figure Fig 7, C1 and C2 are two clusters with varying densities. The problem here lies that what value should be used for the input parameters .C2 forms a denser cluster than C1. If the Eps value is less than the distance between o2 and C2, some objects in C1 are assigned as noise object. If the Eps value is greater than the distance between o2 and C2, the object o2 is not assigned as noise object.

3.3 Non-deterministic nature of DBSCAN

Although, it's a rare situation, it can disrupt the cluster quality, allowing the chances of noise points to be present near the clusters. We describe the problem using an illustration.

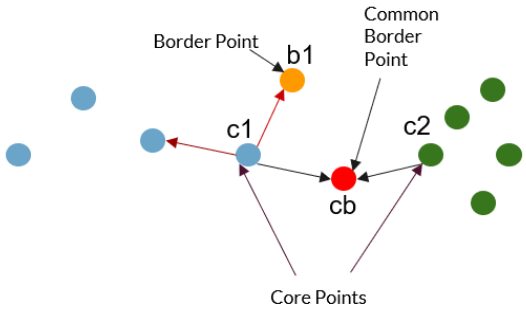

Fig 8: Adjacent clusters sharing a common border point

Consider the above figure. We have 2 clusters sharing a common border point cb, c1 and c2 being the core points from which cb is directly density reachable. Again, b1 is a border point which is directly density reachable from c1. Now, according to the original DBSCAN [16], the order of processing is considered i.e. , the first cluster to visit the common border point will include it. Suppose that, the cluster to which c2 belongs visits cb at first and includes it. We observe the clustering in that case.

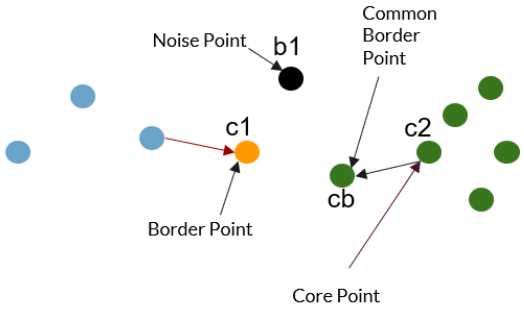

Fig 9: Common border point assigned to a cluster

Here, we can observe that c1 has converted to a border point along with b1 converting to a noise point. So, we have obtained a clustering scenario where due to a faulty processing of the common border point, we observe a noise point way too close to the cluster.

Through our proposed algorithm we have attempted to solve this issue(see section-4).

4. Our Present Work:

4.1 General Idea

The general idea is as such:

- We consider such core points of adjacent clusters from which the common border point is directly density reachable.
- We observe the behaviour of such points in absence of the common border point from their eps neighbourhoods.
- If a core point remains a core point after the removal, it implies that the absence of the common border point will not impact its cluster quality.
- If a core point converts to a border point after the removal, we find such core points and determine if there is any presence of noise points around its eps neighbourhood.
- The cluster with maximum no of noise points around it has the greatest affinity towards the common border point.

We consider the figure(Fig 8). We observe the clustering in absence of cb.

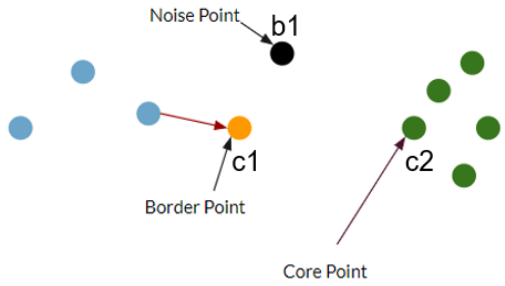

Fig 10: Clustering in absence of common border point

Here, we observe that the cluster to which c2 belongs doesn't have any impact. Whereas, the cluster to which c1 belongs has an impact in the absence of cb with c1 converting to a border point and b1 converting to a noise point. As we get a noise point nearer to the cluster to which c1 belongs, it has a greater affinity towards cb and hence must be included in it.

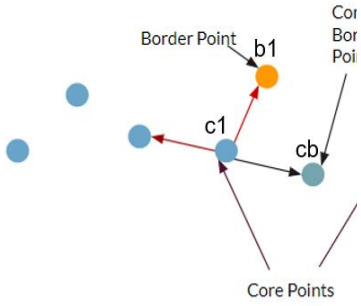

Fig 11: Common border point assigned to cluster with greater affinity

Thus, we observe that compared to the previous clustering result (Fig 9), we have obtained a better clustering result in the above figure(Fig 11).

4.2 Proposed Algorithm

We consider the clustering scenario where the common border point belongs to the adjacent clusters. Given this, we describe our proposed algorithm as such:

Input: K Adjacent Clusters with a common border point cb

Algorithm:

- Start.
- Determine K Adjacent Clusters with a common border point cb
- Given the common border point cb, determine its Ɛ-neighbourhood .
- ***do for each*** *core point x in Ɛ-neighbourhood of cb:*
  - Observe the Ɛ-neighbourhood in absence of cb
  - ***if*** *x remains a core point:*
    - No action is taken
  - ***else:***
    - ***do for each*** *point d in Ɛ-neighbourhood of x:*
- ***if*** *d converts to a noise point:*
  - count such d points as count_noise.

- ***else:***

  - No action is taken

- ***if*** *all x points remain as core points:*

  - Assign cb to any of the cluster

- ***else:***

  - Consider x for which count_noise is maximum. Assign cb to it.

- Stop.

4.3 Inferences

4.3.1.1 Inferences on Ɛ-neighbourhood of the border point

- The Ɛ-neighbourhood of the border point contains only those core points from which it is directly density reachable.

- The eps neighbourhood of the border point does not satisfy the MinPts condition. Hence, a relation can be obtained as such: MinPts>=(Ɛ-neighbourhood of border point)+1

4.3.1.2 Inferences on assignment of the common border point

Let the common border point be cb.

- If the core points remain as core points even after the removal of cb, the border point is assigned to any one of the cluster.

- If the core points convert to border points, after the removal, then, we assign cb to that cluster for which we get the maximum no of noise points in its absence.

- If for two or more clusters the no. of noise points obtained in the absence of cb is maximum and equal then we assign the border point to any one of the clusters.

- The worst case that can be possible is that for a cluster, if the core point from which, the common border point is directly density reachable, is the only core point in the cluster and it fails to satisfy the core point condition in the absence of the border point in its eps neighbourhood then, we get all of its points as noise points, if the border point is not assigned.

5 Result and Analysis

We run our proposed algorithm using a simple 2D dataset in Python.

5.1.1.1 The Dataset

```
[[2 0]
 [2 1]
 [2 2]
 [3 1]
 [3 2]
 [4 1]
 [5 0]
 [5 1]
 [5 2]
 [6 1]]
```

Fig 12: 2D Dataset Sample

Given above in the figure(Fig 12) is the sample dataset that we have considered.

5.1.1.2 The Adjacent clusters with the common border point

We have considered the following adjacent clusters with the common border point for the input parameters (Ɛ=1, MinPts=4).

```
[[2 0]
 [2 1]
 [2 2]
 [3 1]
 [3 2]
 [4 1]]
```

```
[[4 1]
 [5 0]
 [5 1]
 [5 2]
 [6 1]]
```

Fig 13: Two Adjacent Clusters

Given the adjacent clusters, we have determined the common border point.

```
The common value is [4 1]
```

Fig 14: The common border point

5.1.1.3 The Distance Matrix

```
[[0.         1.         2.         1.41421356 2.23606798 2.23606798
  3.         3.16227766 3.60555128 4.12310563]
 [1.         0.         1.         1.         1.41421356 2.
  3.16227766 3.         3.16227766 4.        ]
 [2.         1.         0.         1.41421356 1.         2.23606798
  3.60555128 3.16227766 3.         4.12310563]
 [1.41421356 1.         1.41421356 0.         1.         1.
  2.23606798 2.         2.23606798 3.        ]
 [2.23606798 1.41421356 1.         1.         0.         1.41421356
  2.82842712 2.23606798 2.         3.16227766]
 [2.23606798 2.         2.23606798 1.         1.41421356 0.
  1.41421356 1.         1.41421356 2.        ]
 [3.         3.16227766 3.60555128 2.23606798 2.82842712 1.41421356
  0.         1.         2.         1.41421356]
 [3.16227766 3.         3.16227766 2.         2.23606798 1.
  1.         0.         1.         1.        ]
 [3.60555128 3.16227766 3.         2.23606798 2.         1.41421356
  2.         1.         0.         1.41421356]
 [4.12310563 4.         4.12310563 3.         3.16227766 2.
  1.41421356 1.         1.41421356 0.        ]]
```

Fig 15: The Euclidean Distance Matrix

We have considered the Euclidean Distance Measure and obtained the Distance Matrix.

4.Ɛ-neighbourhood

```
[[3 1]
 [4 1]
 [5 1]]
```

Fig 16: Ɛ-neighbourhood of the common border point

Here we observe that the common border point [4,1] has 2 core points [3,1] and [5,1]. We then observe the Ɛ-neighbourhood of [3,1] and[5,1] in absence of [4,1].

```
Eps neighbourhood of [3,1] in absence of [4,1] is [[2,1],[3,1],[
[3,1] converts to a border point as MinPts=4
[2,1] is a core point: No action is taken
[3,2] converts to a noise point
```

Fig 17: [3,1] in absence of [4,1] in its eps-neighbourhood

From the above figure(Fig 17), we observe that [3,1] converts to a border point along with [3,2] in its Ɛ-neighbourhood converting to a noise point.

```
Eps neighbourhood of [5,1] in absence of [4,1] is [[5,0],[5,1],[5,2],[
[5,1] remains a core point:No action is taken
```

Fig 18:[5,1] in absence of [4,1] in its Ɛ-neighbourhood

From the above figure(Fig 18),
we observe that[[5,1] remains a core point. Hence, no action is taken.

From the above observations we determine that the cluster to which [3,1] belongs has a greater affinity towards [4,1] and hence must be included in it.

5.1.1.4 Observations

- Clustering in absence of common border point [4,1]

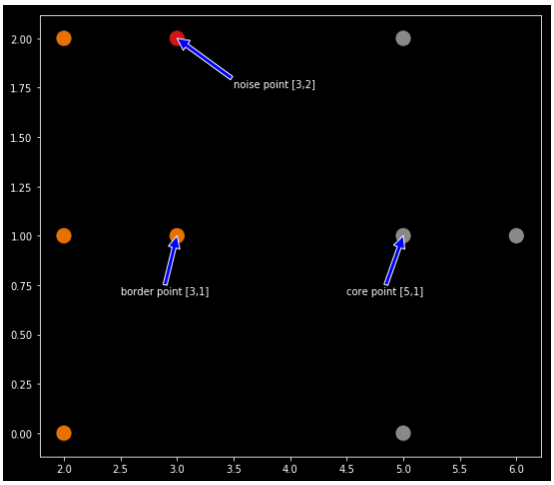

Fig 19: Clustering in absence of common border point [4,1]

We observe that [3,2] converts to a noise point and [3,1] converts to a border point whereas, [5,1] remains a core point in absence of [4,1].

- Final Clusters Obtained

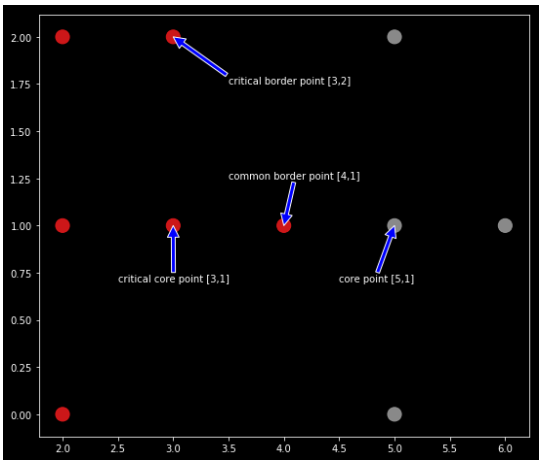

Fig 20: Clusters obtained in presence of common border point [4,1]

The common border point [4,1] is assigned to the cluster to which the core point[3,1] as it has a greater affinity towards the point [4,1]. We have annotated the core point[3,1] as 'critical core point [3,1]' in the above figure(Fig 20) citing that it converts to a border point in absence of the common border point [4,1](Fig 19). We have also annotated the border point[3,2] as 'critical border point' in the above figure(Fig 20) citing that it converts to a noise point in absence of the common border point [4,1](Fig 19).

Thus, we have successfully run our algorithm and obtained the observations to verify our inferences made(see Section 4c).

6 Conclusions:

DBSCAN is an efficient algorithm when it comes to retrieval of clusters and noise points but, it does have some disadvantages. DBSCAN is not efficient in terms of identifying noise objects when the data distribution is not uniform. Secondly, since, it doesn't have any predefined formulation to get the input parameters, it some times becomes cumbersome and we end up getting varying results for the same dataset. Another disadvantage is the non-determinism of DBSCAN in case of a border point belonging to two or more adjacent clusters. The original DBSCAN [16] prefers the order of processing i.e. the cluster to visit the point at first includes it. But from our illustrations (Fig 9), we have observed that following this method might disrupt the clustering resulting in a noise point near the cluster. Our present work demonstrates a simple algorithm to attempt to solve the problem of non-determinism of DBSCAN on determining the cluster to which the common border point must belong to. Through our work, we have presented a simple solution, following the simple idea of observing the behavior of the adjacent clusters under consideration in absence of the common border point and determine which cluster has the greatest affinity towards it. Using the general idea(see Section-4.a), we present our algorithm (see Section-4.b) to analyse such clustering scenarios. In order to demonstrate the applicability of our algorithm, we have created a simple program using Python. After observing the results obtained and the analyses made(see Section-5), it can be concluded that our proposed algorithm appears to be promising in terms of attempting to solve the limitation of DBSCAN regarding its non-determinism in case of a border point belonging to more than one adjacent clusters and has successfully verified all the inferences made (see Section-4.c).

References

1. Han J, Kamber M, Pei J. 2012. Data Mining: Concepts and Techniques. 3rd ed. San Francisco (US): Morgan-Kauffman..
2. Pang- Ning Tan, Michael Steinbach, and Vipin Kumar, "Introduction to Data Mining," Pearson Addison Wesley, 2006.
3. A. Jam and R. Dubes, " Algorithm for Clustering data" Englewood Cliffs, NJ:Prentice-Hall 1988.
4. Arun K Pujari, "Data Mining Techniques", 4th ed., Universities press, 2017.
5. L. A., Zadeh "Fuzzy Sets", Information and Control, Vol. 8, Issue. 3, pp.338-353, 1965.
6. J. C. Bezdek, , "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.
7. Das, S. (2013), Pattern Recognition using the Fuzzy c-means Technique, *International Journal of Energy, Information and Communications*, vol. 4, Issue 1, 1-14.
8. Das, S. and H. K. Baruah (2013), Application of Fuzzy C-Means Clustering Technique in Vehicular Pollution, *Journal of Process Management – New Technologies,* Vol. 1, Issue 3, 96-107.
9. Das, S. and H. K. Baruah (2013), A Comparison of Two Fuzzy Clustering Techniques, *Journal of Process Management – New Technologies,* Vol. 1, Issue 4, 1-15.
10. Das, S. and H. K. Baruah (2014), Dependence of Two Different Fuzzy Clustering Techniques on Random Initialization and a Comparison, *International Journal of Advanced Research in Computer Science and Software Engineering , Vol. 4, Issue 1*,422-428 .
11. Das, S. and H. K. Baruah (2014), An Approach to Remove the Effect of Random Initialization from Fuzzy C-Means Clustering Technique, *Journal of Process*

Management – New Technologies, Vol. 2, Issue 1, 23-30.

12. Das, S. and H. K. Baruah (2014), A New Method to Remove Dependence of Fuzzy C-Means Clustering Technique on Random Initialization, *International Journal of Research in Advent Technology, Vol. 2, No.1*, 322-330.
13. Das, S. and H. K. Baruah (2014), Towards Finding A New Kernelized Fuzzy C-Means Clustering Algorithm, *Journal of Process Management – New Technologies,* Vol. 2, Issue 2,54-65.
14. Das, S. and H. K. Baruah (2014e), A New Kernelized Fuzzy C-Means Clustering Algorithm with Enhanced Performance, *International Journal of Research in Advent Technology, Vol. 2, No.6*, 43-51.
15. Das, S., A. Das and A. U. Islam (2018), Finding Patterns in Crime Against Women Using a Fuzzy Clustering Technique, *International Journal of Computer Sciences and Engineering,* Vol. 6, Issue 8, 356-363.
16. M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density –based algorithm for discovering clusters in large spatial databases with noise in: Proceedings of Second international Conference on Knowledge Discovery and Data Mining, Portland, OR , 1996,pp.226-231.
17. D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial-temporal data. Data Knowl. Eng., 60(1):208{221, 2007.
18. Nadia Rahmah and Imas Sukaesih Sitanggang 2016 IOP Conf. Ser.: EarthEnviron. Sci. 31-01-2012.
19. Elbatta MNT. 2012. An improvement for DBSCAN algorithm for best results in varied densities [disertasi]. Gaza (PS): Islamic University of Gaza.
20. A Morena, M. Y Santos and S. Carnerno, Density-based clustering algorithms DBSCAN and SNN, July 2005.
21. G.H. Shah, C'K. Bhensdadia, A.P. Ganatra, " An Empirical Evaluation of Density Based Clustering Techniques", Vol 1, Issue 2, March 2012.
22. S. Das, A. U. Islam, P.S. Velumani & B. Sarma, "An Analysis of the Input Parameters of the DBSCAN Algorithm", Computer Integrated Manufacturing Systems, Vol. 28 No. 11, 1299-1314.