ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2022.0352

AUTOMATIC GENERATION OF PARAMETERS IN DENSITY-BASED SPATIAL CLUSTERING

Jayasree Ravi and Sushil Kulkarni
Department of Computer Science, University of Mumbai, India

Abstract

As a result of emerging new techniques for scientific way of collecting data, we are able to accumulate data in large scale pertaining to various fields. One such method of data mining is Cluster analysis. Of all clustering algorithms, density-based clustering is better in terms of clustering quality and the way the data are handled. Density based clustering is advantageous over other clustering algorithms in the following ways – arbitrary shaped clusters are formed; number of clusters need not be known and noise is handled. However, there are two main points that are critical in density-based clustering. Firstly, it is not effective while handling datasets of varied density. Secondly, the selection of input parameters ε and MinPts play a critical role in the quality of clustering. This paper proposes a model – Automatic Generation of Parameters in Density-Based Spatial Clustering (AGP-DBSCAN) that aims at improving the density-based clustering by generating different candidate parameters. With these candidates, we will be able to handle both uniform density and varied density datasets. The results of experiments also look promising for different clustering datasets.

Keywords:

Clustering Algorithms, Density-based Clustering, Density Parameters, Generation of Parameters

1. INTRODUCTION

Cluster Analysis is a process of finding groups among data where data can be a set of customer behavior, textual information or geo-tagged tweets. Clusters are formed w.r.t the similarity measure each data shares with other data. There have been many algorithms for forming clusters, each with the aim to resolve issues like multidimensional data, varied density data, data with noise objects, and so on. If one algorithm handles varied density [1], then the other algorithm handles spatial-temporal information [2] and the other handles textual similarity [3].

The popular clustering algorithms like K-Means clustering and Density-Based clustering are based on input parameters which are to be determined by the user but have a great influence on the clustering result. Different algorithms that are proposed in past may have followed a certain methodology to calculate the input parameters. The objective of this paper is to develop an efficient methodology that were adapted in previous algorithms and deliver an improved model of density-based clustering. The proposed algorithm will be able to discover clusters of arbitrary shapes and does not require input parameters.

The concept of clustering can be explained in the following way [4] - Cluster analysis is the process of grouping data objects based on the information present in the data set and the relationship observed in the dataset. The main motive is that the data points within a cluster should be of similar nature or inter-related to one another and they are different or not linked to the data objects belonging to other clusters using Euclidean Distance. Homogenous nature within the cluster and heterogeneous nature between different clusters decide the quality of clustering.

There are different types of clustering algorithms and their improvisations like hierarchical clustering [5], partition-based clustering [6]-[8], grid-based clustering [9], density-based clustering [10] and so on. Our study is based on density-based clustering and the model DBSCAN is explained in [10].

The objective of the proposed model is to minimize the user intervention in deciding the density parameters in density-based clustering. This paper proposes a mathematical approach for automatically generating ε and its corresponding *MinPts* – the two parameters of density-based clustering.

The rest of this paper is organized as follows: Section 2 covers the basic aspects of DBSCAN. Section 3 deals with review of various research paper related to DBSCAN, its variations and improvisations on handling varied data sets, handling high dimensional data. Section 4 proposes a new model Automatic Generation of Parameters in Density-Based Spatial Clustering (AGP-DBSCAN). Section 5 gives details of the datasets used in this study, implementation details of the proposed algorithm on the datasets and compares the result of proposed model with existing density-based clustering model. Section 6 concludes the work with future scope of study.

2. DENSITY-BASED CLUSTERING

The DBSCAN is a basic algorithm which follows density-based clustering. DBSCAN is a popular density-based clustering algorithm which has the following characteristics.

- Number of clusters need not be known in the beginning
- Clusters of any arbitrary shape can be formed
- It handles noisy data

It takes two parameters -ε>0 is the radius of the data point and *MinPts* is the minimum number of points that should be present within the ε neighborhood.

2.1 NOTATIONS

The notations that are used throughout the paper are listed in Table.1.

Table.1. Notations used

Notation	Description
D	Set of Data points
p, q	Data points $\in D$
$d(p,q)$	Euclidean distance between p and q
E	$E \subset D$
$y_1, y_2, \ldots y_n$	Data points $\in E$

ε	Radius from a point
MinPts	Minimum number of points within the ε radius
n	$\|D\|$ Number of data points
m	$\|E\|$ Number of data points in the subset of dataset

2.2 DEFINITIONS

Definition 1:

ε-neighbourhood: The neighbourhood of a data point p within a radius $\varepsilon > 0$ is referred to as the ε-neighbourhood of the point

Definition 2:

Core Point: If the $\boldsymbol{\varepsilon}$ -neighbourhood as defined above of a data point p has at least *MinPts* number of data points, a data point *p* qualifies to be a core point.

Definition 3:

Border Point: If a data point *q* belongs to the ε-neighborhood of a core point p but has a smaller number of points than *MinPts* within its own radius ε, then it is referred to as border point

Definition 4:

Directly Density-Reachable: A data point *q* is directly density-reachable from a data point *p*, if *q* falls within the ε-neighbourhood of *p*, and also p is a core point

Definition 5:

Density-Reachable: Let $E=\{y_1 \text{ to } y_n\}$, a subset of *D* between *p* and *q*, such that y_1 is directly density reachable to *p*, y_2 is directly density reachable to y_1, and so on, and y_n is directly density reachable to *q*, then *q* is density-reachable to *p* w.r.t ε and *MinPts*

Definition 6:

Density-Connected: A data point *p* is density-connected to a data point q with regards to ε and *MinPts*, if there is an object *y* in *D* such that both *p* and *q* are density-reachable from *y* w.r.t ε and *MinPts.*

Definition 7:

Cluster: Cluster *C* w.r.t $\boldsymbol{\varepsilon}$ and *MinPts* is a non-Empty subset of data set *D* satisfying the following conditions.

Reachability: $\forall p,q \in D$; if $q \in C$ and *p* is density-reachable from *q* w.r.t (ε, *MinPts*), then $p \in C$

Connectivity: $\forall p,q \in C$; *p* is density-connected to *q* w.r.t ($\boldsymbol{\varepsilon}$, *MinPts*).

Definition 8:

Noise: Let $C_1, \ldots, C_m$ be the clusters of the dataset D w.r.t (ε, *MinPts)*. Then we define the noise points as the set of data points in D not belonging to any cluster as in Eq.(1) i.e.,

$$noise\ points = \{p \in D | \forall i : p \notin C_i\} \quad (1)$$

This algorithm classifies every point either as a core point, border point or noise. Clusters are formed based on the density. There have been many improvisations on DBSCAN which are covered in the literature review.

3. LITERATURE REVIEW

There have been many improved models built on DBSCAN. All the models fall into any one of the following improvisation aspects (i) handling varied-density data, (ii) automatic generation of density parameters and (iii) improving the performance of DBSCAN in terms of run-time complexity. We have done a critical review on the papers which addresses the above-mentioned challenges of DBSCAN. There have also been extensive survey papers conducted on clustering algorithms and applications of DBSCAN in various fields. Our literature review covers these papers also.

3.1 VARIED-DENSITY DATASETS

VDBSCAN algorithm proposed in [1] addresses the issue of spatial heterogeneity in the data with the help of different parameters for detecting clusters in an area based on the density in that area. But this method is not appropriate for a large dataset such as social media event detection from geotagged tweets.

3.2 AUTOMATIC GENERATION OF DENSITY PARAMETERS

In the proposed method mentioned in [11], the authors have k variable which can be declared by using Cartesian technique and algorithmic average determination method. This will automatically select the input parameters which identify clusters of varied density. The proposed algorithm, similar to DBSCAN discovers clusters of arbitrary shape. This algorithm does not require any input parameters and adapts the terms and definitions of DBSCAN algorithm. The authors of [12] have proposed a mathematical way of calculating one of the input parameters of density-based clustering ε value. [13] presents incremental DBSCAN which can be adapted for a collection of data objects simultaneously, named MOiD (Multiple Objects incremental DBSCAN). As a first step, thus model runs DBSCAN to do the clustering of the incremental dataset. After cluster analysis, the proposed MOiD incorporates the clusters of incremental datasets to that of existing dataset. The model presented by authors in [14] suggest a way to automatically determine the $\boldsymbol{\varepsilon}$ parameter.

3.3 IMPROVISED DENSITY-BASED CLUSTERING MODELS

The hybrid clustering scheme proposed in [15] suggests to develop prototypes using leader clustering scheme. Along with the derived leaders, number of grouped patterns is also stored. In this literature, rough set theory is used for result analysis. This paper also analyses the criteria to be satisfied for this model to show similar working as DBSCAN. DBSCAN++[16] is the proposed algorithm in this paper which is a step towards an improved version of DBSCAN. DBSCAN++ is based on the notion that there need to be only a subset of data points for which the density estimates need to be computed. The authors have proposed two strategies to choose these points - uniform and greedy K-center-based sampling. This paper has proposed DBSCAN++ which runs in a small amount of time compared to DBSCAN, while giving optimum performance and uniformly producing good clustering options across varied hyper-parameter settings. For the detection of outliers, it delivers similar results of DBSCAN. In [17], the paper addresses the instability of DBSCAN for handling border points which are outside the formed clusters. This algorithm retains the key concepts of the DBSCAN algorithm, with an additional potential to improve the results of clustering by solving the issue of border objects.

3.4 SURVEY PAPERS ON CLUSTERING TECHNIQUES

The authors of [18] have conducted a survey on clustering techniques. In [19], the authors have analysed different variations of DBSCAN algorithms. The studies conducted in [20][21][22] also compare different models of DBSCAN algorithm. The authors of [23] have surveyed different clustering algorithms and have concluded that DBSCAN has the higher silhouette coefficient.

3.5 APPLICATIONS OF DBSCAN

The study conducted by the authors of [24] applies an improved version of DBSCAN on student evaluation system. DBSCAN algorithm is popular in text clustering also [25]. DBSCAN model is incorporated on social media data for various applications like text clustering in twitter data [26][27].

4. PROPOSED MODEL

The method proposed in this paper will have a set of *MinPts* values also for every corresponding ε. In short, every density region will have a unique pair of (ε, *MinPts*) values. Instead of leaving the decision of choosing the parameters of Density-Based Spatial Clustering to the users, we have incorporated a mathematical technique of calculating the parameters of Density-Based Spatial Clustering automatically based on the dataset. This improved method takes care of the varied density dataset which is a common feature in social media data. With this understanding of the objectives, we have extended the definitions of density-based clustering.

Definition 9:

Let *den* be the number of density regions identified in D. Then there exists one ε corresponding to every density as represented in Eq.(2) and there exists one *MinPts* corresponding to every ε as represented in Eq.(3).

$$\forall den, \exists \varepsilon, \text{ s.t. } \varepsilon > 0 \tag{2}$$

$$\forall \varepsilon, \exists MinPts, \text{ s.t. } MinPts > 0 \tag{3}$$

Definition – 10:

If a data point *a* belongs to a certain density region *i*, then there exists a certain ε_i and a corresponding $MinPts_i$ where i = 1 to *den*, which the data point *a* is associated with.

Lemma:

If a data point *a* belongs to a certain density region *i*, where *i*=1 to *den*, then that point cannot be a member of some other density region j, where $i \neq j$ as mentioned in Eq.(4):

$$\forall i,j \in den, \exists a \in D, a \in i \Delta j \mid i \neq j \tag{4}$$

This paper handles the challenge of identifying the density regions and their corresponding parameters. For identifying the density regions, we use *k*-dist graph. Based on the density regions, the corresponding ε value and *MinPts* value are calculated. The steps adapted in this method are based on the previous works proposed in [9] and [10].

For plotting the k-dist graph, the value of *k* is calculated mathematically based on the size of the dataset. For plotting the *k*-dist graph, we adapt kd-Tree Data structure for finding the *k*-nearest-neighbour of data points. Kd-Tree is more efficient in performing the task in large datasets [17] [18].

Our proposed work calculates the ε value and *MinPts* value in an iterative fashion. In the first iteration, the parameters are calculated for the whole collection of data points. In the second iteration, a subset of the dataset which was marked as noise in the previous iteration is considered and parameters are calculated for the revised dataset. In case of uniform density dataset, the knee in the k-dist graph is identified by measuring the changes in slopes at regular intervals. If there is a 10% change in slope, then that part of the graph is considered as a knee and the corresponding value is chosen as ε. If there are multiple ε values, then the average of the minimum and maximum of the ε values is chosen.

In case of varied density dataset, the same procedure is adapted as in uniform density datasets and multiple ε values are identified. But it is done in iterations. In every iteration, one ε value and its corresponding *MinPts* value are calculated. Clustering is done with these parameters. In the next iteration, new ε value and its corresponding *MinPts* value are calculated for a revised dataset containing only a subset of the original dataset which were labelled as noise in the previous iteration. The steps of finding the ε values are given in the next subsection.

4.1 MATHEMATICAL APPROACH TO CALCULATE DENSITY PARAMETERS

Step 1: Set *k* value as natural log of the total number of rows in the dataset as in Eq.(5)

$$k = ln(n) \tag{5}$$

Step 2: Using *k* value, find the k-nearest neighbours of every data point. The k^{th} neighbour distance is chosen as in Eq.(6)

$$a_{knn} = max\,(knn(a)) | a \in D \tag{6}$$

where *knn*(*a*) is the collection of *k*-nearest neighbours of a data point *a*.

Step 3: The k^{th} neighbours of all data points are sorted and placed in *k*-list as shown in Eq.(7).

$$k\_list = \{a_{knn}\} |\ a \in D \tag{7}$$

Step 4: The i^{th} member and j^{th} member of the *k*-list are compared, where *i-j* = 0.1 * size of the dataset. If the difference is more than 10%, then i^{th} member is qualified to be a candidate for ε list. This is shown in Eq.(8)

$$\forall x,y \in k\_list,\ \varepsilon\_list = \{x\} | x\text{-}y > 0.1 * x \tag{8}$$

where *x* and *y* are i^{th} and j^{th} members of *k_list* respectively and $diff(i\ j) = 0.1*n$

Step 5: The ε for the current iteration is calculated by finding the average of maximum and minimum of ε_list as in Eq.(9)

$$\varepsilon = \sigma(max(\varepsilon\_list) \text{-} min(\varepsilon\_list)) \tag{9}$$

Eq.(5)- Eq.(9) are formulated to find the ε value of the current iteration. The ε value thus calculated is used for calculating the other parameter of density-based clustering, which is *MinPts.* This paper adapts the methodology mentioned in [9] for calculating the *MinPts* value. For every tweet, find the number of tweets available in the ε Neighbourhood. The average of all the *MinPts* gives the *MinPts* value for the corresponding ε.

$\forall a \in D$, *MinPts* (*a*) is given in Eq.(10)

$$\frac{1}{n}\sum_{i=1}^{n} M_i \quad (10)$$

In Eq.(10), n is the size of the dataset, M_i is the number of data points present in the ε neighbourhood of a data point i. The parameters that are calculated as above are then used in clustering the dataset based on the Euclidean proximity to each other. After calculating ε value and *MinPts*, DBSCAN algorithm is adapted for discovering geo-clusters in the data set.

4.2 COMPLEXITY ANALYSIS

The time complexity of finding k-nearest neighbour is O($n$log$k$) where n is the size of the dataset and k value is the natural log of the n. Then, the time complexity in the proposed method to calculate the parameters is $O(n^2\log k)$. The proposed method needs to store the distance of every pair of data points. So, the space complexity of the algorithm is $O(n^2)$. Along with this, the k-nearest neighbours of every data point also need to be preserved for which the space complexity comes to $O(nk)$. But, the value of k is too small, the overall space complexity is $O(n^2)$.

5. PERFORMANCE EVALUATION

In this section, we compare the performance of AGP-DBSCAN with AGED and DBSCAN on six clustering datasets. Table.2 gives the details of datasets used in this paper.

Table.2. Datasets Used

Dataset	Size	Dimension	Density
Aggregation [30]	788	2	Uniform
S1 [31]	5000	2	
Iris [32]	150	4	
Wine [33]	178	13	
Compound [34]	399	2	Varied
Zoo [33]	101	17	

The Table.3 given below shows the $\boldsymbol{\varepsilon}$ Value and *MinPts* Value that was arrived for the datasets by using AGP-DBSCAN.

Table.3. Parameters generated using AGP-DBSCAN

Dataset	$\boldsymbol{\varepsilon}$	*MinPts*	Clusters
Aggregation [30]	1.18	7	9
S1 [31]	35405.38	123	15
Iris [32]	0.61	14	2
Wine [33]	25.24	7	5
Compound [34]	1.53, 2.46	12, 4	8
Zoo [33]	2.44, 2.2	12, 3	7

The Table.4 displays the $\boldsymbol{\varepsilon}$ Value and *MinPts* value set by the user for Uniform-Density Datasets to be used in DBSCAN algorithm

Table.4. User-Defined Parameters of DBSCAN

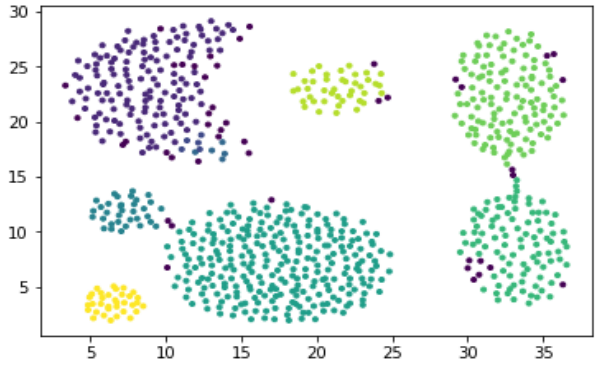
Dataset	$\boldsymbol{\varepsilon}$	*MinPts*	Clusters
Aggregation [30]	1.0	6	20
S1 [31]	32000	100	15
Iris [32]	0.5	15	2
Wine [33]	18	5	8
Compound [34]	0.92	5	5
Zoo [33]	0.8	8	1

In the case of AGED algorithm, there is user intervention in choosing the ε value from the generated values. If a different ε other than what authors have chosen is selected, then the performance may vary. To demonstrate that, we have chosen a different ε value from the list of ε for selected datasets shown in the study. The Table.5 shows the parameters chosen for clustering.

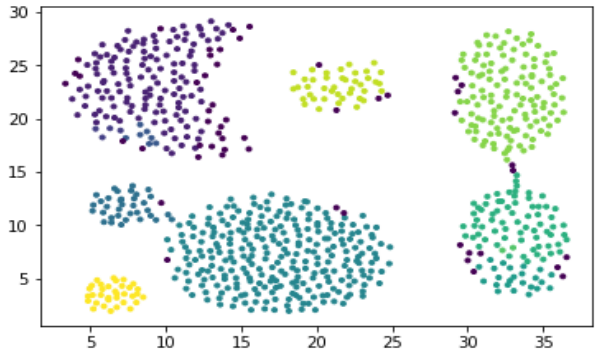
Table.5. $\boldsymbol{\varepsilon}$ chosen from the set of $\boldsymbol{\varepsilon}$ generated using AGED[12]

Dataset	$\boldsymbol{\varepsilon}$	*MinPts*	Clusters
Aggregation [30]	1.09	6	11
S1 [31]	26047	50	15
Iris [32]	0.42	6	4
Wine [33]	12.71	10	No clusters
Compound [34]	1.06, 2.67	4	6
Zoo [33]	1.03, 1.7	15	2

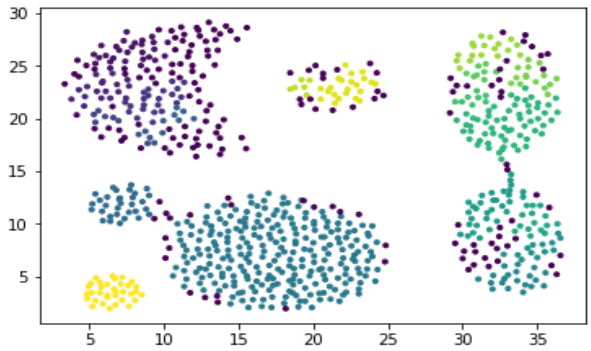
The clusters that were generated using AGP-DBSCAN, AGED, and DBSCAN for aggregation dataset (Uniform Density Dataset) are given in Fig.1.



(a) AGP-DBSCAN



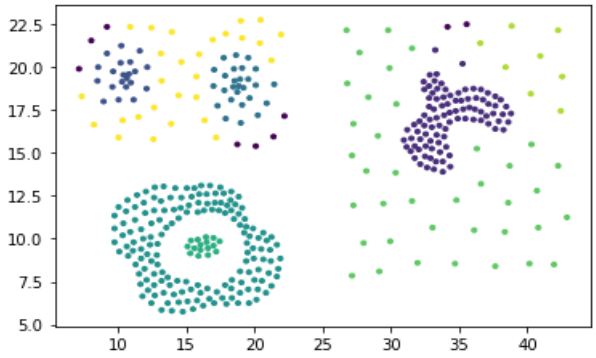
(b) AGED



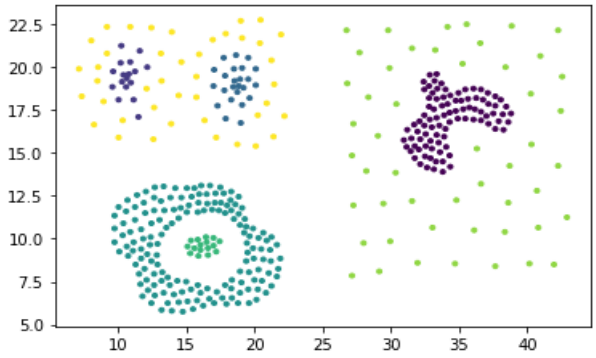
(c) DBSCAN

Fig.1. Clustering in Aggregation Dataset

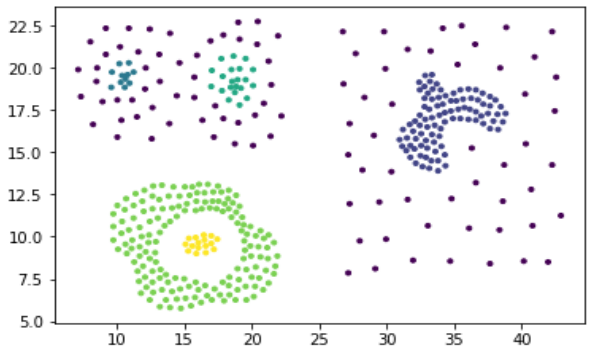
The clusters that were generated using AGP-DBSCAN, AGED, and DBSCAN for Compound dataset (Varied Density Dataset are given in Fig.2.



(a) AGP-DBSCAN



(b) AGED



(c) DBSCAN

Fig.2. Clustering in Compound Dataset

We compared the performance of our proposed model AGP-DBSCAN with the traditional DBSCAN Model with the help of Silhouette Coefficient. Silhouette Score Index [35] is a method of interpreting and validating the consistence within the clusters. The silhouette value is a way to measure the similarity of data to its own cluster compared to other clusters.

The Silhouette Coefficient is defined as in Eq.(11)

$$\text{Silhouette Coefficient} = \frac{a(i)-b(i)}{\max\{a(i),b(i)\}} \quad (11)$$

where

$a(i)$ is the average of the dissimilarity index of i^{th} object to all other objects in the same cluster and

$b(i)$ is the average of the dissimilarity index of i^{th} object with all objects in the closest cluster. The results are shown in Table.6.

Table.6. Silhouette Coefficients of AGP-DBSCAN, AGED and traditional DBSCAN method

Name of the Dataset	Silhouette Score		
	AGP-DBSCAN	AGED	DBSCAN
Aggregation[30]	0.379	0.279	0.147
S1[31]	0.583	0.520	0.551
Iris[32]	0.510	0.28	0.277
Wine[33]	0.163	0.00	0.055
Compound[34]	0.084	0.06	0.019
Zoo[33]	0.517	0.16	-0.030

From the Table.6, it is clearly evident that AGP-DBSCAN has out-performed DBSCAN and AGED in terms of clustering quality.

6. CONCLUSION AND FUTURE WORK

The proposed model AGP-DBSCAN implemented density-based clustering on datasets of varied nature and its performance was compared with the traditional density-based clustering algorithm and proved to be better. The proposed model has incorporated a novel way for finding ε values for different density regions and also has given a method of calculating *MinPts* for different ε values. This model needs only the dataset as input. ε and *MinPts* are calculated in the algorithm. There is very less human intervention in this proposed model which makes it less error prone.

Our suggestions on future work are to implement the proposed model on real data sets and streaming data such as Twitter where there are multiple dimensions to be considered-spatial similarity, Textual similarity and Temporal proximity. Also, k-dist plot is a time-consuming way of discovering varied densities in a dataset as it involves distance metric. Finding an alternate way to this problem is also another scope for future work.

REFERENCES

[1] L. Peng, Z. Dong and W. Naijun, “VDBSCAN: Varied Density Based Spatial Clustering of Applications with

Noise", *Proceedings of International Conference on Service Systems and Service Management*, pp. 1-4, 2007.

[2] D. Birant and A. Kut, "ST-DBSCAN: An Algorithm for Clustering Spatial-Temporal Data", *Data and Knowledge Engineering*, Vol. 60, No. 1, pp. 208-221, 2007.

[3] M.D. Nguyen and W.Y. Shin, "DBSTexC: Density-Based Spatio-Textual Clustering on Twitter", *Proceedings of ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 23-26, 2017.

[4] P.H. Tan, M. Steinbach and V. Kumar, "*Introduction to Data Mining*", Pearson Education, 2006.

[5] T. Zhang, R. Ramakrishnan and L. Miron, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", *Data Mining Knowledge Discovery*, Vol. 1, No. 2, pp. 141-182, 1997.

[6] Raymond T. Ng and Jiawei Han, "CLARANS - A Method for Clustering Objects for Spatial Data Mining", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 5, pp. 1003-1016, 2002.

[7] E.A. Pambudi, A.Y. Badharudin and A.P. Wicaksono, "Enhanced K-Means by Using Grey Wolf Optimizer for Brain MRI Segmentation", *ICTACT Journal on Soft Computing*, Vol. 11, No. 3, pp. 2353-2358, 2021.

[8] D. Murugan and S.S. Rathna, "Fuzzy based Privacy Preserved K-Means Clustering", *ICTACT Journal on Soft Computing*, Vol. 10, No. 1, pp. 2011-2014, 2019.

[9] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", *SIGMOD Record*, Vol. 27, No. 2, pp. 94-105, 1998.

[10] X.X. Martin Ester, Hans Peter Kriegel and Jiirg Sander, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 1-6, 1996.

[11] M. Naik Gaonkar and K. Sawant, "DBSCAN with Eps Automatic for Large Dataset", *International Journal on Advanced Computer Theory and Engineering*, Vol. 2, No. 2, pp. 2319-2526, 2013.

[12] N. Soni and A. Ganatra, "AGED (Automatic Generation of Eps for DBSCAN)", *International Journal of Computer Science and Information Security*, Vol. 14, No. 5, pp. 536-559, 2016.

[13] N. Soni and A. Ganatra, "MOiD (Multiple Objects incremental DBSCAN) - A Paradigm Shift in Incremental DBSCAN", *International Journal of Computer Science and Information Security*, Vol. 14, No. 4, pp. 316-346, 2016.

[14] F.O. Ozkok and M. Celik, "A New Approach to Determine Eps Parameter of DBSCAN Algorithm", *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 5, No. 4, pp. 247-251, 2017.

[15] P. Viswanath and V. Suresh Babu, "Rough-DBSCAN: A Fast Hybrid Density based Clustering Method for Large Data Sets", *Pattern Recognition Letters*, Vol. 30, No. 16, pp. 1477-1488, 2009.

[16] J. Jang and H. Jiang, "DBScan++: Towards Fast and Scalable Density Clustering", *Proceedings of 36th International Conference on Machine Learning*, Vol. 2019, pp. 5348-5359, 2019.

[17] T.N. Tran, K. Drab and M. Daszykowski, "Revised DBSCAN Algorithm to Cluster Data with Dense Adjacent Clusters", *Chemometrics and Intelligent Laboratory Systems*, Vol. 120, pp. 92-96, 2013.

[18] S. Anitha Elavarasi and J. Akilandeswari, "Survey on Clustering Algorithm and Similarity Measure for Categorical Data", *ICTACT Journal on Soft Computing*, Vol. 4, No. 2, pp. 715-722, 2014.

[19] T. Ali, S. Asghar and N.A. Sajid, "Critical Analysis of DBSCAN Variations", *Proceedings of International Conference on Information Emerging Technologies*, pp. 1-7, 2010.

[20] K. Kameshwaran and K. Malarvizhi, "Survey on Various Clustering Techniques in Data Mining", *International Journal of Science and Research*, Vol. 5, No. 2, pp. 2272-2276, 2014.

[21] W.K. Loh and Y.H. Park, "A Survey on Density-Based Clustering Algorithms", *Lecture Notes in Electrical Engineering*, Vol. 280, pp. 775-780, 2014.

[22] P. Bhattacharjee and P. Mitra, "A Survey of Density Based Clustering Algorithms", *Frontiers of Computer Science*, Vol. 15, No. 1, pp. 1-14, 2021.

[23] M.A. Ahmed, H. Baharin and P.N.E. Nohuddin, "Analysis of K-means, DBSCAN and OPTICS Cluster Algorithms on Al-Quran Verses", *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 8, pp. 248-254, 2020.

[24] S. Priyadarshini and A. Freeda, "Implementation of Adaptive DBSCAN for Cluster Analysis", *International Journal of Science Technology and Engineering*, Vol. 2, No. 9, pp. 164-168, 2016.

[25] R.G. Creţulescu, D.I. Morariu, M. Breazu and D. Volovici, "DBSCAN Algorithm for Document Clustering", *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences*, Vol. 9, No. 1, pp. 58-66, 2019.

[26] A. Mustakiml, "DBSCAN Algorithm: Twitter Text Clustering of Trend Topic Pilkada Pekanbaru", *Journal of Physics: Conference Series*, Vol. 1363, No. 1, pp. 1-9, 2019.

[27] Z. Ghaemi and M. Farnaghi, "A Varied Density-based Clustering Approach for Event Detection from Heterogeneous Twitter Data", *ISPRS International Journal of Geo-Information*, Vol. 8, No. 2, pp. 1-8, 2019.

[28] J.H. Friedman, J.L. Bentley and R.A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time", *ACM Transactions on Mathematical Software*, Vol. 3, No. 3, pp. 209-226, 1977.

[29] J.L. Bentley, "Multidimensional Binary Search Trees used for Associative Search", *Communications of the ACM*, Vol. 18, No. 9. pp. 509-517, 1975.

[30] A. Gionis, H. Mannila and P. Tsaparas, "Clustering Aggregation", *ACM Transactions on Knowledge Discovery from Data*, Vol. 1, No. 1, pp. 1-17, 2007.

[31] P. Franti and S. Sieranoja, "K-Means Properties on Six Clustering Benchmark Datasets", *Applied Intelligence*, Vol. 48, No. 12, pp. 4743–4759, 2018.

[32] R.A. Fisher, "The use of Multiple Measurements in Taxonomic Problems", *Annals of Human Genetics*, Vol. 7, No. 2, pp. 179-188, 1936.

[33] C. Dua, Dheeru and Graff, "UCI Machine Learning Repository", Available at http://archive.ics.uci.edu/ml, Accessed at 2019.

[34] C.T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters", *IEEE Transactions on Computers*, Vol. 20, No. 1, pp. 68-86, 1971.

[35] P. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis", *Journal of Computational and Applied Mathematics*, Vol. 20, pp. 53-65, 1986.