International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

A New Approach to Determine *Eps* Parameter of DBSCAN Algorithm

Fatma Ozge Ozkok[1*], Mete Celik[2]

Accepted : 30/10/2017 Published: 30/12/2017

Abstract: In recent years, data analysis has become important with increasing data volume. Clustering, which groups objects according to their similarity, has an important role in data analysis. DBSCAN is one of the most effective and popular density-based clustering algorithm and has been successfully implemented in many areas. However, it is a challenging task to determine the input parameter values of DBSCAN algorithm which are neighborhood radius *Eps* and minimum number of points MinPts. The values of these parameters significantly affect clustering performance of the algorithm. In this study, we propose AE-DBSCAN algorithm which includes a new method to determine the value of neighborhood radius *Eps* automatically. The experimental evaluations showed that the proposed method outperformed the classical method.

Keywords: *AE-DBSCAN, clustering, data mining, density-based clustering.*

1. Introduction

Clustering is one of the most important data analyses methods which groups unlabeled data based on their similarities. It is used in many areas for various applications, such as image segmentation, document retrieval, meteorology, pattern recognition, etc. [7, 9, 10].

Clustering algorithm of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) discovers clusters based on user-given parameters of neighborhood radius, *Eps,* and minimum number of points, *MinPts,* [1]. DBSCAN has the potential of discovering clusters with different densities, shapes, and sizes. It is also independent of data ordering and can handle noisy datasets [5,7,10].

However, clustering with DBSCAN is challenging due to the difficulty of determining the values of input parameters of neighborhood radius, *Eps,* and minimum number of points, *MinPts,* [5]. Determination of parameter values can be very difficult for a user who has no experience on the dataset to be clustered. Because of these reasons, automatic techniques should be developed to determine the values of these parameters.

In this study, we proposed AE-DBSCAN algorithm which includes a new method to automatically determine the value of neighborhood radius, *Eps*. The main idea of the proposed method is to assign first the sharp change in the k-dist plot as epsilon value. In our method, to find the first sharp change, we, first, generate *k-dist* plot of the dataset, and then, we take the first slope, which is above the *mean+standard deviation* of all non-zero slopes.

The rest of the paper is organized as follows. Section 2 presents the discussion of related works. Section 3 gives the basic concepts of DBSCAN algorithm. Section 4 presents the proposed method. Section 5 presents the experimental evaluations and Section 6 presents the conclusions and future works.

[1] *Engineering Faculty, Dept. of Computer Engineering, Erciyes University, 38039, Kayseri, TURKEY. {fozgeozkok, mcelik}@erciyes.edu.tr*
**Corresponding Author: Email: fozgeozkok@erciyes.edu.tr*

2. Related Works

In the literature, clustering techniques can be broadly categorized as partitional, hierarchical, density-based, grid-based, and model-based clustering algorithms [7]. This study deals with density-based clustering algorithm of DBSCAN.

Partitional clustering algorithms, such as K-means, PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications), CLARANS (Clustering Large Applications based on Randomized Search), etc., partition objects according to their similarity [7]. Hierarchical clustering algorithms, such as BIRCH (balanced iterative reducing and clustering using hierarchies), CURE (Clustering Using Representatives), ROCKS (A Robust Clustering Algorithm for Categorical Attributes), etc., produce tree structure to represent clusters [7,12,13]. Grid-based clustering algorithms, such as STING, WaveCluster, etc., discover clusters by dividing the dataset into grids [7,11,14]. Model-based clustering algorithms use probability models, such as Gaussian model, Latent Dirichlet Allocation etc., to cluster objects [7].

Density-based clustering algorithms assign dense regions as clusters and are able to cluster arbitrary shaped noisy dataset. Examples of density based clustering algorithms are OPTICS, DENCLUE, CLIQUE, and DBSCAN [1,3,7].

DBSCAN algorithm has been inspiration of several studies since it was first proposed due to its potential to discover clusters with different shapes and sizes in noisy data [1-10]. In contrast to its popularity, determination of input parameters of DBSCAN is challenging. Because of this limitation, in the literature, methods for automatic determination of the value of these parameters were proposed. In APSCAN, which is a parameter-free density based clustering algorithm, affinity propagation clustering is used to detect local densities and to determine their values of input parameters [17]. Zhou et al. proposed I-DBSCAN algorithm to determine the values of *Eps* and *MinPts* [18]. DBCLASD (Distribution-Based clustering Algorithm for Mining Large Spatial Databases) which assumes that points inside each cluster are uniformly distributed is proposed to discover arbitrarily

shaped clusters without requiring any input parameters [19]. Hou et al. proposed a parameter free clustering algorithm based on the combination of DSets (dominant sets) and DBSCAN algorithms [20]. In AGED algorithm, the value of the epsilon parameter of DBSCAN is determined based on the local densities [21]. Wang et al. proposed a modified DBSCAN algorithm to automatically determine the epsilon values for different data distributions [22]. Lakshmi et al. proposed an efficient density based subspace clustering method by computing the value of epsilon dynamically for each subspace based on the maximum spread of the data [23]. Daszykowski et al. developed an analytical approach to determine the value of epsilon using a gamma formula [24].

In this study, we propose AE-DBSCAN algorithm. It has a new method to determine the value of neighborhood radius, *Eps,* parameter of DBSCAN algorithm automatically by utilizing *k*-dist list.

3. Basic Concepts

DBSCAN algorithm requires two input parameters, such as, *Eps*, which is used to determine the neighbouring area of an object (or point) and *MinPts*, which is the minimum number of points within *Eps* radius. Basic concepts of the method given as follows [1, 7] then we present modeling of AE-DBSCAN.

3.1. Basic Concepts

Definition 1: (*Eps*-neighbourhood) For a given dataset *D*, *Eps*-neighbourhood of a point *p* is the set of neighbouring points of *q* in a given radius Eps which is expressed by
$\{q \epsilon D | \ dist(p,q) \leq Eps\}$.

Definition 2: (Directly density reachable) A point *p* is defined as density reachable from a point *q* if *p* is within *Eps*-neighbourhood from *q* and *Eps*-neighbourhood of *p* contains at least *MinPts* number of points.

Definition 3: (Density reachable) A point *p* is defined as density reachable from *q* with respect to *Eps* and *MinPts* if there is a chain of points $p1 \ ... \ p_n$, $p_1=q$, $p_n=p$ such that p_{i+1} is directly reachable from p_i.

Definition 4: (Density connected) A point *p* is defined as density connected to a point *q* with respect to *Eps* and *MinPts* if there is a point *o* such that both, *p* and *q* are density reachable from *o* with respect to *Eps* and *MinPts*.

Definition 5. (Cluster) For a given dataset *D* a cluster *C* is a non-empty subset of *D* satisfying connectivity and maximality conditions.

Connectivity: *p* is density connected to *q* with respect to *Eps* and *MinPts* for all $\{ p,q \in C \}$

Maximality: If $p \in C$, and *q* is density-reachable from *p* with respect to *Eps* and *MinPts*, then $\{ q \in C \}$ for all *p* and *q*.

Definition 6. (Core point): If the number of points in the *Eps*-neighbourhood of a point *p* is not less than *MinPts*, then the point *p* is called as a core point.

Definition 7. (Border Point): If the number of points in the *Eps*-neighbourhood of a point *p* is less than *MinPts* and at least one of neighbours of point *p* is a core point, then the point *p* is called as a border point.

Definition 8. (Noise Points): If point *p* neither a core point nor a border point than it is marked as a noise point.

3.2. Modelling AE-DBSCAN

Definition 9. (*k*-dist list): For a given dataset D, *k*-dist list is defined as the list of the sorted values representing k^{th} nearest neighbor distances of each point in the dataset.

Definition 10. (slope of a point with respect to another point): For a given *k*-dist list $k\text{-}dist= (k_1, k_2, k_3, ..., k_n)$, the slope of point k_i with respect to next point k_{i+1} is defined as the slope of line segment k_ik_{i+1}.

To find the slope of a point with respect to another point, different definitions or line segment representations can be used. In this study, to calculate the slope of a point k_i , we used its sequential point of k_{i+1} in the line segment representation.

4. The Proposed Method AE-DBSCAN

This section presents the proposed algorithm AE-DBSCAN. In contrary to the classical DBSCAN algorithm, it finds the *Eps* value automatically. The proposed AE-DBSCAN algorithm requires a dataset and a *k* value (or *MinPts*) as inputs. The proposed algorithm has two stages, such as determining the value of *Eps* and clustering the dataset. The first stage of the algorithm discovers the value of neighbourhood radius *Eps* and then this *Eps* value is used in the second stage with *k* (or *MinPts*) value to discover the clusters out of the dataset. The clustering stage of the algorithm works similar to the classical DBSCAN algorithm. The pseudo code of the algorithm is given in Algorithm 1.

Alg. 1. The proposed AE-DBSCAN algorithm

Input: Dataset, *k* (or *MinPts*)
Output: Discovered clusters
Algorithm:

Determination of Epsilon value

1. Calculate *k*-dist values of all point in dataset
2. Sort the *k*-dist values and draw *k*-dist plot
3. Calculate slopes of each changes (or point) in *k*-dist plot
4. Calculate the *mean* and *standard deviation* of non-zero slopes.
5. Find the first slope which is above *mean(slope)+standard deviation(slope)*.
6. Find corresponding *k*-dist value of the found slope in step 5 and assign this value as *Eps*.
7. Assign *k* as *MinPts*

Clustering the dataset

8. For each point *p* in dataset
9. Retrieve all point density-reachable from *p* with respect to *Eps*
10. If neighbour of *p* contains at least *MinPts* number of points then *p* is a core point and a cluster is formed.
11. If *p* is not a core object but one of its neighbours a core object, *p* is assigned as border point.
12. If *p* neither core point nor border point, mark it as noise point.
13. Output the discovered clusters

The aim of the determination of *Eps* value is to find the value of *Eps* by utilizing *k*-dist plot. The sharp changes in the *k*-dist plot are candidate *Eps* values. In the proposed approach, first, the *k*-dist values are calculated by taking the distance of each point to its k^{th} nearest neighbour. Then the *k*-dist values are sorted. Using these sorted values the *k*-dist plot is drawn. The sharp changes in this plot represent candidate *Eps* values. To determine the sharp changes we calculated slopes of each point with respect to the next point. The slope of a point k_i is calculated as the absolute differences of k^{th} neighbour distances of k_i and k_{i+1}. Then, we calculated mean and standard deviation of the non-zero slopes. In this stage, the slopes whose values are zero are excluded to focus on changes on the *k*-dist plot. In our method, the first slope which is above the *mean(slopes)+standard deviation(slopes)* is determined and the corresponding *k*-dist value of this slope is selected as *Eps* value. We also tested two other strategies to find *Eps* value. In the first strategy, the first slope which is above the *mean(slopes)+2×standard deviation(slopes)* is determined and

corresponding *k*-dist value of this slope is selected as *Eps* value. In the second strategy, the first slope which is between *mean(slopes)-standard deviation(slopes)* and *mean(slopes)+ standard deviation(slopes*) is determined and corresponding *k*-dist value of this slope is selected as *Eps* value However, our empirical results showed that the method which finds the first slope which is above the *mean(slopes)+standard deviation(slopes)* gave the best results.

In the clustering stage of the AE-DBSCAN algorithm, it clusters the dataset using *k* as *MinPts* and the discovered *Eps* value. This stage is run for each point to discover clusters by marking each point as core point or border point or noise point.

5. Experimental Results

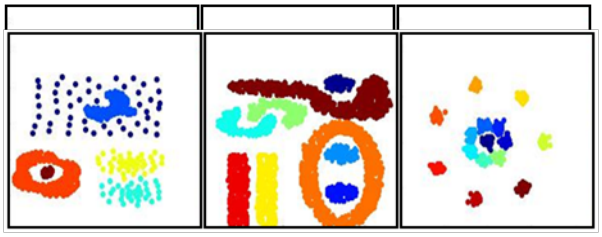
We compared the performance of the proposed method AE-DBSCAN with that of the analytical method [24]. Experiments were performed to answer the following four questions:

- What are the effect of different densities?
- What are the effect of different sizes?
- What are the effect of adherent clusters?
- What is the effect of *MinPts* parameter?

Experiments were conducted on an Intel Core i7 2.4 GHz computer with 8 GB RAM.

5.1. Datasets

To evaluate the performance of proposed algorithm we used 3 synthetic 2-D dataset [16, 25] . First dataset, called compound, has 6 clusters with varied density. The dataset contains 399 points (Fig. 1(a)). Differences in the density among clusters makes it difficult to cluster correctly [25]. Second dataset, called Complex9, has 9 clusters. The dataset has different sized clusters and contains 3031 points (Fig. 1(b)) [16]. Third dataset, called R15, has 15 clusters. The dataset has adherent clusters and consists of 600 points (Fig. 1(c)) [25].



(a) (b) (c)

Fig. 1. (a) Compound dataset (Dataset1), (b) Complex9 dataset (Dataset2), (c) R15 dataset (Dataset3) (best viewed in color)

5.2. Experiments

5.2.1. Performances of Different Strategies to Determine *Eps*

In this section, we compared the performance of epsilon finding strategies using Compound Dataset (Dataset1), Complex9 Dataset (Dataset2), and R15 Dataset (Dataset3). The first strategy finds the epsilon value by finding *k*-dist value whose slope is above the *mean(slopes)+2×standard deviation(slopes)* and the second strategy finds the epsilon value by finding the *k*-dist value whose slope is in between *mean(slopes)-standard deviation(slopes)* and *mean(slopes)+standard deviation(slopes*). In this experiment, *k* value were selected as 3 for Compound Dataset, 7 for Complex9 dataset, and 4 for R15 dataset.

Fig. 2 presents clustering results of the first strategy, Fig. 3 presents the clustering results of the second strategy, and Fig. 4 presents the clustering results of the proposed AE-DBSCAN algorithm for three datasets. As can be seen, AE-DBSCAN outperformed other strategies. The clustering accuracy of the first strategy is 69.04%, 90.42%, and 84.46% for Compound, Complex9, and R15 dataset, respectively. The clustering accuracy of the second strategy is 26.97%, 42.73%, and 21.49% for Compound, Complex9, and R15 dataset, respectively. The clustering accuracy of the AE-DBSCAN is 96.56%, 99.81%, and R15 96.13% for Compound, Complex9, and R15 dataset, respectively.

(a) (b) (c)

Fig. 2. Performances of the first strategy (a) Compound dataset (k=3), (b) Complex9 dataset (k=7), (c) R15 dataset (k=4) (best viewed in color)

(a) (b) (c)

Fig. 3. Performances of the second strategy (a) Compound dataset (k=3), (b) Complex9 dataset (k=7), (c) R15 dataset (k=4) (best viewed in color)

Fig. 4. Performances of the AE-DBSCAN (a) Compound dataset (k=3), (b) Complex9 dataset (k=7), (c) R15 dataset (k=4) (best viewed in color)

5.2.2. The Effect of Different Density

In this experiment, we used Compound Dataset to evaluate the performances of the proposed method AE-DBSCAN and analytical method [24] since the dataset has clusters with different densities. For the AE-DBSCAN algorithm, the value of *k* (or *MinPts*) was set to 3 and for the analytical method the value of *k* was set to 3 and 16. The results of the experiment can be seen in Fig. 5. As can be seen, AE-DBSCAN method can find clusters correctly with the clustering accuracy of 96.56% (Fig. 5 (a)). In contrast, the analytical method cannot discover all the clusters. When *k* is 3, the analytical method assigns some points of the clusters as noise (Fig. 5(b)). When *k* is 16, the analytical method combines some of the clusters and so it cannot discover all clusters completely (Fig. 5(c)). The clustering accuracy of analytical method is 89.15% for *k=3* and 62.61 % for *k=16*.

(a) (b) (c)

Fig. 5. Effect of different density (a) AE-DBSCAN (*k*=3), (b) analytical method for *k*=3, (c) analytical method for *k*=16 (best viewed in color)

5.2.3. The Effect of Different Size

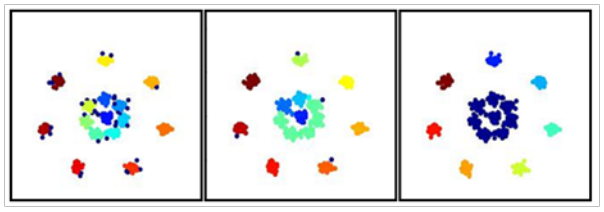
In this experiment, we used Complex9 Dataset to evaluate the performances of the proposed method AE-DBSCAN and analytical method [24] since it has different-sized clusters. For the AE-DBSCAN algorithm, the value of *k* (or *MinPts*) was set to 7 and for the analytical method the value of *k* was set to 7 and 4. The clustering results of both methods can be seen in Fig. 6. The clustering accuracy of AE-DBSCAN is 99.81% and the clustering accuracy of analytical method 90.42% for *k=7* and 99.78% for *k=4* (Fig. 6).

(a) (b) (c)

Fig. 6. Effect of different size (a) AE-DBSCAN (k=7), (b) analytical method for k=7, (c) analytical method for k=4 (best viewed in color)

5.2.4. The Effect of Adherent Clusters

In this experiment, we used R15 Dataset to evaluate the performances of the proposed method AE-DBSCAN and analytical method [24] since the dataset has adherent clusters. For the AE-DBSCAN algorithm, the value of *k* (or *MinPts*) was set to 4 and for the analytical method the value of *k* was set to 2 and 4. The clustering results of both method can be seen in Fig. 7. The clustering accuracy of AE-DBSCAN is 96.13% and the clustering accuracy of analytical method 76.9% for *k=2* and 58.18% for *k=4* (Fig. 7).



(a) (b) (c)

Fig. 7. Effect of adherent cluster (a) AE-DBSCAN (k=4), (b) analytical method for *k*=2), (c) analytical method for *k*=4 (best viewed in color)

5.2.5. The Effect of *k* (or *MinPts*) Parameter

In this section, we evaluated the effect of *k* (or *MinPts*) parameter on the proposed AE-DBSCAN algorithm.

First, we evaluated how the accuracy of clustering changes as the value of k increases on three datasets. As can be seen in Fig. 8, for low and high *k* values the accuracy of clustering gets worse. In the experiments, we figure out that when the value of *MinPts is* around 3 or 4, the classification accuracy becomes higher.

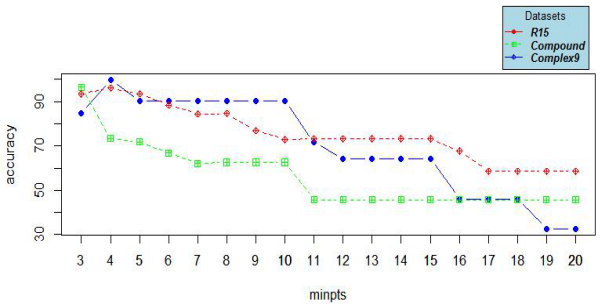


Fig. 8. Effect of *MinPts* parameter on clustering accuracy

In the second experiment, we evaluated the effect of *MinPts* value on the number of clusters (Fig. 9). As seen in Fig. 9, as the value of *MinPts* increases, the number of clusters decrease.

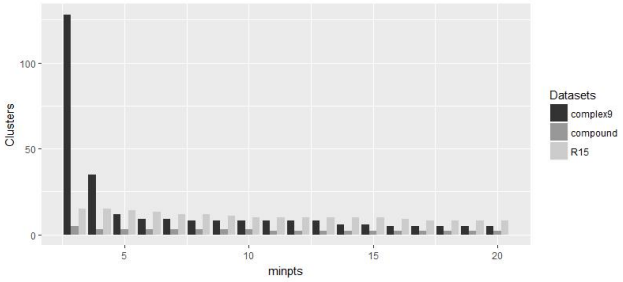


Fig. 9. Effect *MinPts* parameter on the number of clusters

6. Conclusion

DBSCAN has the potential of discovering clusters with different densities, shapes, and sizes. It is independent of data ordering and can handle noisy datasets. However, determining the value of neighborhood radius *Eps* is a difficult task. In this study, AE-DBSCAN, which includes a new method for determination of the value of neighborhood radius *Eps* automatically, is proposed. Experimental results showed that the proposed AE-DBSCAN outperformed the classical algorithm [24].

As the future work, we plan to study on more datasets, to improve the proposed method and to compare it with the performances of other algorithms.

References

[1] M. Ester, H.-P. Kriegel, and X. Xu "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD,* Oregon, USA, 1996, pp. 226-231.

[2] X. P. Yu, D. Zhou, and Y. Zhou, "A New Clustering Algorithm Based on Distance and Density," in *Proc. ICSSSM*, Chongquing, China, 2005, pp. 1016-1021

[3] S. K. Popat and M. Emmanuel, "Review and Comparative Study of

Clustering Techniques," *Int. J. of Computer Science and Information Technologies*, vol. 5, no.1, pp. 805–12, 2014.

[4] P. Liu, D. Zhou, and N. J. Wu,"VDBSCAN: Varied density based spatial clustering of applications with noise," in *Proc. ICSSSM*, Chengdu, China, 2007, pp 1-4.

[5] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future." in *Proc. ICADIWT*, Bangalore, India, 2014, pp. 232-238.

[6] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar "A density based algorithm for discovering density varied clusters in large spatial databases," *Int. J. of Computer App.*, vol. 3, no. 6, pp. 1-4, 2010.

[7] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[8] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[9] M. Celik, F. Dadaser-Celik, and A. Dokuz, "Anomaly detection in temperature data using dbscan algorithm," in *Proc. INISTA*, Istanbul, Turkey, 2011, pp. 91–95.

[10] P. N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," Boston Addison-Wesley, April 2005.

[11] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wave Cluster: A multi-resolution clustering approach for very large spatial databases," in *Proc. VLDB*, San Francisco, CA, 1998, pp.428-439.

[12] G. Sudipto, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large Databases," in *Proc. ACM SIGMOD*, Seattle, WA, 1998, pp.73-84.

[13] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD*, 1996, pp. 103–114.

[14] W. Wang, J. Yang, and R. R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc VLDB*, San Francisco, CA, USA, 1997, pp. 186–195.

[15] M. Halkidi, Y. Batistakis, and M. Varzirgiannis, "On clustering validation techniques," *J. of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[16] Karypis, G., Han, E.H., and Kumar, V.: "Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling," *IEEE Computer*, vol. 32, no. 8, pp 68-75, August 1999.

[17] X. Chen, W. Liu, H. Qui and J. Lai, "APSCAN: A parameter free algorithm for clustering", *Pattern Recognition Letters,* vol. 32, pp. 973-986, 2011.

[18] H. Zhou, P. Wang, and H. Li, "Research on adaptive parameters determination in DBSCAN algorithm," *J. of Information & Computational Science*, vol. 9, no. 7, pp. 1967-1973, 2012.

[19] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander "A distribution-based clustering algorithm for mining in large spatial databases", in *Proc. ICDE*, Orlando, USA, 1998.

[20] J. Hou, H. Gao, and X. Li, "DSets-DBSCAN: a parameter-free clustering algorithm", *IEEE Transaction on Image Processing*, vol.25, no. 7, pp. 3182-3193, 2016.

[21] N. Soni and A. Ganatra, " AGED (Automatic Generation of Eps for DBSCAN, *Int. J. of Computer Science and Information Security (IJSIS),*, vol. 14, no. 5, pp. 536-559, 2016.

[22] W.-T. Wang, Y.-L. Wu, C.-Y. Tang, and M.-K. Hor, "Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data ", in *Proc. ICMLC,* Guangzhou, 2015, pp. 445-451.

[23] B. J. Lakshmi, K. B. Madhuri, and M. Shashi, "An efficient algorithm for density based subspace clustering with dynamic parameter setting", *Int. J. of Information Technology and Computer Science (IJITCS)*, vol. 6 , 2017, pp. 27-33.

[24] M. Daszykowski, B. Walczak, and D. L. Massart, "Looking for Natural Patterns in Data. Part 1: Density Based Approach", *Chemometrics and Intelligent Laboratory Systems,* vol. 56, no. 2, pp. 83-92, 2001.

[25] Clustering datasets, Available: http://cs.uef.fi/sipu/datasets/. Accessed on: April 23, 2017.