DBSCAN: Past, Present and Future

Kamran Khan
Department of Computer Science
SZABIST
Islamabad, Pakistan
Kamran_3388@yahoo.com

Saif Ur Rehman, Kamran Aziz
Center of Excellence in Data Engineering
Mohammad Ali Jinnah University
Islamabad, Pakistan
Saifi.ur.rehman@gmail.com,
kamrandik@gmail.com

Simon Fong
Department of Computer and Information Science
University of Macau
Taipa, Macau SAR
ccfong@umac.mo

S.Sarasvady
Amrita Vishwa Vidyapeetham University
Ettimadai
Coimbatore – 641 112. India
s_sarasvady@cb.amrita.edu

Abstract— **Data Mining is all about data analysis techniques. It is useful for extracting hidden and interesting patterns from large datasets. Clustering techniques are important when it comes to extracting knowledge from large amount of spatial data collected from various applications including GIS, satellite images, X-ray crystallography, remote sensing and environmental assessment and planning etc. To extract useful pattern from these complex data sources several popular spatial data clustering techniques have been proposed. DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a pioneer density based algorithm. It can discover clusters of any arbitrary shape and size in databases containing even noise and outliers. DBSCAN however are known to have a number of problems such as: (a) it requires user's input to specify parameter values for executing the algorithm; (b) it is prone to dilemma in deciding meaningful clusters from datasets with varying densities; (c) and it incurs certain computational complexity. Many researchers attempted to enhance the basic DBSCAN algorithm, in order to overcome these drawbacks, such as VDBSCAN, FDBSCAN, DD_DBSCAN, and IDBSCAN. In this study, we survey over different variations of DBSCAN algorithms that were proposed so far. These variations are critically evaluated and their limitations are also listed.**

Index Terms— **Clustering, density, sampling, DBSCAN, spatial data, data mining algorithms.**

I. Introduction

Datamining is the process of extracting hidden and interesting patterns or characteristics from very large datasets and using it in decision making and prediction of future behavior. This increases the need for efficient and effective analysis methods to make use of this information. One of the tasks is clustering where a set of objects is divided into several clusters where the intra-cluster similarity is maximized and the inter-cluster similarity is minimized [15].

The disadvantages in most of the traditional clustering algorithms are high computational complexity and that they do not scale well with the size of the very large datasets, so the development of enhanced clustering algorithms has received a lot of attention in the last few years. There are different clustering methods that can be used for handling very large datasets [15]. These techniques can be categorized into partitioning, hierarchal, grid-based, density-based, model-based and constrain-based methods.

Techniques under the partitioning category are PAM [2], CLARA [2], and CLARANS [3]. These methods segment data into k groups where the value of k is supplied by the user. The well-known algorithms for hierarchical category are CURE [4] and CHEMELEON [5]. The hierarchical algorithms built a tree like structure, called the dendogram, to reveal a cluster structure on every iteration. Grid-based clustering techniques such as CLIQUE [6], ENCLUS [7], and WaveCluster [8] attempts to segment the space into finite number of cells which make it possible to perform all the clustering operations. Density-based techniques were introduced to determine the arbitrary shaped cluster in spatial databases having noise. The DBSCAN [9], DENCLUE [10] and OPTICS [11] are commonly used density-based clustering techniques. Examples of the model-based clustering algorithms are COBWEB [12] and SOM [13]. These algorithms optimize the best fit between the given data and a mathematical model. The constraint-based clustering techniques find the clusters that satisfy the user-specified preference or constraint Algorithms proposed under each of these categories try to challenge the clustering problems treating the large amount of data in large database. However, none of them are most effective. A clustering technique is considered to be good if it satisfies the following requirements [1]:

- Minimal requirements of the domain knowledge to determine the values of its input parameters, which is very important problem especially for large data sets.
- Discovery of arbitrary shaped clusters.
- Good efficiency on large data sets.

The density-based clustering algorithms are useful to discover clusters from the datasets with arbitrary shape and of large size. These algorithms typically cluster as dense regions of points in the data space that are separated by regions of low density. DBSCAN [9] is the first density based clustering technique. It grows clusters according to a density based connectivity analysis.

In this paper, we have explored the different DBSCAN enhancements proposed so far. These are proposed as modification and improvements to the DBSCAN for the purpose of effective clustering analysis of the underlying datasets. These modified DBSCAN techniques include VDBSCAN [14], FDBSCAN [16], GRIDBSCAN [17], IDBSCAN [18], EDBSCAN [19] etc. These are discussed with

978-1-4799-2259-14/$31.00©2014

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.

their strengths and weaknesses in this study. In addition, a critical evaluation on these research works is also provided with respect to the different parameters listed in TABLE-1. Furthermore, some future directions are also highlighted at this end of the study.

The rest of this paper is organized as follow. More discussion on DBSCAN algorithm is provided in Section II. Section III summaries the existing density-based clustering variation of DBSCAN with their advantages and limitations. Section IV outlines the critical review of the different DBSCAN variations. Finally, concluding points and some future direction are provided in Section IV.

II. DBSCAN ALGORITHM

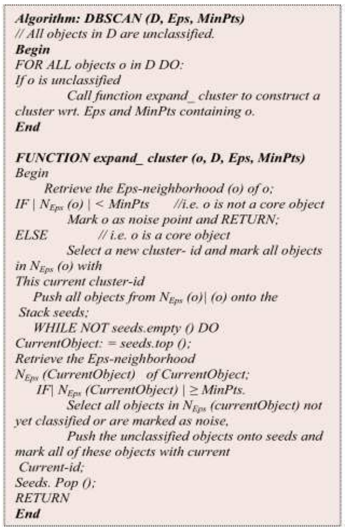
DBSCAN is the first density based clustering algorithm. It was proposed by Ester et al. in 1996, and it was designed to cluster data of arbitrary shapes in the presence of noise in spatial and non-spatial high dimensional databases. The key idea of DBSCAN is that for each object of a cluster the neighborhood of a given radius (*Eps*) has to contain at least a minimum number of objects ($MinPts$), which means that the cardinality of the neighborhood has to exceed some threshold. The ϵ-neighborhood of an arbitrary point 'p' is defined as,

$$N_{Eps} = \{q \in D \,/ dist\,(p,q) < Eps\} \qquad (1)$$

Here, D is the database of objects. If the ϵ –neighborhoods of a point P at least contain a minimal number of points, and then this point is called core point. The core point is defined as:

$$N_{Eps}\,(P) \;\; > MinPts \qquad (2)$$

Here *Eps* and *MinPts* are the user's specified parameters which mean the radius of the neighborhood and minimum number of points in the ϵ-neighborhood of a core point respectively. If this condition is not satisfied then this point is considered as non-core point.



```
Algorithm: DBSCAN (D, Eps, MinPts)
// All objects in D are unclassified.
Begin
FOR ALL objects o in D DO:
If o is unclassified
            Call function expand_ cluster to construct a
cluster wrt. Eps and MinPts containing o.
End

FUNCTION expand_ cluster (o, D, Eps, MinPts)
Begin
       Retrieve the Eps-neighborhood (o) of o;
IF | $N_{Eps}$ (o) | < MinPts       //i.e. o is not a core object
            Mark o as noise point and RETURN;
ELSE           // i.e. o is a core object
            Select a new cluster- id and mark all objects
in $N_{Eps}$ (o) with
This current cluster-id
    Push all objects from $N_{Eps}$ (o)| (o) onto the
 Stack seeds;
    WHILE NOT seeds.empty () DO
CurrentObject: = seeds.top ();
Retrieve the Eps-neighborhood
$N_{Eps}$ (CurrentObject)   of CurrentObject;
     IF| $N_{Eps}$ (CurrentObject) | $\geq$ MinPts.
            Select all objects in $N_{Eps}$ (currentObject) not
yet classified or are marked as noise,
            Push the unclassified objects onto seeds and
mark all of these objects with current
 Current-id;
Seeds. Pop ();
RETURN
End
```

Fig. 1. Pseudo code of DBSCAN

DBSACN searches for the clusters by checking the ϵ -neighborhood of each object in the dataset. If the ϵ -neighborhood of an object p contains more than *MinPts*, a new cluster with p as a core object is created. It then iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a new density-reachable cluster. The process terminates when no new object can be added to any cluster [6]. The pseudo code of *DBSCAN* algorithm is shown in Figure 1.

III. LITERATURE REVIEW

This section summarizes the different proposed variations of DBSCAN with their major research contributions and limitations. DBSCAN [18] is an improved version of the DBSCAN, introducing a sampling technique to address the two issue of DBSCAN and its variations: (1) - to make it effective while dealing with large volume of spatial data objects; (2) - reduces the I/O cost. From the experiments sampling based IDBSCAN outperforms the DBSCAN in minimizing I/O cost and memory requirement for clustering with no compromise on the quality of cluster. Although, IDBSCAN improved the I/O cost but still it needs users to specify the value of threshold parameters manually.

In 2004, El-Sonbaty *et al.* [15] provided an enhancement version of DBSCAN, to generate efficient clustering result from large size of datasets using the following procedure. In the pre-processing step, the dataset to be analyzed is partitioned using CLARANS [3]. By this partitioning search effort for the core object is minimized. As the searching is limited to the single partitioned region rather than scanning the whole dataset for the core object. Afterwards, the dense region obtained from each of these partitioned regions, are merged together. This merging results in the required number of clusters. This merging is done using relative inter connectivity defined in [5]. The major success points of their study include: (1) It takes less time to cluster the dataset, by partitioning dataset and then limiting the search space to only of the partitioned data object rather than dataset as a whole; (2) Memory efficient, due to requiring small buffer size [15]. Like DBSCAN this work also requires the users to input *Eps* and *MinPts* parameter values.

In 2005, Yu *et al.* [25] developed a density-based clustering algorithm called KNNDBSCAN (K-nearest neighbors DBSCAN). The quality of clustering result, either by applying DBSCAN or any other of its extension such as VDBSCAN [14], EDBSCAN [19] etc. mainly depends on the appropriate values for the density threshold (ϵ, *MinPts*) values. The main problem associated with most of the density-based clustering algorithms is the determination of the global values for these density thresholds. Unlike DBSCAN, which needs two threshold parameters to process, *KNNDBSCAN* is based on the single user input parameter values which is "*K*". This parameter is able to unsupervisedly determine the density thresholds (ϵ, *MinPts*) [25]. The resulting clustering is not affected due to the value of "*K*". The *KNNDBSCAN* merges two approaches to discover the arbitrary shaped clusters from the density-based datasets. These two approaches are K-nearest neighbors and DBSCAN. This combined (KNN and

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.

DBSCAN) approach operates as fallow. In the first step, for each data point window width and related neighbors are determined. The whole dataset is then partitioned into Fuzzy clusters (FCs). This partitioning is carrying out with the help of KNN based on KDE-based rules. This will reduce the number of scans, thus resulting in improved performance. In the next step, density threshold ϵ and *MinPts* are calculated for each FCs. This calculation is determined according to the Entropy theory. In the final step, each local threshold values are mapped to the global value of ϵ and at the same time each FCs is clustered in parallel independently. This speeds up the clustering process and also saves the main memory by keeping the only FC which is to be clustered rather than the whole dataset. The major success points include: (1) automating computation of density threshold; (2) since it can clusters the datasets in parallel, so it speeds up the clustering process when it is compared with DBSCAN; (3) as only the partition of a dataset is taken for clustering so it save the main memory, as whole dataset is not stored in memory while clustering.

GRIDBSCAN [17] is another important variation of DBSCAN. That tackled the issues associated with most of the density-based clustering algorithms is that they are not effective to perform clustering accurately in the presence of clusters with different densities. Uncu *et al.* proposed a three-level clustering mechanism to provide a solution to this problem. In the first level, it provides appropriate grids such that density is similar in each grid. In the next level, it merges the cells having same densities. At this level, the appropriate value of ϵ and *MinPts* are also identified in each grid. In the final step, the DBSCAN algorithm is applied with these identified parameters values to obtain the required final number of clusters. Although accuracy of GRIDBSCAN is better as compare to DBSCAN but on the other hand GRIDBSCAN may be handy in terms of computational complexity when applied on large spatial data.

In 2006, Viswanth et al. [22] tried to improve the DBSCAN by following a hybrid clustering technique. This method is called l-DBSCAN, where *l* stands for leaders. The hybrid-clustering is the novel clustering technique. I-DBSCAN technique work as follow: (1) find the suitable prototypes from large dataset; (2) and then it uses the clustering methods on these selected prototypes. The leader clustering method is a fast method and it runs in linear time of the input dataset size. In l-DGBSCAN, first two types of prototypes are derived with the help of leader clustering method. Afterwards DBSCAN is applied to perform density based clustering on this prototype respectively. For details refer literature in [22]. Although I-DBSCAN can achieve clustering results in the same way as that of DBSCAN, the run time for I-DBSCAN is comparatively less than time required by DBSCAN.

In 2006, Liu [16] proposed an enhanced version of DBSCAN clustering technique called “Fast Density Based clustering algorithm for large Database”, (FDBSCAN).This was introduced to overcome: (1) its slow speed (slow in comparison due to neighborhood query for each object); (2) and setting threshold value of DBSCAN algorithm. The FDBSCAN starts by ordering the dataset object by a certain dimensional co-ordinates. Then it considers a point having minimal index, retrieves its ϵ -neighborhood. If this point is proved as a core object then a new cluster is created to label all objects in its neighborhood. In this way, next unlabelled point is analyzed outside the core object to expand clusters. When all the points are analyzed for clustering then these objects are further passed through a Kernel function. This will ensure the distribution of object as uniform as possible. From experimental proofs author concluded that FDBSCAN can accurately cluster the datasets and it can achieve its result in less time as compare to DBSCAN. It is time efficient algorithm; as it decreases the time by ignoring the region objects already clustered. Before clustering, the FDBSCAN applies the Kernel function on the labelled objects; which result in more accurate clusters [16].

In 2007, Liu et al. [14] have devised a new enhancement to DBSCAN, called VDBSCAN, in order to analyze the dataset having varied densities. It is a two-step procedure. In the first step, the values of *Eps* are calculated for different densities according to a K-dist. plotting. These calculated values are then further used to analyze the clusters with different densities. In the second step, the DBSCAN algorithm is applied with the parameter *Eps* values calculated in previously discussed step. It ensures that all of the clusters with corresponding densities are clustered. In VDBSCAN, once the points are clustered they are given a label indicating that they have been clustered and there is no need to process these points again. In VDBSCAN the value of *Eps* are computed automatically. It is the basic approach that is used to compute *Eps* and *MinPts* value by looking at the behaviour of the distance from a point to its *K* the neighbor.

An enhancement of DBSCAN algorithm is provided by [19] called *EDBSCAN* to handle the local density variation within the cluster and for a good clustering a significant density variation may be allowed within the cluster. EDBSCAN finds the density variation of a core object with respect to its ϵ -neighborhood. If these variations turn out to be less than a user specified threshold value and satisfy homogeneity index with respect to its ϵ–neighborhood then this core object will be allowed for expansion. EDBSCAN uses two users’ specified parameters which are *Minpts* and *Maxpts*, and *Minpts* < *Maxpts* < 20, in addition to the basic DBSCAN parameters. These parameters are used to limit the amount of allowed local density variation within the cluster. Furthermore, EDBSCAN and DBSCAN were tested using synthetic datasets. From the EDBSCAN experimental results authors have concluded that EDBSCAN is effective than DBSCAN in the sense that it can handle the local density variations existing within cluster effectively as compare to DBSCAN. EDBSCAN make use of two parameters which are used to achieve its goal of locating the local density within the cluster. Their values require manual user input.

DD_DBSCAN [20] is another variation of the DBSCAN to identify those clusters that differ in densities. In addition, it can find the clusters having different shapes, sizes and differ in local density. It introduced ‘α’ as upper limit during the expansion of clusters. DD_DBSCAN can find clusters that

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.

represent relatively uniform regions without being separated by sparse region. One of the limitations of this work is that this technique could not handle the density variation within the cluster.

In 2008 Xiaoyun et al. [21] pointed out two problems of *DBSCAN*. (1) It cannot choose density threshold (*MinPts* and ϵ) according to distribution of data space. (2) When it is used with large datasets then running of DBSCAN cost too much to get the dataset clustered. The paper introduces a new enhancement to DBSCAN called GMDBSCAN. It is presented to deal with these two issues. GMDBSCAN introduces the concept of local *MinPts*. It clusters the dataset in different regions by co- responding local *MinPts* and use grid-density as the approximation of local *MinPts*. This clustering is done in each grid using *DBSCAN*. In *GMDBSCAN*, first of all the data space is partitioned into grids. Afterwards, for each of the grid SP-Tree is constructed. Then in the next step Bitmap is farmed by computing the distance of the data points which exists in the same or adjacent grid. Before performing actually clustering GMDBSCAN select the parameter values by using the following equations. For details readers are referred to [21]:

$$MinPts = [Factor * G\ D] \tag{3}$$

$$Eps = 2/\,[d\,/\,sqrt\,(n\,/\,k)] \tag{4}$$

In the next step, by using the *MinPts* and *Eps* value, the GMDBSCAN algorithm can perform clustering on each of the identified grid using DBSCAN algorithm. Two sub-clusters are merged if they have same points and two having the same density. For noise and border point, GMDBSCAN set a parameter according to the size of dataset. A cluster is classified as noise if the amount of data in a cluster is less than this parameter value. The contribution of GMDBSCAN, is that it provides a way to calculate the value of *MinPts* and *Eps* automatically.

In 2007, Birnat et al. [23] improved the DBSCAN algorithm by introducing a new density-based clustering algorithm called *ST-DBSCAN*. This algorithm is proposed for clustering spatial–temporal data. This research work tries to enhance the *DBSCAN* in three following different ways. (1) To cluster the spatial-temporal data according to its non-spatial, spatial and temporal attributes. (2) ST-DBSCAN introduces the Density-Function for each cluster in the given dataset. This factor has solved the problem of DBSCAN of not detecting some of the noise points when the dataset contains the clusters of different densities. (3) The third major modification made into DBSCAN is that *ST*-DBSCAN has solved the problem of identifying the adjacent clusters. This modification is achieved by the comparison of the average value of a cluster with the new coming value. ST-DBSCAN uses four parameters for its changes introduced in DBSCAN. Form the experiments result authors have proved that he changes they suggested for the enhancement of DBSCAN appear to be efficient and effective when the dataset to be clustered has spatial-temporal data characteristics.

In 2008, Mahran et al.[24]proposed a Grid-based clustering technique called GRIDBSCAN. It is proposed to enhance the performance of DBSCAN algorithm. This algorithm is based on divide and conquers technique. This algorithm operates as follow. (1) A grid is created that partitions the surrounding space based on the number of cells provided by the users. (2) Furthermore, the given dataset is partitioned according to the cells of the grid. (3) Now, DBSCAN is applied on each of these partitions separately. (4) Finally the resulting clusters from the previous step are merged together in order to get the final global clustering results. GRIDBSCAN run much faster than DBSCAN, and also the performance of GRIDBSCAN is better than Enhanced DBSCAN [19], having the same mechanism of partitioning and merging to enhance the clustering results of DBSCAN as that of GRIDBSCAN.

The Density Clustering Based on Radius of Data (DCBRD) [1] is another DBSCAN enhanced version. It has overcome the problem of dependence on the user to supply the density threshed parameter values *Eps* and *MinPts*. It uses the knowledge acquired from the underlying datasets and then performs the clustering. It can discover clusters of arbitrary size and shapes from large datasets having large dimensions. It does not depend on the users specified input parameters. The clustering is done in DCBRD with the help of two stages. In the first stage, data space is partitioned into overlapped circular regions (sphere of hypersphere).Partitioning is done in such a way that radius of each region is larger than the expected density threshed *Eps*. The radius of the data space is calculated by the formula devised by DCBRD, for detail refer to literature [1]. The radius of the circles depends on the dimension and regions of the data space. After partitioning the data space, DBSCAN is applied on each of the region using the optimal value of the *Eps*. This value is calculated from the circle that span to all data space. In this way the dataset is clustered which is more efficient when its results are, experimentally, compared with the DBSCAN technique result on the same datasets.

In 2010, Ram et al. proposed DVBSCAN [27] algorithm for handling the local density variation within the cluster. To achieve this objective they Ram et al. have incorporated the following input parameters: minimum objects (μ), radius, and threshold values (α, λ). It calculates the growing cluster density mean and then the cluster density variance for any core object, which is supposed to be expanded further by considering density of its *e*-neighborhood with respect to cluster density mean. If cluster density variance for a core object is less than or equal to a threshold value and is also satisfying the cluster similarity index, then it will allow the core object for expansion.

In 2013, a new enhancement of DBSCAN has been introduced. This is called Dynamic Method DBSCAN (DMDBSCAN). This technique has pointed out that in clusters, generated by DBSCAN, there is wide density variation. As compare to DBSCAN which uses global *Eps,* this technique has successfully given the method to compute *Eps* automatically for each of the different density level in the dataset based on *k-dist.* plot. The major success of this technique included: (1) clusters generated are easy to interpret;

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.

(2) no limit on the shape of the clusters generated. From the experiments Mohammad et al. have concluded that their DMDBSCAN is giving more promising results as compare to DBSCAN and DVBSCAN [27].

In 2013, Manisha et al. [28] have proposed an enhanced algorithm that automatically selects the input parameters based on the knowledge acquired from the dataset. This technique requires one input parameter and discovers arbitrary size and shaped clusters. It is efficient even for large data sets. It can detect the cluster automatically by explicitly finding the input parameters and finding clusters with varying density. The basic idea is that, before adopting traditional DBSCAN algorithm, some methods are used to select several values of parameter *Eps* for different densities according to a *k-dist* plot. With different values of *Eps*, it is possible to find out clusters with varied densities simultaneity [4]. For each value of *Eps*, DBSCAN algorithm is adopted in order to make sure that all the clusters with respect to corresponding density are clustered. And for the next process, the points that have been clustered are ignored, which avoids marking both denser areas and sparser ones as one cluster. The experimental results shows that the proposed algorithm can detect the clusters of varied density with different shapes and sizes from large amount of data which contains noise and outliers, requires only one input parameters and gives better output then the DBSCAN algorithm.

IV. CRITICAL REVIEW

This section primarily reflects the comparison and contrast of the above reviewed literature regarding the different DBSCAN variations and modifications. It identifies the similarities and differences among the various research works on the DBSCAN algorithm enhancements. The critical review is given in Table 1.This will help for the future research in the DBSCAN modification and enhancements.

Liu et al. [14] have modified the DBSCAN to deal with the datasets that are varied in densities. Their algorithm is called VDBSCAN. VDBSCAN is able to calculate the density threshold parameters automatically based on the K-distance plotting. Its computational complexity is same as that of DBSCAN. The same work is explored in GRIDBSCAN [17] to deal with the dataset that have cluster with different densities.

The research work proposed in [14, 17] are identical in that they do not require any user supplied input parameters. The study carried out by [17] can cluster the dataset efficiently as that of [14] but [17] is expensive as compare to that of [14]. Fahim et al. [1] carried out the research in the same dimension as that of [14] in the sense that it does not require any user supplied density threshold parameters. These parameters are ϵ, *MinPts*.

Uncu et al. [17] have introduced an extension of DBSCAN such that it can cluster the datasets having different densities. The author in [17] has used the concept of grid while performing clustering. Its clustering results are more efficient than results produced by DBSCAN [9]. Similar grid based technique is also used by Mahran et al. [24] to generate efficient clustering output from the underlying dataset and it has proved more faster than DBSCAN [9]. The method in [17] was more costly than that of [24] when applied on the large volume of datasets. EI-Sonbaty et al. [15] have modified DBSCAN to develop its enhanced version. The working of [15] is similar to that of [17]. It partitioned the dataset using CLARANS [3].

The enhanced DBSCAN algorithm proposed in [2] takes less time and its memory requirement is also optimal as compare to that of [17] when applied on the large datasets. DCBRD [1] is the clustering technique based on the Radius of the Data. Its most important research strength is that it solved the problem of the user dependency for the supply of the density threshold parameters. YU et al. [25] have combined two different techniques to provide an enhanced version of the DBSCAN. This new DBSCAN modified version obtained from the combination of these techniques is called KNNDBSCAN clustering technique. The density-based algorithms described in [1, 25] are similar in the respect that these can cluster the underlying datasets independently of the users input density thresholds.

Borah et al. [20] have proposed an enhanced DBSCAN technique. This is called DD_DBSCAN. It can cluster the dataset having different shapes, size and differ in local density. Similar to study of [20] GMDBSCAN [21] also modified DBSCAN to deal with the local density, which can cluster the dataset in different regions by corresponding local *MinPts* and use grid-density as approximation of local *MinPts*. Its average runtime complexity is same as that of DBSCAN. Unlike [20], [21] does not discussed computational complexity. YU *et al.* [25] also used the local density in its clustering technique for large datasets. EDBSCAN [19] also focused on the local density variation and provided an enhancement to DBSCAN.

The clustering techniques described in [20, 21] have achieved the efficient clustering result by using the local density in their clustering technique. The density-based techniques discussed in [20, 25] does not need density threshold to be input by the end users. The technique described in [19] requires the user input density threshold manually.

V. CONCLUSION AND FUTURE WORK

In this study, we have presented the summary information of the different enhancement of density-based clustering algorithm called the DBSCAN. The purpose of these variations is to enhance DBSCAN to get the efficient clustering results from the underlying datasets. In addition, we also have highlighted the research contributions and found out some limitations in different research works. Consequently, this work also depicts the critical evaluation in which comparison and contrast have been taken out to show the similarities and differences among different author's works. The spatiality of this work is that it reveals the literature review of different DBSCAN modification and provides a vast amount of information under a single paper. In our future work, we have planned to enhance the DBSCAN and provide its implementation and compare its results with the different existing DBSCAN algorithms variations.

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.

TABLE I. Critical Evaluation of various DBSCAN enhancements

DBSCAN Enhancements	Enhanced Features	Density Threshold *Eps & MinPts*	Computational Cost
Fahim et al. [1]	can perform efficient clustering without requiring any user supplied input parameters	Not Required	$O(nk+ m2k+nm)$ K= no of circles over the data space m=average points in each circle. m=n/k (n= number of points in each circle
Liu et al. [14]	Clustering uneven dataset Efficiently varied in density	Computed automatically	$O (n * \log n)$
EI-Sonbaty et al. [15]	Scalability & Better performance than DBSCAN with speed up faster unto 5 times	User Dependent	Not discussed
Liu [16]	Introduction of Kernel function to make clustering more accurate	User Dependent but less dependency on *Eps*	- Complexity Linear - Much less than that of $O (n * \log n)$
Uncu et al. [17]	Efficient clustering results of the datasets having different densities	Computed automatically	Expensive in terms of Computational
Borah et al. [18]	Memory efficient & I/O cost minimized	Computed automatically (uses only *Eps*)	$O(n * \log n)$
Ram [19]	Limited the amount of allowed local density variation to achieve better results	User Dependent	Not discussed
Borah et al. [20]	α = used to limit the amount of allowed local density variation.	User Dependent	$O (n * \log n)$ n = number of object
Xiaoyun et al. [21]	Reduced Computational Cost & Efficient cluster results for large dataset	User Dependent	Not Discussed
Viswanath et al. [22]	Excellent Clustering results as compare to its DBSCAN & early Variations of it	User Dependent	$O (n + k2)$
Birant et al. [23]	1.Discovering Cluster on spatial-temporal data, 2.Identification of adjacent Clusters. 3.Identification of Noise objects from cluster with different densities	Calculated Automatically	Same as that of DBSCAN technique complexity
Mahran et al. [24]	-Provide high performance with the advantage of high degree of parallelism	Calculated Automatically	N^2 / C C = Total number of cells in each grid
Yu et al. [25]	Offer DBSCAN to determine density threshold in an unsupervised way	Calculated Automatically	$O(n * \log n)$

References

[1] A. M. Fahim, A. M. Salem, F. A. Torkey, and M.A. Ramadan, ”Density Clustering Based on Radius of Data (DCBRD),” World Academy of Science, Engineering and Technology 2006.

[2] L. Kaufman and P. J. Rousseeuw, “Finding Groups in Data. An Introduction to Cluster Analysis,” Wiley, 1990

[3] R. Ng and J. Han, "Efficient and Effective Clustering Method for Spatial Data Mining," Proc. Of the International Conference on Very Large Data Bases, Santiago, Chile, 1994, pp.144-155.

[4] G. Sudipto, R. Rastogi and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases," Proc. Of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, WA, 1998, pp.73-84.

[5] G. Karypis, E. H. Hanand, V. Kumar, "Chameleon: Hierarchical Clustering using Dynamic Modelling," Computer, Aug 1999, vol. 32, pp.68-75.

[6] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," Proc. of the ACMSIGMOD '98 International Conference on Management of Data, Montreal, Canada, 1998, pp.94-105.

[7] C. H. Cheng, A. W. Fu, and Y. Zhang, "Entropy-Based Subspace Clustering for Mining Numerical Data," Proc. of the 5th International Conference on Knowledge Discovery and Data Mining, San Diego, CA, 1999, pp.84-93.

[8] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wave Cluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases," Proc. of the 24th International Conference on Very Large Databases, San Francisco, CA, 1998, pp.428-439.

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.

[9] M. Ester, H. P. Krigel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial DatabaseswithNoise,"Proc.ofthe2ndInternationalConferenceonKnowledgeDiscoveryandDataMining, Portland, WA, 1996, pp. 226-231.

[10] A. Hinneburg and D. A. Keim, "An Efficient Approach to Clustering in Multimedia Databases with Noise," Proc. of the 4th International Conference on Knowledge Discovery and Data Mining, New York, 1998, pp.58-65.

[11] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: Ordering Points to Identify the Clustering Structure," Proc. of the ACMSIGMOD'99 International Conference on Management of Data, Philadelphia, PA, 1999, pp. 49-60.

[12] D. H. Fisher, "Knowledge Acquisition via Incremental Conceptual Clustering," Machine Learning, vol. 2, pp. 139-172, 1987.

[13] T. Kohonen, "Self-Organization and Associative Memory," New York, NY, Springer-Verlag, 1988.

[14] P. Liu, D. Zhou, and N. J. Wu,"*VDBSCAN*: Varied Density Based Spatial Clustering of Applications with Noise," in proceedings of IEEE International Conference on Service Systems and Service Management, Chengdu, China, pp 1-4, 2007.

[15] Yasser El-Sonbaty, M. A. Ismail, and Mohamed Farouk, "An Efficient Density Based Clustering Algorithm for Large Databases," in proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004).

[16] B. Liu, "A Fast Density-Based Clustering Algorithm For Large Databases," in proceedings of the fifth International Conference on Machine Learning and Cybernetics, Dalian,13-16 August 2006.

[17] O. Uncu, W. A. Gruver, D. B. Kotak, D. Sabaz, Z. Alibhai, and C. Ng, "*GRIDBSCAN*: GRId Density-Based Spatial Clustering of Applications with Noise," 2006 IEEE International Conference on Systems, Man, and Cybernetics October 8-11, 2006, Taipei, Taiwan.

[18] B. Borah and D. K. Bhattacharyya, "An Improved Sampling-Based *DBSCAN* for Large Spatial Databases," presented in the international Conference on Intelligent Sensing and Information Processing, Chennai, India, January 2004.

[19] A. Ram, A. Sharma, A. S. Jalall, R. Singh, and A. Agrawal, "An Enhanced Density Based Spatial Clustering of Applications with Noise," 2009 IEEE International Advance Computing Conference (IACC2009) Patiala, India, 6-7 March 2009.

[20] B. Borah and D. K. Bhaftacharyya, "A Clustering Technique using Density Difference," IEEE - ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb. 22-24, 2007. pp. 585-588.

[21] X. Y. Chen, Y. F. Min, Y. Zhao, and P. Wang, "GM*DBSCAN*: Multi-Density *DBSCAN* Cluster Based on Grid," IEEE International Conference on e-Business Engineering (ICEBE 2008).

[22] P. Viswanath, and V. S. Babu, "l-*DBSCAN*: A Fast Hybrid Density Based Clustering Method," Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006.

[23] D. Birant and A. Kut, "ST-*DBSCAN*: An algorithm for clustering spatial–temporal data," Data and Knowledge Engineering, Volume 60, Issue 1(January 2007), pp. 208-221, Year of Publication: 2007, ISSN: 0169-023X.

[24] S. Mahran and K. Mahar, "Using Grid for Accelerating Density-Based Clustering," Computer and Information Technology, CIT2008, 8th IEEE International Conference on. 08/08/2008, ISBN: 978-1-4244-2357-6, Sydney, NSW.

[25] X. P. Yu, D. Zhou, and Y. Zhou, "A New Clustering Algorithm Based on Distance and Density," presented in proceedings of International Conference on Services Systems and Services Management (ICSSSM-2005), Vol. 2.

[26] M. T. H. Elbatta and W. M. Ashour, "A Dynamic Method for Discovering Density Varied Clusters", Published in International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 6, No. 1, February, 2013

[27] A. Ram, S. Jalal, A. S. Jalal and M. Kumar, "DVBSCAN: A Density based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases", International Journal of Computer Applications (0975–8887), vol. 3, no. 6, (2010) June.

[28] M. N. Gaonkar and K. Sawant, "Auto Eps DBSCAN: DBSCAN with Eps Automatic for Large Dataset", Published in International Journal on Advanced Computer Theory and Engineering (IJACTE), ISSN (Print) : pp. 2319 – 2526, Volume-2, Issue-2, 2013.

Authorized licensed use limited to: UNIST. Downloaded on May 12,2024 at 01:37:24 UTC from IEEE Xplore. Restrictions apply.