International Journal of Database Theory and Application
Vol.7, No.5 (2014), pp.17-28
http://dx.doi.org/10.14257/ijdta.2014.7.5.02

Critical Analysis of Density-based Spatial Clustering of Applications with Noise (DBSCAN) Techniques

Said Akbar and M.N.A. Khan

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan
E-mail: syedakbar_cs@yahoo.com, mnak2010@gmail.com

Abstract

Clustering is the most used technique in data mining. Clustering maximize the intra-cluster similarity and minimize the inter clusters similarity. DBSCAN is the basic density based clustering algorithm. Cluster is defined as regions of high density are separated from regions that are less dense. DBSCAN algorithm can discover clusters of arbitrary shapes and size in large spatial databases. Beside its popularity, DBSCAN has drawbacks that its worst time complexity reaches to O (n2). Similarly, it cannot deal with varied densities. It is hard to know the initial value of input parameters. In this study, we have studied and discussed some significant enhancement of DBSCAN algorithm to tackle with these problems. We analysed all the enhancements to computational time and output to the original DBSCAN. Majority of variations adopted hybrid techniques and use partitioning to overcome the limitations of DBSCAN algorithm. Some of which performs better and some have their own usefulness and characteristics.

Keywords: *Data Mining, Spatial databases, Clustering, DBSCAN*

1. Introduction

Data mining is the branch of KDD (Knowledge Discovery in Databases) that analyse and extract meaningful sequence of knowledge from huge databases. Knowledge discovery got importance with the advancement of Internet technologies especially receiving huge amount of spatial data from different sources. When data mining works in finding meaningful pattern in huge spatial datasets, it is known as spatial data mining. Discovering patterns in spatial dataset is normally more complex task than traditional datasets. With the increased rise of spatial data improved demands for data mining techniques have risen [6]. Among the data mining techniques, Clustering performs the most important role. Clustering is the process of combining homogeneous objects, which can either be the form of groups as well as can be abstract or physical objects. Clustering is used in many fields of studies like botany, zoology, medical science, business study and e-commerce. Clustering is very useful in categorizing living things such as plants and animals, fraud detection, pattern recognition, digital image processing and exploring the web documents etc. A high standard clustering technique can discover clusters of different shapes of any size in large datasets in one scan, thus having lesser time complexity. Clustering is divided into hierarchical, partitioning, density and grid based [1].

Density based techniques are based on separating regions of high density from that of low dense regions. Such techniques can detect clusters of arbitrary shapes easily. Such methods use the concept of neighbourhood to mine data sets efficiently. A region is

ISSN: 2005-4270 IJDTA

treated as a cluster which has more objects from the given value while low dense regions are treated as noise [13].

DBSCAN is the most popular and widely used density based clustering algorithm that can detect clusters of arbitrary shapes and size in large spatial databases. DBSCAN discovers clusters of denser regions and regions with low density are marked as noise or outliers. The basic idea behind such algorithm for finding clusters is that for every point in the cluster the neighbour points of a specified radius Eps will be consist of the lowest number of points (MinPts). There are two parameters required for DBSCAN algorithm: one is ε (Eps) and the second is MinPts, which is the lowest number of points needed to form a cluster. DBSCAN randomly selects an object and examine it only once. The neighbourhood of the object is examined in such a way that if it meets the minimum criteria for creating a cluster, a cluster is created, to which objects can be added later, and if the neighbourhood objects do not satisfy the lowest threshold criterion then it is declared as a noisy object [7]. Many clustering techniques cannot detect clusters of arbitrary shapes. DBSCAN has the advantage that it can detect a cluster of arbitrary shapes. The disadvantages of DBSCAN includes that the worst case complexity tends to $O(n^2)$. It is limited in dealing with different densities and requires more memory space to load the entire database. To overcome these limitations, many enhancements have been developed to DBSCAN algorithm. In most of the enhancement generic techniques are used to remove the problems of time complexity and varied densities.

2. Literature Review

According to [5], data mining plays a vital role due to its popularity and increased usability. Clustering is a data mining technique which also accounts for density based algorithms. It groups a set of similar data in the matching cluster and unalike data id grouped in other clusters, and these are the two subclasses of density based clustering algorithms. Many efficient clustering algorithms have been developed but DBSCAN (Density Based Spatial Clustering of Application with Noise) is the most widely used algorithm because of generating a good standard noiseless cluster with arbitrary shapes. DBSCAN finds clusters based on objects density. While making clusters, DBSCAN algorithms check for each item in the database. DBSCAN first calculates the quantity of objects in the neighbour region called Epsilon (Eps). When the number of neighbour objects is less than the given threshold, the objects are marked as Noise and if they are more than the given threshold value, the objects are ticked as clusters. These clusters are generated by the core objects which combines the density reachable set of attached objects. DBSCAN algorithm has a limitation that it cannot mine clusters with different densities. Many algorithms have been proposed to tackle with various densities in the database. Some of which are responsive to input parameters to know the difference of densities. Hence, a small change in parameters produces irrelevant result. DBSCAN algorithm has many variants like OPTICS, DENCLUE, CHAMELEON, DDSC and EDBSCAN.

The proposed algorithm has been introduced to overcome the limitation of original DBSCAN that can handle clusters with various densities, and is known as HDBSCAN (Heterogeneous DBSCAN). To tackle with different densities of clusters, the algorithm at first receives the neighbour objects by calling the LongRegionQuery function (InnerRegionObjects and OuterRegionObjects). The prevailing ground of objects density performs the counting of InnerRegionObjects. Performance enhancement has been done by LongRegionQuery and ShortRegionQuery in the new algorithm. Thus the

ShortRegionQuery takes the items which are returned from the array of LongRegionQuery function. It will also sort the entire database in the successive loop. The magnitude of Eps is not vulnerable, hence the improvement is assured. All the InnerRegionObjects which are less and equal to Eps will be operated in the successive loop using the ShortRegionQuery function when the call for LongRegionQuery is compiled. The ID of new cluster is produced when the first core object is acquired. A new cluster ID is created from all the InnerRegionObjects e.g., unsupervised, anomaly, and similarity core objects density range. The only drawback of the proposed solution is that the responsive value of Eps degrades performance even if the memory effect technique is applied.

In [2], Spatial-Temporal Density Based Clustering (ST-DBSCAN) is the improved version of DBSCAN algorithm. The changes which have been done in DBSCAN include, the new ST-DBSCAN can detect clusters from spatial, non-spatial, and temporal values. Secondly, the algorithm can detect input from the noise data in different density. Finally, it solves the problem of border points. As it processes spatial data, therefore it is very vast and has broad applications such as GIS, weather forecasting and medical imaging. Varied Density Based Spatial Clustering of Application with Noise (VDBSCAN) is proposed to deal with large variation in densities. In VDBSCAN the input parameter values are set by default. DBSCAN and VDBSCAN both offer the same level of performance. Vibration Method DBSCAN (VMDBSCAN) finds all clusters with various structures and sizes also with varied densities. VMDBSCAN has more performance than DBSCAN but it has low performance in time complexity than DBSCAN. Instead of selecting one point, Dynamic Method DBSCAN (DMDBSCAN) algorithm choose many points simultaneously for various densities and the upcoming points for making clusters are neglected. Then it checks for both scanty and dense regions. This algorithm detects appropriate Epsi for thick region.

The paper [3] provided a detailed overview of enhanced clustering algorithms. The author surveyed different clustering algorithms on the basis of density e.g., DBSCAN, DVBSCAN, ST-DBSCAN, VMDBSCAN, and DMDBSCAN. VMDBSCAN provides more efficiency while making clusters of various sizes and structures. DMDBSCAN defeats the limitation of local value. The overall extended versions of DBSCAN are superior in performance while discovering clusters of varied densities and removing outliers.

In paper [4], the enhanced DBSCAN (EDBSCAN) is proposed to get rid of difference in density problem. The working mechanism of EDBSCAN is that it calculates the difference of core point and also calculates the densities of all neighbour points Ɛ. If the difference in density of a core point to its adjacent point is smaller than the fixed value δ, then only core point is authorized for further processing otherwise the point is simply put in same cluster. The values of δ can be described by the user and is helpful in finding clusters of varied densities in scanty and those which has significant difference in density. The variation in density of core point becomes larger than the threshold value of δ, when the core point has both scanty and populated region in the Ɛ -neighbourhood.

According to paper, [5] CLARANS uses DBSCAN algorithm to identify clusters in huge datasets. First CLARANS is used to divide the database then it use DBSCAN algorithm. It divides the database into many small divisions. In this way, the database is divided into small pieces and then scanned by the DBSCAN, which has a great effect in

improving the complexity. CLARANS does not compute the region which is denser but it works with the idea of comparing gaps among the border objects. The DBSCAN defines a fixed value, CLARANS compares connections between data if the objects have less connectivity among the data objects from the fixed value, it removes the noise first and then forms a cluster.

L-DBSCAN is a high speed hybrid clustering technique. Two types of models are used by l-DBSCAN: rough level and finer level. The rough level is used to minimize time demand while the finer level is used to provide precise results. Hybrid clustering is the process of choosing an appropriate model from huge databases, only the fetched models are then applied by the clustering techniques. The leaders approach keeps a group of leaders L. A number of clusters are formed by leaders as results which are then further divided for merging into clusters. Two variables are used by l-DBSCAN, Tf for Fine leader in the border area and Tc coarse leader lies in the centre of the circle. DBSCAN and l-DBSCAN both provides the same output when the value of Tf = 0. It is difficult for a user to provide more variables due to the understanding of that area.

Parzen-Window DBSCAN for huge databases is the enhancement of L-DBSCAN. In this algorithm the number of parameters is increased in a model. It also uses leader approach. A special function is used for measuring density. Parzen-Window technique introduced hybrid Clustering is same as DBSCAN but it uses L* as an alternative of D. The only distinction is that it uses a function known as smooth kernel at leader to find the density. The experimental evaluation shows that it is have better complexity that DBSCAN but with increased parameters its efficiency is like DBSCAN. LSH (Local Sensitive Hashing) Based DBSCAN look for the most closest points and makes LSH list first and then use DBSCAN on the LSH list to form clusters. This technique also has the problem of increased input patterns, finding the values of input variables are quite difficult. Grid Based DBSCAN splits up the database into small slots and then use DBSCAN algorithm to form clusters. All the cells are scanned simultaneously to minimize the execution time. A core point is used for making clusters. DDSC is comparatively better than OPTICS because it can deal with different densities very efficiently. It has the problem of that it can't minimize the compiling time. Motion Determination Using Non-Uniform Sampling Based Clustering differentiates crowded regions from low dense regions using two thresholds density and use DBSCAN for clusters forming. C-DBSCAN (DBSCAN with Constraints) reduces time complexity and blocks the forming of vacant clusters. Constraints applied clustering is also called semi-supervised. C-DBSCAN divides the data set into partitions and constraints are applied on each stage.

All the algorithms are basically the extensions of DBSCAN. By comparing them, all the algorithms are based on overcoming two main problems of DBSCAN; complexity and varied densities. Two approaches are adopted by the researchers; one is partitioning the dataset and second is hybrid techniques. Comparing the result with DBSCAN, the performance of all algorithms were either same as the DBSCAN or were better. Majority of the approaches are experimented on synthetic databases not on real datasets.

In paper [6] proposed a technique to overcome the problem of varied density of DBSCAN that works on databases from small to large proportions. Different steps take place in the proposed system. First necessary step is data pre-processing. Pre-processing deals with unknown values and removes the noise. In the second step, the data set is divided into four partitions in order to find non-overlapped clusters. Starting

from the top level and dividing the data set until the similarity goal is not encountered. It divides the dataset and sub regions to find the centre object. Then it computes the Euclidian distance among the main and sub regions. If the acquired distance is more than threshold value, regions is divvied otherwise regions will remain same. The above process is repeated until the criterion is met. The value of threshold is set average in order to get accurate divisions of regions. Thirdly, applying DBSCAN, two types of distances are used; Euclidian and Manhattan distances. For too small clusters, post-processing is required to make accurate clusters and remove noise. It also checks if clusters are surrounded by another cluster. The final step merges noise to the closest cluster in the neighbourhood forcefully. The proposed method creates more accurate and more clusters when compared to the existing system. The author concluded that the database is divided into partitions on the basis of threshold value before creating clusters. A synthetic database is evaluated in bottom-up approach. Euclidian distance performs better with small datasets but is not responsive to large dataset. It also creates more clusters but produces more incorrect cluster as compared to Manhattan. Manhattan distance takes less time to produce a cluster than Euclidian but Manhattan is responsive to noise as contrast of Euclidian.

In paper [6], the authors surveyed five different density based clustering algorithms in detail *e.g.*, DBSCAN, DVBSCAN, VDBSCAN, DBCLASD and ST-DBSCAN on the basis of mandatory demands needed for a clustering technique in spatial data. They compared different aspects and features of the algorithms like the nature of data; structure of clusters and about the parameters, each and every algorithm has different features and has its own unique characteristics.

In [7], Incremental DBSCAN is used to find clusters in dynamic fashion. Dynamic databases always get updated. When the data warehouse is updated, the clusters need to be updated using DBSCAN. Incremental clustering efficiently identifies the global optimum. In the modified dataset, the idea of DBSCAN is used to form clusters. The proposed technique first sorts the core object to calculate its mean points in the cluster and the upcoming data. This new data is put into a particular cluster fulfilling the mean distance criterion. If the new data does not fulfill the requirement for making clusters, it is handled as noise or outliers. The experimental results show that the proposed solution is better than existing one.

In paper [8], varied DBSCAN is a new technique to overcome the limitations of DBSCAN. VDBSCAN at first computes and saves k-dist for every operation and divides the dataset into k-dist plots. K-dist plot specifies all density variations initiatively. For every density then it selects the parameters Epsi by default. Then sort the entire database and form clusters of various densities on the basis of matching Epsi. There are two steps in VDBSCAN. First select parameter Epsi. This is the basic step to process a database. The algorithms draws k-dist plot in order to find both the parameter Epsi and separate different levels of density in the database. If the graph has “n” different curves then there are “n” levels of density. If the density does not differ then there is only a single density. In the second step, it deals with different densities using DBSCAN method. It applies DBSCAN technique to each Epsi. Epsi are arranged ascending order. Only marked objects can be processed and objects those are not marked are considered as outliers. Clusters *Ci-t* is showed as final clusters. New parameters like α and λ are applied to reduce difference in the local density inside the cluster.

In [9], the authors proposed Parallel DBSCAN (PDSDBSCAN) on the basis of disjoint set data structure. The PDSDBSCAN algorithm is compiled to calculate the number of local objects and also calculate the number of local clusters in parallel manner without communicating. Thus, final clusters are acquired by merging local clusters while the master-slave approach uses merging in sequential manner. During merging of two trees, only root pointer is interchanged in PDSDBSCAN while in master-slave strategy the whole tree is traversed for renaming. Due to the parallel calculations, speedup and efficiency is obtained. PDSDBSCAN algorithm has the problem of building kd-free which is not parallelized and takes more time than DBSCAN. This step is taken before running DBSCAN algorithm. Another drawback is related to space issues.

In [10], the authors proposed a new version of DBSCAN algorithm that can overcome the problem of border objects. The revised DBSCAN can deal with the clusters with dense data that are enclosed in other clusters. The parameters Eps and MinPts are used to describe data density. Data objects are identified as border objects to generate clusters precisely. The improvement of the algorithm is the result of using core-density-reachable concept, which uses core-density-reachable chains as a replacement of density-reachable chains. Therefore, they retain similar number and same core points for every cluster are discovered. Border objects are not processed until the identification of all the clusters. Finally core point is allocated to the best fit density reachable chain and the border point is allocated to the cluster containing core-density-reachable chain.

In revised DBSCAN, setting the border point placed between close clusters is random, which results in non-robust clustering. This issue becomes vulnerable with the increased size of neighbourhood leads to identify several border points. The revised DBSCAN technique can identify the same number of clusters as the original DBSCAN algorithm. The proposed algorithm performs better and assigns border objects to the related cluster efficiently. It means that the algorithm solved the problem of border objects. The revised DBSCAN shows that it does not follow the order in which clusters are detected.

The paper [11] discusses privacy of distributed DBSCAN. To achieve privacy two parties *i.e.*, Alice and Bob need a trusted third party to share their clusters information to perform necessary computations. For this purpose Alice and Bob should know cluster number. However it is difficult to get trusted third party in real. In this paper, the authors provide protocol to carry out the function needed by two parties to share data with each other. In the calculation of multiple parties, a malicious opponent changes the input, which is risky. This attack cannot be handled but the risk can be prevented to stop the malicious party from execution. This adversary is called semi honest and goes after protocol specification. The privacy of protocol depends on the input output calculated by a party using protocol. Scalar products have a special type of multiplication problem; this problem cannot be dealt by popular scalar products protocol. The authors proposed Paillier's additive homomorphic technique and multiplication protocol to resolve the multiplication problem.

In paper [11], the proposed algorithms are totally private and the arbitrary model make it able to the practically horizontally and vertically partition of data. The algorithm produces zero knowledge of clusters in the neighbourhood of points from the other party, which decrease the confidence of revealed information of the accurate points.

In [12], the authors proposed a new improvement to DBSCAN algorithm called MR-DBSCAN. The main focus of the paper is to implement and sketch the MR-DBSCAN, an effective technique that is using MapReduce, which is a programming procedure for high intensive data set. The suggested technique first takes care of the data sets that are sensitive to the overall performance. The technique also focuses on the density of the point in the large dataset. The technique MR-DBSCAN which is suggested in the paper operates on three main stages. It first divides the data into partition. Second it logically organizes it. Third it merges the whole globally to achieve the desired results. To make clusters logically it is the most important part as far as the time required for computation is concerned. To make the partitions the suggested techniques takes advantage of binary space partitioning. The technique uses the specified rules which are defined earlier to make the partitions. The basic rules can be such that divide the point as evenly if possible. Or the other rule may be such that minimize the number of points inside the boundary. On the other hand these techniques may not be as efficient as required when take into account to balance the load. The technique mainly focuses on the parallel aspects of the DBSCAN algorithm. However, it can also be used for other purposes depend on the available data and keep in consideration its application in different fields.

In paper [13], the authors proposed a new clustering algorithm called DBSCAN-GM in order to remove the shortcomings of DBSCAN. The algorithm is carried out in several steps as described below: In the first step, DBSCAN-GM needs the database as input. It divides the database into "k" clusters; as a result a number of sub datasets are produced and finds the centre of each dataset. In second step, DBSCAN-GM determines the value of important parameter Eps. Smaller value of Eps marks objects as noise and cannot be sorted for further processing. All sparse clusters with similar objects are divided into many clusters. A user should be careful to set the value of Eps. In next step, local MinPts are calculated in two dimensional data space. Finally, DBSCAN algorithm is applied on the input parameters after calculating the value of both the parameters Eps and MinPts. The value of parameters comes to be automatic because it discovers noiseless standard clusters and partitions dataset.

By comparing different clustering techniques, G-means provides better results as it can tackle with noise points and can produce appropriate automatic value for parameters. DBSCAN-GM has more time complexity than DBSCAN; hence, it has the limitation of time complexity. In [14], the authors proposed a technique G-DBSCAN constitutes the data in the form of graph G (V, E) where V shows the objects of clusters and E shows edges which connects objects within the radius. The algorithm takes the dataset as input parameters R and MinPts and generate graph. Euclidean distance function is used to compute the distance of other objects. In the second step, if the number of neighbours is more than MinPts the object is categorized as core object otherwise noise. In the next part of the algorithm clusters are discovered. Breadth First Search (BFS) is took place from core object and examines all objects reachable from the core object. The objects are placed in the same clusters as members and also marked as border objects. The search is continuing until all the objects are visited iteratively and until it becomes it. Thus the proposed algorithm contains two steps: constructing the graph and discovering the clusters. To decrease the time complexity both the stages are parallelized.

The experimental evaluation of the proposed algorithm shows a speedup up to 112X, which is very appreciable result. However the time complexity of the proposed

technique remains same as DBSCAN. The algorithm has a major drawback of memory. In future, the authors suggested that G-DBSCAN is applicable in real applications and scenarios for large data sets in multiple dimensions.

In [15], the proposed solution called P-DBSCAN randomly selects a photo that is neither part of a cluster nor noise. A photo can be part of the cluster if it is core image otherwise noise is considered and all the adjacent photos are pushed in queue for processing additionally. This process remains continue until the queue is empty. Similarly another image is picked and put it in the same cluster. The neighbours of the photos and adaptive density ID examined; if the photo is core, adaptive density is used otherwise the image is placed in the queue. A number of conditions are examined when adaptive density is running.

Iqbal *et al.*, [16,17] proposed performance metrics for software design and software project management. Process improvement methodologies are elaborated in [18,19] and Khan *et al.*, [20] carried out quality assurance assessment. Amir *et al.*, [21] discussed agile software development processes. Sarwar *et al.*, [22] and Khan *et al.*, [23] analyzed issues pertaining to requirement engineering processes. Umar and Khan [24, 25] analyzed non-functional requirements for software maintainability. Khan *et al.*, [26, 27] proposed a machine learning approaches for post-event timeline reconstruction. Khan [28] suggests that Bayesian techniques are more promising than other conventional machine learning techniques for timeline reconstruction. Rafique and Khan [29] explored various methods, practices and tools being used for static and live digital forensics. In [30], Bashir and Khan discuss triaging methodologies being used for live digital forensic analysis.

3. Critical Analysis

In this section, we provide a critical analysis of the DBSCAN technique.

Table I. Critical Review of the DBSCAN Techniques

Ref.	Technique Used	Focus Area	Pros	Cons
[1]	Heterogeneous DBSCAN	DBSCAN is not enough intelligent to mine clusters with different densities. The authors focused on dealing with varied densities	HDBSCAN is Simple, fast and can deal with different densities efficiently	The technique discussed is efficient and produces better results. However sensitive value of Eps degrades performance.
[2]	DBSCAN, DVBSCAN, ST-DBSCAN, VMDBSCAN AND DMDBSCAN	Enhanced DBSCAN algorithms	Better performance as compared to DBSCAN while discovering clusters of varied densities and noise removal	Most of the algorithms can't are sensitive to find the value of parameters automatically.
[3]	EDBSCAN	Finding clusters of varied densities	Create clusters with uniform regions. It can deal with the	Number of cluster depends on the value of δ. Another issue is

			variation in local density using additional parameters	still it can't handle varied densities in some way.
[4]	Indexing Scheme	Computational time and difference densities	Reduced time complexity and can find clusters of different densities	Inefficient in handling multiple dimensionalities.
[5]	Euclidean and manhattan distance	High dimensional data sets handling	Cluster exists within a cluster detection and performs better from low to high dimensions	Incorrect clustering and high time complexity in Euclidean and high ratio of noise in manhattan. Cannot determine input parameters
[6]	DBSCAN, VDBSCAN, DVBSCAN, ST-DBSCAN and DBCLASD	Input parameter, type and density of the data and shape of the cluster	Deals Large spatial databases, arbitrary shapes clusters discovering and find varied densities clusters	Computationally expensive and automatic value of parameter limitation
[7]	Incremental DBSCAN	Dynamic clustering in databases	Less time and effort is required and suitable for large multidimensional dynamic database	
[8]	VDBSCAN	Variation in density	Limits local density variations	Technique discussed is better in terms of efficiency and applicability. However its time complexity is high and doesn't find automatic value for parameters
[9]	PDSDBSCAN based on Disjoint-set data structure	Parallelization of DBSCAN algorithm	More scalable and concurrent in performance than DBSCAN	K-d tree is still sequential and is computationally expensive. It also has memory limitation
[10]	Core-density-reachable chains	Issue of border objects removal	Generate quality clusters due to resolving the issue of border objects	The technique outperforms but doesn't follow the order in which clusters are detected.

[11]	Privacy preserving algorithms	Clustering over vertically, horizontally and arbitrary data	The algorithm provides security and can partition data horizontally and vertically	Lack of confidence due to zero knowledge of clusters in the neighbourhood of points from the other party.
[12]	Map Reduce	Heavily skewed data	Ensure scalability and load is balanced through parallel DBSCAN	Poor performance in high dimensionas.
[13]	Gaussian Mean	Automatic computing of parameters Eps and MinPts	First, Advance declaration of parameter. Second, dealing with noise to produce arbitrary shaped clusters. Third, generate quality clusters.	Time consumption problems
[14]	G-DBSCAN	Parallelization of DBSCAN	The technique used is very fast	The technique shows high speed up. However it has a major memory drawback

4. Conclusion and Future Work

With the increased demand of data mining techniques, especially clustering techniques are becoming very popular. DBSCAN is a density based clustering technique, which is famous due to its high usability and efficiency. Besides its advantages that it can identify clusters of arbitrary shapes and size in large spatial databases, it also has disadvantages like; DBSCAN has high computational time in its worst case O (n2). It is also difficult for a user to determine the values of input parameters. Finally, DBSCAN has a major drawback that it is not enough intelligent to identify clusters of varied densities. To overcome these limitations, a number of techniques have studied in this survey like HDBSCAN, EDBSCAN, Improved DBSCAN, Incremental DBSCAN, MR-DBSCAN, DBSCAN-GM, P-DBSCAN and G-DBSCAN etc. Each and every algorithm has its own uniqueness and characteristics ranging in different applications. However each variation has still drawbacks. In the future, we will work on one these variations to improve DBSCAN algorithm.

References

[1] J. H. Peter and A. Antonysamy, "Heterogeneous Density Based Spatial Clustering of Application with Noise," International Journal of Computer Science and Network Security, vol. 10, no. 8, **(2010)**, pp. 210-214.
[2] T. Verma and D. Gaur, "A Survey on Study of Enhanced DBSCAN Algorithm," In International Journal of Engineering Research and Technology, vol. 2, no. 11, **(2013)**.
[3] A. Ram, A. Sharma, A.S. Jalal, A. Agrawal, and R. Singh, "An enhanced density based spatial clustering of applications with noise", In Advance Computing Conference, IACC 2009. IEEE Internationa, **(2009)**, pp. 1475-1478.
[4] T. Ali, S. Asghar, and N. A. Sajid, "Critical analysis of DBSCAN variations," In Information and Emerging Technologies (ICIET), 2010 International Conference, **(2010)**, pp. 1-6.

[5] G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," InEngineering (NUiCONE), 2012 Nirma University International Conference, **(2012)**, pp. 1-6.
[6] M. Parimala, D. Lopez, and N.C. Senthilkumar, "A survey on density based clustering algorithms for mining large spatial databases", International Journal of Advanced Science and Technology, vol. 31, no.1, **(2011)**.
[7] S. Chakraborty, and N. K. Nagwani, "Analysis and study of incremental DBSCAN clustering algorithm", Int. J. Enterprise Comput. Bus. Syst, **(2011)**.
[8] K. G. Swathi, and K. N. V. S. S. K. Rajesh, "Comparative analysis of clustering of spatial databases with various DBSCAN Algorithms," IJRCCT, vol.1, no.6, **(2012)**, pp. 340-344.
[9] M. Patwary, M. Ali, D. Palsetia, A. Agrawal, W. K. Liao, F. Manne, and A. Choudhary, " A new scalable parallel dbscan algorithm using the disjoint-set data structure", In High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference, , **(2012)**, pp. 1-11.
[10] 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)**.
[11] J. Liu, J. Z. Huang, J. Luo, and L. Xiong, "Privacy preserving distributed dbscan clustering," In Proceedings of the 2012 Joint EDBT/ICDT Workshops, **(2012)**, pp. 177-185.
[12] Y. He, H. Tan, W. Luo, S. Feng, and J. Fan, "MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data", Frontiers of Computer Science, vol. 8, no.1, **(2014)**, pp. 83-99.
[13] A. Smiti, and Z. Elouedi, "DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques," In Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference, pp. 573-578.
[14] G. Andrade, G. Ramos, D. Madeira, R. Sachetto, R. Ferreira, and L. Rocha, "G-DBSCAN: A GPU Accelerated Algorithm for Density-based Clustering," Procedia Computer Science, vol. 18, **(2013)**, pp. 369-378.
[15] S. Kisilevich, F. Mansmann, and D. Keim, "P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos", Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, **(2010)**, pp. 38.
[16] Iqbal S., Khalid M., Khan, M N A. A Distinctive Suite of Performance Metrics for Software Design. International Journal of Software Engineering & Its Applications, vol. 7, no. 5, **(2013)**.
[17] Iqbal S., Khan M.N.A., "Yet another Set of Requirement Metrics for Software Projects", International Journal of Software Engineering & Its Applications, vol. 6, no. 1, **(2012)**.
[18] Faizan M., Ulhaq S., Khan M N A., Defect Prevention and Process Improvement Methodology for Outsourced Software Projects. Middle-East Journal of Scientific Research, vol. 19, no. 5, **(2014)**, pp. 674-682.
[19] Faizan M., Khan M NA., Ulhaq S., Contemporary Trends in Defect Prevention: A Survey Report. International Journal of Modern Education & Computer Science, vol. 4, no. 3, **(2012)**.
[20] Khan K., Khan A., Aamir M., Khan M N A., "Quality Assurance Assessment in Global Software Development", World Applied Sciences Journal, vol. 24, no. 11, **(2013)**.
[21] Amir M., Khan K., Khan A., Khan M N A., "An Appraisal of Agile Software Development Process", International Journal of Advanced Science & Technology, vol. 58, **(2013)**.
[22] Sarwar, A., & Khan, M. N., "A Review of Trust Aspects in Cloud Computing Security", International Journal of Cloud Computing and Services Science (IJ-CLOSER), vol. 2, no. 2, **(2013)**, 116-122.
[23] MNA. Khan, M. Khalid and S. ulHaq, "Review of Requirements Management Issues in Software Development", International Journal of Modern Education & Computer Science, vol. 5, no. 1, **(2013)**.
[24] M., Umar and M N A. Khan, "A Framework to Separate Non-Functional Requirements for System Maintainability", Kuwait Journal of Science & Engineering, vol 39(1 B) ,**(2012)**, pp. 211-231.
[25] M. Umar, M. N. A. Khan, "Analyzing Non-Functional Requirements (NFRs) for software development", IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), **(2011)**, pp. 675-678.
[26] M. N. A. Khan, C. R. Chatwin, & R. C. Young, "A framework for post-event timeline reconstruction using neural networks. digital investigation", vol. 4, no. 3, **(2007)**, pp. 146-157.
[27] M. N. A. Khan, C. R. Chatwin, & R. C. Young, "Extracting Evidence from Filesystem Activity using Bayesian Networks", International journal of Forensic computer science, vol. 1, **(2007)**, pp. 50-63.
[28] M. N. A. Khan, "Performance analysis of Bayesian networks and neural networks in classification of file system activities", Computers & Security, vol. 31, no. 4, **(2012)**, pp. 391-401.
[29] M. Rafique & M. N. A. Khan, "Exploring Static and Live Digital Forensics: Methods, Practices and Tools", International Journal of Scientific & Engineering Research vol. 4, no. 10, **(2013)**, pp. 1048-1056.
[30] M. S. Bashir & M. N. A. Khan, "Triage in Live Digital Forensic Analysis", International journal of Forensic Computer Science, vol. 1, **(2013)**, pp. 35-44.

