Critical Analysis of DBSCAN Variations

Tariq Ali, Sohail Asghar, Naseer Ahmed Sajid

Department of Computer Science

Muhammad Ali Jinnah University

Islamabad Pakistan

tariqali.1982@gmail.com

sohail.asghar@jinnah.edu.pk

nasajid2005@gmail.com

Abstract— **DBSCAN is a widely used technique for clustering in spatial databases. DBSCAN needs less knowledge of input parameters. Major advantage of DBSCAN is to identify arbitrary shape objects and removal of noise during the clustering process. Beside its familiarity, DBSCAN has problems with handling large databases and in worst case its complexity reaches to $O(n^2)$. Similarly, DBSCAN cannot produce correct result on varied densities. Some variations are proposed to DBSCAN, to show its working in some other domains. In this paper we surveyed some important techniques in which original DBSCAN is modified or enhanced with improvement in complexity or result improvement on varied densities. We define criteria and analyse these variations with complexity (time and space) and output to the original DBSCAN algorithm. We also compare these variations with one another to select the efficient algorithm. In most of the variations partitioning and hybrid methodologies are originated to deal DBSCAN problems. We concluded with some variations which perform better than other variation over defined criteria (objectives).**

Keywords— **DBSCAN, Clustering, Data Mining**

I. INTRODUCTION

Data mining is the process of identifying hidden and interesting patterns from large data set, which can further be used in decision making and future prediction [27]. Clustering is an important technique of class identification in spatial databases. Objective of the clustering is to maximize the intra cluster similarity and minimizing the inter cluster similarity. Clustering is used to find useful patterns in unlabeled data.

DBSCAN is an important and widely used technique for class identification in spatial databases [4]. Many variations to the DBSCAN exist. DDBSCAN was first proposed in [14], for clustering over large datasets. Two major draw backs are seen with DBSCAN, one is the time complexity which reaches to O(n2) in worst case and another is clustering accuracy over the varied density [15]. In this paper we survey the shortcomings of DBSCAN, and techniques proposed in the literature to overcome these limitations. The new techniques are evaluated with certain parameters (objectives). In most of the variations some generic methodologies are adopted to improve DBSCAN in terms of time complexity and output correctness. Conclusion and some future work are identified at the end of this survey to improve DBSCAN.

Rest of the paper is organized as; Section 2 discusses DBSCAN and its main limitations. Section 3 provides a survey of the variations to the DBSCAN; Section 4 discusses the criterion for analysis of all the variations. Section 5 elaborates the analysis of all the variation on the defined criterion. Section 6 provides the conclusion and future work.

II. DBSCAN

DBSCAN (Density based spatial clustering of application with noise) [14] is density based method which can identify arbitrary shaped clusters where clusters are defined as dense regions separated by low dense regions. DBSCAN starts with an arbitrary object in the dataset and checks neighbor objects within a given radius (*Eps*). If the neighbours with in that *Eps* are more than the minimum number of objects required for a cluster, it is marked as core object and if the objects in it surrounding within given *Eps* are less than the minimum number of objects required, then this object is marked as noise.

The search continues for all the objects in the dataset. Later on if the minimum numbers of objects within a given radius are met subsequently previously marked objects as noise are renamed, in this way the DBSCAN differentiate between the border points of a cluster and noisy objects.

A. DBSCAN Advantages

Most of the clustering methods use distance as a measure between two clusters, which fails in detecting arbitrary shaped clusters. DBSACAN can detect arbitrary shaped clusters, which is the main feature of this technique in identifying clusters

B. DBSCAN Algorithm

Complete algorithm of DBSCAN is given by Ester et al. in [14].

C. DBSCAN Limitations

- DBSCAN has problem of high complexity, in some cases its complexity reaches to $O(n^2)$ [15].
- DBSCAB cannot work properly on significance density difference datasets [16].
- Much memory space is needed for loading the whole dataset in main memory [10].

III. DBSCAN VARIATIONS

Due to the high complexity (time and space), and density difference, many variations of DBSCAN exist in literature to overcome these problems, each variation minimizes the

978-1-4244-8003-6/10$26.00©2010 IEEE

complexity in its own way and addresses problems of the DBSCAN in different domains.

A. An Efficient Density Based Clustering Algorithm for Large Databases

Yasser et al. partitions the dataset first, by using CLARANS [2], and afterward DBSCAN is applied on each partition [10]. CLARANS is used as a pre-processing step, which partition the dataset into several small partitions. This methodology has two major contributions (dataset scans and buffer size) to reduce the complexity.

This approach uses concept of relative distance between two boarder points, instead of calculating relative inter connectivity distance between all objects in the dense region. Based on border objects, if the relative connectivity between objects is less than the *Eps* threshold defined by DBSCAN, next those clusters are merged. Noise is removed after the merging stage.

CLARANS [2] has two parameters which can affect this algorithm, one is number of partitions, and another is maximum number of neighbours. In their experimental results it is suggested by using different synthetic datasets that the number of partitions relatively high is from *8-10* and the maximum number of neighbours suggested is to be *3*. But these numbers for the above parameters are not evaluated on real datasets.

B. l-DBSCAN: A Fast Hybrid Density Based Clustering Method

l-DBSCAN uses two types of prototypes at coarser level and at finer level [15]. One is used at the coarser level, to reduce the time requirement, and another at the finer level to reduce the deviation of the results. Hybrid clustering [11] is basically used to fetch suitable prototypes from the large dataset and then to apply the clustering method using only the prototypes. In this approach the concept of leader is used, where *L* stands for leader.

In leaders method [12] a set of leaders *L* is maintained. Initially, output is produced as clusters of leaders (i.e. partition of *L*), which are further expand by mapping into clusters of the input data patterns by expanding each leader by the patterns in *D* for which it was the representative. l-DBSCAN uses two more input parameters, *Tf* and *Tc*. Fine leader *Tf* is in the boundary region, where as coarse leader *Tc* is in the middle portion of the hyper sphere. For *Tf = 0*, l-DBSCAN and DBSCAN can give exactly same clustering results.

Adding more input parameters causes overhead for user to provide and identifying suitable values for these parameters increasing the overhead of domain knowledge for setting these parameters. Similarly the output depends on these input parameters.

C. An Efficient and Fast Parzen-Window Density Based Clustering Method for Large Data Sets

Suresh et al. proposed generalization of the L-DBSCAN [15, 26], which prototypes datasets [8]. An additional parameter count is used to count the number of patterns under a prototype by using counted leader method [12]. Smooth kernel function is used to estimate density of a prototype. The algorithm produces output leader as a set shown in Eq(1).

$$L^* = \{(l, count(l) \mid l \varepsilon L\} \qquad Eq(1)$$

Hybrid clustering methodology is proposed in Parzen-Window density based clustering method [8] which assumes threshold density value *MinPts* and ε as same parameter of original DBSCAN and uses *L** instead of *D*. Main difference between these two methods, is the use of smooth kernel function which estimates density at leader. In this approach dense leaders within ε are estimated and then these leaders are enlarged with merging neighbours dense leaders. In this amalgamation a non dense leader may be a border point or a noisy leader. Finally, all leaders are clustered by DBSCAN and mapped into pertinent patterns.

Performance of experiments shows that it is good in quality and execution time than DBSCAN but as the number of patterns increases its performance reaches to DBSCAN.

D. A Linear Dbscan Algorithm Based On LSH

High time complexity of DBSCAN on large scale data is optimized by using hashing for nearest neighbor search [17]. Instead of searching for the precise neighbor points, *LSH* (Local Sensitive Hashing) [13] searches for approximate Nearest Neighbor Points. In this approach first *LSH* index is build and then clustering through DBSCAN is applied on the LSH generated index.

The main problem in this technique is adding more input parameters. The problem is of identifying suitable values for all the input parameters. Comparing the output to the DBSCAN can vary.

E. Using Grid for Accelerating Density Based Clustering

In this approach the performance of the DBSCAN is increased by using grid to partition the dataset and then merging the results [18]. Parallelism is introduced by scanning all partitions in parallel to reduce the time complexity. Grid divides the dataset into cells and then on each cell DBSCAN is applied. Cluster are merged on the basis of core point, because a point may belong to multiple partitions so those partitioned clusters should be merged, to make original clusters like that produced by DBSCAN on the original dataset. Divide in too many cells creates overhead of merging process.

Experimentally the output is not compared with the DBSCAN output. The sequence of executing steps in this approach matches to the approach given in [10].

F. A Fast Density-Based Clustering Algorithm for Large Databases

The objective of the Fast DBSCAN algorithm [19] is to improve the performance by reducing time complexity to linear. Fast DBSCAN sort all the objects, calculate the one having minimum index point neighbourhood. If the point has minimum number of objects within its radius, it is declared as core object and all its neighbourhood points are labeled through core object. All neighborhood points are not evaluated for region query, therefore reducing this recursive process of

recalculation of region query within the core object radius. DBSCAN in the expansion process revisit the previously evaluated points. Fast DBSCAN does not calculate the points more than once as it maintains labelling with the support of indexing.

Fast DBSCAN works well with the dataset with the dense region as it reduces the effective factor of number of region evaluated. The accuracy of any new density based clustering algorithm is to generate the same consistent result. The number of clusters and their shapes depends on *Eps* value. Fast DBSCAN at some extent has lesser dependency on the parameter *Eps*. To make the objects uniformly distributed in the space, kernel function is applied before clustering for better accuracy. FDBSCAN runtime complexity is inversely proportional to the *Eps* value.

FDBSCAN algorithm does not address the problem of varied density clusters. Similarly, cannot handle higher dimensional data very well. The experiments performed were on synthetic dataset.

G. *An Improved Sampling-Based DBSCAN for Large Spatial Databases*

Sampling-based DBSCAN [20] is faster than DBSCAN algorithm as dataset size increases. The execution time of IDBSCAN is much better compare to DBSCAN. IDBSCAN execution time decreases as the number of MinPts are increases, and vice versa. IDBSCAN finds the same number of clusters as DBSACAN. But in some cases IDBSCAN producing more noise points, as it treat non-core boundary objects as noise. IDBSCAN can be generalized to any possible dimension of data greater than two.

IDBSCAN algorithm does not address the problem of varied density clusters. The experiments performed are on synthetic datasets. In some cases this algorithm treats more noise points in the dataset.

H. *VDBSCAN: Varied Density Based Spatial Clustering Of Applications with Noise*

On real datasets DBSCAN algorithms find only dense clusters, but it is necessary to find sparse clusters, those sparse cluster are necessary for data mining. DBSCAN algorithms uses two fixed parameters *Eps* and MinPts, cluster density is depend on *Eps* (hard to determine), *Eps* low value making so many cluster and high value by the user can result with so many marked noise points.

The idea of varied density based method [37] is to measure different value of *Eps*, so that different densities are calculated. VDBSCAN efficiently makes varied density clusters of uneven 2-dimensional datasets. VDBSCAN computes distance of all points from their K^{th} nearest neighbor called *K-dist*; consequently, the sorted values of the points are plotted on the basis of *K-dist*. The sharp change of the *k-dist* value indicates new value of *Eps*. Two smooth curves represent the two density levels. After this DBSCAN algorithm is applied on the basis of selected value for *Eps*. DBSCAN does not repeat this process on similar values.

VDBSCAN algorithm performs good to identify varied densities cluster on 2-dimensional synthetic data. VDBSCAN is not efficient with respect to time complexity; it has same time complexity as that of the original DBSCAN. VDBSCAN is not completely automated, as it still requires user defined parameters. VDBSCAN does not handle higher dimensional dataset very well. The experiments performed were on synthetic datasets.

I. *DDSC: A Density Differentiated Spatial Clustering Technique*

DBSCAN finds clusters with different shapes and sizes, but within cluster there is wide variation of densities. DDSC overcome this shortcoming of the DBSCAN [9]. If we increase *Eps* value, one big cluster is formed, which contain different density clusters within it radius and outside cluster so many noise points are identified, too small value of *Eps* come with the small clusters.

DDSC requires minimal requirements of domain knowledge. DDSC is better algorithm as compared to OPTICS [3], which is an old approach to find the difference in densities. OPTICS does not use predefined *Eps* value and it also does not detect noise as well. Comparable to OPTICS, DDSC handle this problem in a better way by removing the dependency of *Eps* and *MinPts*.

DDSC algorithm does not reduce the time complexity significantly. The experiments performed were on synthetic datasets.

J. *Mining Biomedical Images with Density-Based Clustering*

DBSCAN is used for identification of homogenous colour regions in biomedical images of skin tumours. In this approach input image is split into four segments, mean colour of image and its segments computed [1], the Euclidean distance of mean color of image and sub regions is calculated. This process is repeated to get homogeneity, which occurs when the difference is less than the threshold value.

DBSCAN results were visually examined by experts, out of which 80 % of the results were found correct. The proposed algorithm accurately detects the lesion borders in 80% of the test images. The performance of this technique is verified by Kappa statistics [25]. DDSC algorithm does not reduce the time complexity significantly. The experiments performed are on synthetic datasets.

K. *Motion Determination Using Non-Uniform Sampling Based Density Clustering*

This technique tries to overcome the problem of DBSCAN over significant density difference between different moving regions in frame difference images [16]. Frame difference images are normally continuous and having multiple densities regions with varying densities. The non uniform sampling method is viewed as a preprocessing scheme for frame difference image. In order to distinguish dense regions from sparse regions, two density thresholds are used (DT_{low} and DT_{upper}).

Frame difference is transformed in to binary images, and then uses non uniform sampling method to obtain samples of frame difference images. In third step the sample frame

difference images are transformed into points for applying DBSCAN on them. Based on the result of the DBSCAN minimum rectangle for each cluster is identified and the portion of the image and size is computed. Finally, the moving objects identified by the minimum rectangles are labeled as moving objects.

This technique is specifically evaluated with spatial data. Overall the performance or output of DBSCAN is not improved; only working of DBSCAB for motion detection is elaborated.

L. A General Framework for Adaptive and Online Detection of Web Attacks

Anomaly intrusion detection [28] is a commonly used concept in the computer networks that build a sketch of a subject's normal and abnormal activities to find any improper divergence from normal activities to detect attacks. Intrusion Detection System (*IDS*) have some problems like obtaining large amount of precise labelled data to prepare the detection model and regular up-gradation of model with new labelled data.

DBSCAN is extended to Str-DBSCAN [7] that is appropriate for clustering streaming data and for detection models of the framework. For streaming environments, Str-DBSCAN clusters all the current exemplars as well as the outliers. The exemplars are continuously assigned some weights to strengthen frequent exemplars and forgot non-frequent exemplars in the process of clustering. Adaptive anomaly detection methods, Str-DBSCAN are more effective in online detection of attacks than static detection method because it adapts to the behavioural changes and summarize the historical data into simple concepts.

Other algorithm like StrAP can also be used in the framework to build detection models. Similarly complexity of this approach remains the same.

M. C-DBSCAN: Density-Based Clustering With Constraints

Clustering quality [29], enhancement in computation time [30], and construction of empty clusters prevention [31] is improved by using constraints. C-DBSCAN [6] shows the use of the constraint driven cluster creation at instance level, which improves the quality of the cluster. It has been shown that constraint based density clustering at instance level improves the cluster quality up to great extent.

Initially in C-DBSCAN the data space is partitioned into subspaces and subsequently, it enforces instance-level constraints on the data to derive the cluster construction. Clustering with constraints is also known as semi-supervised clustering. Must-Link and Cannot-Link constraints are supported by C-DBSCAN among data instances/points, where must-link has to be in same cluster and cannot-link must be in different clusters. C-DBSCAN extended DBSCAN by applying KD-Tree to partition data space, and enforces Cannot-Link constraints to produce local clusters. Then Must-Link constraints are implemented to merge these local clusters.

The interpretation and enforcement of constraints must be careful to make correct clusters otherwise it may prevent merging of clusters.

N. Privacy-Preserving DBSCAN Clustering Over Vertically Partitioned Data

Generally clustering algorithms deals single data source to collect similar data, however, the data in many applications are from different multiple sources. The extraction of information from these data is to get them together, raises need for its privacy [5]. A technique called privacy preserving data mining [32] solves the problem of privacy and is based on *SMC* (Secure Multi-party Computation protocol) [33] which is a basic protocol to DBSCAN algorithm that keeps privacy and mine clusters from vertically partitioned data.

The two main aspects of the protocols are the Correctness and Security, and here it is proved that the protocol *SMC* correctly implements region Query. The *SMC* protocols are not perfectly secured although disclose information does not damage the privacy of data. This algorithm increases time and communication complexity very little but preserves privacy. To reduce time in local clustering *MinPts* and *Eps* values can be change accordingly but finding these values is a difficult task. Working is not supported with results on synthetic or real datasets.

O. ST-DBSCAN: An Algorithm for Clustering Spatial-Temporal Data

KDD [34] main focus is on the discovering of non-spatial and non-temporal data clusters that is why they are unfeasible for spatial temporal data [35] [36]. ST-DBSCAN [4] requires four parameters *Eps1, Eps2, MinPts* and $\Delta\varepsilon$. *Eps1* is used for spatial attributes as a distance parameter and *Eps2* is the distance parameter for non-spatial attributes. The minimum number of points between *Eps1* and *Eps2* is the *MinPts*. To avoid discovering of combined clusters $\Delta\varepsilon$ is used. The main function of this algorithm is the Retrieve_Neighbours function which is used to retrieve density-reachable objects from particular object according to *Eps1, Eps2*, and *MinPts*.

Here the need is to pick up performance of algorithm by, running algorithm in parallel, making use of functional heuristics to find appropriate values of Eps and MinPts, making better spatial indexing structure and adding some filters can have better impact on performance which are necessary to make it mature.

IV. PARAMETERS

All the methods to enhance the DBSCAN algorithm with different deficiencies are analyzed on certain parameters to compare all these methods with one another and with the original DBSCAN.

A. Problem to overcome:

This parameter shows the problem with the original DBSCAN and shows the need for the new variation.

B. Methodology:

This parameter shows the approach to overcome the problems identified in *A*.

TABLE 1: COMPARISON OF ALL VARIATIONS

Parameters variations	Problem to overcome	Methodology	Reduce Complexity	Input parameters	Output to DBSCAN	Dataset for experimental evaluation	Noise removal
A. [11]	Complexity	Portioning	Yes	Same	Same	Synthetic	Final stage
B. [15]	Complexity	Prototypes leader	Yes	add (TF, TC)	depends on input	Synthetic + real world but	Separate class
C. [8]	complexity	Counted leader method Prototypes	yes	L* instead of D,	Little improved	Synthetic + Real-world	Final stage
D. [17]	complexity	hashing	Yes	Add M,L,K	Can vary	Synthetic + UCI	Yes
E. [18]	Complexity	Portioning and merging	yes	Same	Same but not proved	synthetic	Yes
F. [19]	Complexity	Partition	Yes	Same	Better	Synthetic	Final stage
G. [20]	small memory and minimum I/0 cost	Partitioning in quadrants	Yes	Same	Same	Synthetic	Final stage (more noise points generated)
H. [37]	Varied density	Partitioning	No	Predefined K-dist	Better (identify varied densities)	synthetic	Final stage
I. [38]	Varied density	Partitioning	no	No predefined parameters	Better (identify varied densities)	synthetic	Does not noise well
J. [1]	skin identification	partitioning	no	Npred, wCard(), MinCard	NA	Real data set	Final stage(damaged skin)
K. [16]	Varied density, complexity	Portioning Frame difference images	Yes	Same	Better (identify varied densities)	Subimages	Yes
L. [7]	Detect Online web-Attacks	Scane but to specific Framework	No	Same	Better (identify varied densities)	Real-world data	First stage
M. [6]	Quality clustering, complexity	Partioning Enforcing constraints on data	yes	Same	Improved	synthetic, Real data	Initial stage
N. [5]	Correctness, Security	SMC protocols in Region Query	no	Same	Same	proven by theorems only	Final stage
O. [4]	Spatial-Temporal data, varying density	Partitioning	no	Add two more (Eps2, $\Delta\varepsilon$)	Better (identify varied densities)	Real world problems	Final stage

C. *Reduce Complexity:*

This parameter further analyse the complexity of the proposed approach, as it is major problem with the original DBSCAN.

D. *Input parameters:*

Inputs are important factor for the performance of any approach, normally the DBSCAN works with two input parameters, where as certain methods adds some more input parameters.

E. *Output to DBSCAN:*

In most of the cases it is intended that the new approach produce the same result as that of the original DBSCAN or not.

F. *Dataset for experimental:*

Evaluating the dataset used for experimental performance either synthetic or real world dataset.

G. *Noise Removal:*

Noise removal is one of the important features of the DBSCAN. This feature is analyzed in each approach and the way how it is handled.

The values of all these parameters are given in Table 1 against all the approaches discussed in Section III.

V. ANALYSIS

Two problems are basically tried to overcome by most of the techniques, one is of complexity and another is of ill working of DBSCAN on varied densities. Two main strategies are proposed by most of the researchers tried to overcome the complexity problem. One important strategy is portioning and second one is using hybrid approach. In few cases the complexity of the varied density algorithms [20] [1], does not reduces the complexity. In these algorithms, a partition with different parameters is formed and then traditional DBSCAN is applied. So, in these complexity of the algorithms is same as DBSCAN.

Another success of the proposed approaches is on varied densities, for this purpose normally two *Eps* are used. This solves the problem up to certain extent, but still working is needed to address this problem.

In terms of input parameters it is analysed that in most of the cases the new working is proposed with same input parameters, but for varied densities few more parameters are added to achieve the objective. In case of partitioning the working is defined with the same input parameters. The reason is that in partitioning the original DBSCAN is applied on smaller datasets. To compare the output of the new variations to the original DBSCAN it is absorbed that in most cases the output remains same or is improved. In improvement of output we means that DBSCAN was not able to retrieved the desired output in terms of clusters objectivity (increasing the intersimilarity between the cluster objects). To show the performance of the new variations, in most of the cases the experimental results are given on synthetic dataset instead of applying on real world datasets. The reason of applying on synthetic dataset instead on real world dataset is the comparison on artificial benchmark data with other techniques. Noise removal and the effect of high dimensionality also remain same like the original DBSCAN. Almost all the methods remove the noisy objects and suffer with degrading problem on high dimensionality data.

VI. CONCLUSION

Research focused on the two main problem of DBSCAN after its fame in the field of data mining. Different new

variations are proposed to focus these problems. Each variation still has some limitations more research can be done on the problem of density differences. “A Linear DBSCAN Algorithm Based On LSH” [17] and “A Fast Density-Based Clustering Algorithm for Large Databases” [19] are the two most efficient approaches which reduces the time complexity up to linear, and provide comparative good result to DBSCAN because of the reason that DBSCAB does not perform well on varying densities.

The reduction in time complexity in these two approaches is due to the use of indexing scheme. Indexing improves the time complexity up to great extent. The increase in dimensionality effect the original DBSCAN, the performance of the DBSCAN gets worst with the curse of increasing dimensionality. Similar effect of increasing dimensionality is analyzed in each approach. In few approaches this problem is handled up to certain level but still in most of the cases the curse of dimensionality remains a problem.

REFERENCES

[1] M. Emre Celebi ,Y. Alp Aslandogan and Paul R. Bergstresser ,” Mining Biomedical Images with Density-based Clustering”

[2] R.T. Ng and J. Han.:” Efficient and Effective Clustering Methods for spatial data mining”, Proc. 20th Int. Conf. on Very Large Data Bases, santiago, Chile, 1994, pages 144-155.

[3] M. Ankerst, M. M. Breunig, and H-P Kriege1,”OPTICS: Ordering Points to Identify the Clustering Structure,” In proceeding of ACM SIGMOD, 1999, pp 49-60

[4] Birant, D. Kut, A.,“ST-DBSCAN: An algorithm for clustering spatial-temporal data”, Data and knowledge engineering, volume 60, 2007

[5] XU Wei-jiang, HUANG Liu-sheng, LUO Yong-long, YAO Yi-fei, JING Wei-wei, "Privacy-Preserving DBSCAN Clustering Over Vertically Partitioned Data," mue, pp.850-856, 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07), 2007

[6] Ruiz, C., Spiliopoulou, M., Menasalvas, E.: C-DBSCAN: Density-Based Clustering with Constraints. In: RSFDGrC’07: Proc. of the Int. Conf. on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing held by JRS’07 (2007)

[7] Wang W., “A General Framework for Adaptive and Online. Detection of Web attacks”, Project AxIS, INRIA Sophia. Antipolis. 2004

[8] V. Suresh Babu, P. Viswanath, "An Efficient and Fast Parzen-Window Density Based Clustering Method for Large Data Sets," icetet, pp.531-536, 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008

[9] B. Borah, D.K. Bhattacharyya ,” DDSC: A Density Differentiated Spatial Clustering Technique”, Journal Of Computers, Vol. 3, No. 2, February 2008

[10] Yasser El-Sonbaty, M. A. Ismail, and Mohamed Farouk,”An Efficient Density Based Clustering Algorithm for Large Databases,” ICTAI 2004.16th IEEE International Conference, pp673 - 677

[11] E. Y. Cheu, C. K. Kwoh, and Z. Zhou. On the two-level hybrid clustering algorithm. In Proceedings of International Conference on Artificial Intelligence in Science and Technology, pages 138–142, 2004.

[12] H. Spath. Cluster Analysis Algorithms for Data Reduction and Classification. Ellis Horwood, Chichester, UK, 1980.

[13] P. Indyk and R. Motwani, “Approximate Nearest Neighbor: Toward Removing the Curse of Dimensionality”, Proc. Symp. Theory of Computing, Dallas, TX. May, 1998.

[14] Ester, M., Kriegel, H. P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (1996) 226-231

[15] P. Viswanath and R. Pinkesh. l-dbscan : A fast hybrid density based clustering method. In Proceedings of the 18th Intl. Conf. on Pattern Recognition (ICPR-06), volume 1, pages 912.915, Hong Kong, 2006. IEEE Computer Society.

[16] Sang, Y. Yi, Z. “Motion Determination Using Non-Uniform Sampling Based Density Clustering” , Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008.

[17] Wu, Y. Jou, J. Zhang, X., “A Linear Dbscan Algorithm Based On Lsh”, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

[18] Mahran, S. Mahar, K., “Using grid for accelerating density-based clustering”, Computer and Information Technology CIT '09. Ninth IEEE International Conference, Sydney, 2009

[19] Bing Liu, "A Fast Density-Based Clustering Algorithm for Large Databases," Machine Learning and Cybernetics, 2006 International Conference on , vol., no., pp.996-1000, 13-16 Aug. 2006

[20] Borah, B.; Bhattacharyya, D.K., "An improved sampling-based DBSCAN for large spatial databases," Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on , vol., no., pp. 92-96, 2004

[21] R. Xu, “Survey of Clustering Algorithms,” IEEE Transaction on Neural Networks, vol. 16, no. 3, May 2005.

[22] B. Borah and D. K. Bhattacharyya, “A clustering technique using density difference,” in Proceedings of International Conference on Signal Processing, Communications and Networking(ICSCN-2007), Mar. 2007, pp. 585–588.

[23] L. Ertoz, M. Steinbach, and V. Kumar, “Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data,” in Proceedings of Second SIAM International Conference on Data Mining, Jan. 2003.

[24] Density Clustering Based on Radius of Data (DCBRD), A.M. Fahim, A. M. Salem, F. A. Torkey, and M. A. Ramadan World Academy of Science, Engineering and Technology 22 2006

[25] S Zhou, A Zhou, J Cao, J en, Y Fen and Y Hu, Combining sampling technique with DBSCAN Algorithm for Clustering and Data Mining.

[26] E. Y. Cheu, C. K. Kwoh, and Z. Zhou. On the two-level hybrid clustering algorithm. In Proceedings of International Conference on Arti_cial Intelligence in Science and Technology, pages 138.142, 2004.

[27] Usama Fayyad and Ramasamy Uthurusamy. 1999. Data mining and knowledge discovery in databases: Introduction to the special issue. Communications of the ACM, 39(11), November.

[28] K. Ingham and H. Inoue. Comparing anomaly detection techniques for http. In 10th International Symposium on Recent Advances in Intrusion Detection, 2007.

[29] Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-means Clustering with Background Knowledge. In: ICML’01: Proc. of 18th Int. Conf. on Machine Learning. (2001) 577–584

[30] Davidson, I., Ravi, S.S.: Clustering with Constraints: Feasibility Issues and the k-Means Algorithm. In: SIAM’05: Proc. of the SIAM Int. Conf. on Data Mining. (2005)

[31] Bennett, K., Bradley, P., Demiriz, A.: Constrained K-Means Clustering. Technical report, Microsoft Research (2000) MSR-TR-2000-65.

[32] Y Lindel, B Pinkas, “Privacy Preserving Data Mining”, In Advances in Cryptology-CRYPTO’00, volume 1880 of LNCS. Springer-Verlag, 2000.36-54

[33] A. C. Yao, “Protocols for secure computations”, In proceedings of the 23rd Annual IEEE symposium on Foundations of Computer Science, 1982.

[34] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of Intelligent Information Systems 17 (2–3) (2001) 107–145.

[35] T. Abraham, J.F. Roddick, Survey of spatio-temporal databases, GeoInformatica, Springer 3 (1) (1999) 61–99.

[36] E. Kolatch, Clustering algorithms for spatial databases: a survey [online]. Available on the web, 2001

[37] Peng Liu; Dong Zhou; Naijun Wu, "VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise," Service Systems and Service Management, 2007 International Conference on , vol., no., pp.1-4, 9-11 June 2007