IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. III (Sep - Oct 2018), PP 09-16
www.iosrjournals.org

A Parameter free Clustering of Density Based Algortihm

Mr. Aakash Kulmitra, Mr. Ram Nivas Giri
Computer Science and Engineering,Raipur Institue of Technology, Raipur India
Assitant Professor ,Computer Science and Engineering,Raipur Institue of Technology, Raipur India
Corresponding Author: Mr. Aakash Kulmitra

Abstract*—Clustering is a kind of unsupervised learning process in data mining and pattern recognition, most of the clustering algorithms are sensitive to their input parameters. So it is necessary to evaluate results of the clustering algorithms. It is difficult to define which clustering designs are acceptable hence several clustering validation measures are developed. In the present paper we have a study of implementation on some of very common data sets with internal index compared to our new and improved parameter free Density Based Clustering with the help of differentical evolution. Density based clustering methods are being used for clustering spatial databases with noise. Density Clustering of Spatial Data and its Application with noise (DBSCAN) can discover clusters of arbitrary shapes and sizes effectively with the help of Eps(radius of the cluster) and MinPts (minimum number of points to be inside the cluster). The value of these parameters is very important in determining clustering results as the output varies significantly with the little changes in the values and is also very hard to determine these parameters a priori. In this paper we present a new algorithm named Parameter free Density Based Algorithm using Differential evolutions, which uses the combination of analytical ways to determine the efficient values of Eps and Minpts using Differential Evolution Method. The Experimental results show that our algorithm is precise in selecting the parameters and efficient.*

Keywords*—DBSCAN , validation measures,sensitive input parameters,differential evolution.*

Date of Submission: 08-10-2018 Date of acceptance: 22-10-2018

I. Introduction

Data clustering is the kind of unsupervised learning technique which divides the given data into no of groups of subclasses such that the object of the same subclasses are much more similar compared to the objects of the another subclasses.

The technique of clustering is being used in various fields such as image analysis [1] , pattern recognition[2], knowledge discovery[3] , and medical analysis[1] which is being posed to identify clusters with arbitrary shapes and sizes, by determiningthe input parameters of algorithms with minimum requirements of domain knowledge and still a good efficiency on large databases.

Various clustering methods follow various different approaches to solve a specific data set. One of the methods of clustering is thepartitional clustering which uses the k-means[4] algorithm to find clusters of spherical shapes only and need to supply with the number of clusters as an input to the algorithm. Kernel K-means detect arbitrary clusters by transforming them into kernel functions which is having a time complexity of O (n^2) and hence not efficient for large data sets. Hierarchical clustering is the another method of clustering which partitions the data sets into hierarchical structure clusters, which are obtained by combining the subsets at various levels using minimum distance criteria[5].The hierarchical method are having a time complexity of O(n^2) and should also define an appropriate stopping condition for thesplitting and merging of partition for deriving a cluster. BIRCH[6], is a kind of hierarchical algorithm which uses tree based representation for reducing time complexity but it can only find spherical data sets and also clustering result is affected by input order of data. Currently, semi-supervised[7] and multi-view based[8][11] methods have shown effective improvement in the accuracy of clustering. Semi-supervised clustering algorithms utilize small amount of labeled data from the user for better clustering.

Clustering algorithms categorize the data into a specific number of clusters C. There is not a single precise definition available about this[9]. But in most of the research papers cluster is being defined by considering into the internal consistency and the external disassociation [10],[11] that is the paradigms in the same clusters must be identical to each other and that in the different clusters must not. The identical and non-identical both the parameterscould be represented in a clear and significant way

Density Based Spatial Clustering of Applications with Noise(DBSCAN)[12] is the initial lead of density based clustering algorithms which can discover arbitrary shapes and also handle noise outliners efficiently. DBSCAN has the quadratic time complexity with data set size which can be extended to large datasets by reducing its complexity using spatial index structures like R-trees[13] for finding neighbors of the pattern

Still, they cannot be applied of the high dimensional data set. In the following paper we are showing the proposed DBSCAN algorithm with the comparison to our new and improved Parameter Free Density Based Clustering of Spatial Data with noise removal (PDBSCAN) and on evaluating we could see that the clustered results are similar to the output of the traditional DBSCAN algorithm but their efficiency or accuracy is increased in terms of indexes.

II. Related Work

DBSCAN[14]is the Density Based Spatial Clustering algorithm with noise elimination. An algorithm which uses density based clustering approach to form spherical shaped clusters in order to cluster datasets with varied density. In this paper Ergun Bicici proposed a new algorithm LSDBC[15] which impulse local scaling by guessing the threshold based on the locally available data in the density clustering. Here the K-nearest neighbors used are potential centers of the clusters with the discovering of local maxima. The value of the cluster is increased until its density falls below the already specified density parameters resulting in the clustering of noise and arbitrary shaped data sets on density gradients. DeryaBirant and Alp Kut proposed an another algorithm with the complexity same as DBSCAN algorithm named STDBSCAN[16]which used the values of two radius Eps1 and Eps2 for two different dimensions spatial dimension and temporal dimension which is being used to handle the varied density points by defining the density factors. Min points with a density threshold value to give the outputs for the datasets with two different dimensions. In 2007 Peng Liu ,Dong Zohou proposed another variant of DBSCAN named VDBSCAN[17] which was also having the complexity similar to the DBSCAN algorithm which uses the input parameters as minpts, eps the radius and the K-nearest neighbors. Basically dependent on the concept in which we could use the values of eps as radius which is being determined by the values of K-nearest neighbors to get the varied density values of various varied density datasets. In the year 2008 B.Borah and D.K. Bhattacharya proposed a new technique named DDSC[18] with the complexity of O(nlogn) having radius Eps minimum points Minpts and the density threshold value K as the input parameters for the technique which mainly focuses on partitioning the datasets such that the adjoining regions significantly get differs in the density by the use of homogeneity testing for detecting the variations in the density. In the year 2009 A.Fahim ,G.Saake and their fellow authors introduced an another new technique called as Enhanced DBSCAN[19] whose complexity was equal to the complexity of the DBSCAN with the input parameters as no of nearest neighbors and a new parameter with the limitation of the greatest density minpts. Which is based on the concept of local density functions and to finds its value at each points which will be an approximation of overall density function.Huang Darong and Wag Peng came up with a new idea and a new technology in the year 2012 by name GRPDBSCAN[20] and this technique was also having the same complexity as DBSCAN this technique was quite different from the above technique as it was also consisting of the grid technique with it and the input parameters which were used as the number of grid units N. The main goal of this is, it combines the grid partition technology with the multidensity based clustering. Since, this technique also had the same drawback it could also not handle the datasets with varied density.Year 2013 Carmelo Cassisiand fellow team members proposed an algorithm named ISDBSCAN[21] which is used to enhance the density-based clustering , having the same complexity of the DBSCAN where we take no of nearest neighbor –K as the input parameter and used the main concept of space stratification based on both the INFLO function and KNN distance.

A paper was proposed to reduce the computational complexity related to the conventional DBSCAN. By implementing the merging function at the starting at the evaluation of the clusters. Though these DBS algorithms are not computationally efficient but still they are effective on giving the outputs so in this algorithm they merged with the new cluster merging technique which will lead to computationally efficient outputs. In year 2015LE-MHIN KUE, Ashish Bhakar, Edward Chung, proposed a new technique A Modified Density Based Scanning Algorithm with noise for spatial Travel pattern Analysis from Smart Card AFC data[12]. An application of the DBSCAN algorithm by using smart AFC data while proposing an algorithm named weighted-DBSCAN where they tested the passenger dataset and examined that dataset with their given algorithm, compared with traditional DBSCAN and proved that it worked more efficiently thanthe conventional DBSCAN but this was also failing to implement the high dimensional dataset. This algorithm could run 45% faster than conventional DBSCAN. In year 2015 JAISHU ZHANG and JhonKerekes implemented a new technique based on density based spatial clustering named Adaptive Density Based Model for Extracting Surface Returnsfrom Photon-Counting Laser Altimeter Data[22]. They proposedan application of density based algorithm by computing them against the datasets taken from the real time such as The Ice cloud and the land Elevation satellite 2 to smooth the rounding edged of the algorithm and gave them efficient output by adaptive based model.

Density based clustering methods can find random shaped clusters in the dataset and also insensitive to noise. In density based clustering approaches clusters are formed by merging solid areas separated by regions of free areas. DBSCAN is proposed for clustering large spatial databases with noise or outliers. OPTICS [23] is an extension to DBSCAN which can find clusters with fluctuating densities by creating an augmented ordering of given dataset demonstrative a density-based cluster structure. This assembling is equal to a density-based clustering with diverse range of parameter conditions. H Gao[24]proposed a parameter free clustering method that utilizes Affinity Propagation algorithm [25] to detect local densities in the dataset and obtains a standardized density list . Later, DBSCAN method is modified to cluster the dataset in terms of the parameters in the standardized density list. DENCLUE [8] defines clusters by a local maximum of estimated kernel density function . A hill climbing practice is used for assigning points to adjacent local maximum. L-DBSCAN [26] is a hybrid density based clustering process that first develops a set of prototypes from the dataset using bests clustering method[27]and runs DBSCAN on the prototypes to discover clusters. Further, Rough-DBSCAN [28] is proposed by smearing rough-set theory [29] to L-DBSCAN method It has a time complexity of O(n) but the cluster results are prejudiced by threshold parameter that is specified to originate the prototypes . Recently, fast and scalable density based clustering method using graphics processing units(GPU) are proposed to advance the performance DBSCAN[30]. M Tang(2016) [31] parallel and distributed versions of DBSCAN are proposed for management large datasets using graph algorithmic perceptions and achieves well balanced capacity by taking compensations of tree based bottom-up approach to construct clusters.

2.1 DBSCAN : A Denisty based Clustering approach -

DBSCAN algorithm defines cluster as a region of compactly connected points parted by regions of non-dense points. If similarity measure is taken as Euclidean distance the region is a hyper sphere of radius eps at the given point c as midpoint.

- Eps-neighborhood: for a point $i \in P$, the eps-neighborhood symbolizes the set of points whose remoteness from i is less than or equal to eps. The cardinality of eps-neighborhood defines the threshold density of i.
- Eps-connected: for a pair of points i; $y \in P$, if $||i - y|| \leq eps$, then i ,y are eps-connected points. From the view of a DBSCAN method all point in the dataset will plunge into either core point or border point. Further a border point can be whichever noise point or density connected point
- Core point: A point with threshold density superior than or equal to minpts.
- Border point: A point with threshold density fewer than minpts.
- Noise point: A point d is a noise point if the threshold density of p is fewer than minpts and all points in the eps-neighborhood of d are border points.
- Density connected point: A border point with at least one core point in its eps-neighborhood.

The following table1 consist of the notations used in the paper. While the rest of the paper is organized as follows. Description of the original DBSCAN method is shown in section II. Various algorithms which are used with DBSCAN (Density based clustering of spatial data) are shown in Section III. The description of our proposed algorithms and its working is shown in section IV. The experimental operations performed on our proposed PDBSCAN algorithm and on DBSCAN algorithm with their performance analysis is shown in section V. Concluding remarks of the operations are shown in section VI.

Table-1Notations

Symbol	**Denotes**
P	Input set of data patterns
I	A pattern in P
D	No of dimensions of a pattern
M	Size of data set

DBSCAN algorithm takes two constraints as an input eps and the minpts.eps which stipulates the maximum distance neighborhood for the specified point minpts is the minimum number of facts required in the eps-neighborhood of a point of a cluster.Originally all points are discernable unvisited.

The algorithm starts by randomly choosing an unvisited point and discovering its eps-neighborhood. If the number of points in the eps-neighborhood are less than the minpts then it is noticeable as noise or outliner, otherwise it is measured as dense point and a new cluster is created. Further the same procedure is carried on is no new points can be additional to the cluster then we see that the new cluster is whole and no points will be added to the cluster in succeeding iterations. Hence to find the new cluster from the given data set we have to replicate the same process up till the new cluster is molded. The process stops when all the nodes are assigned to some cluster or originated as noise points. Each points in the cluster is eps-connected with at least one point in the matching cluster to which it belongs and is not eps-connected with any other points in residual clusters. However there may be points (border) which may be associated to the border points of other cluster, in that case the cluster is allocated to the cluster that managed it first. Such cases are infrequent in practice. Total no of eps-neighborhood processes performed is the size of the dataset.

Algorithm 1. DBSCAN(eps,minpts,P)

```
Mark all patterns in P as unvisited
cluid←1
for each unvisited pattern i in P
Do
    Z←FindNeighbours(i,eps,minpts)
    if|$z$|<minpts
        Mark I as noise
    Else
        Mark i and each pattern of Z with child
        Queuelist ←all unvisited patterns of z
        Until queue is empty
        Do
            y←delete a pattern from queue
            Z←FindNeighbors(y,eps,mimpts)
            if|$z$|≥minpts
                for each pattern w in z
                    mark w with child
                    If w is unvisited
                        queue←wU queue
                end for
            end if
            mark y as visited
        end until
    end if
    mark I as visited
end for
Output  all patterns in P marked with cluid or noise
```

2.2 DIFFERENTIAL EVOLUTION ALGORITHM :-

Differential Evolution (DE) is a new Algorithm in evolutionary computing method for global optimization over continuous spaces proposed by Storn in 1977[32].Differential evolution is similar to the overall structure of genetic algorithm[33][34].DE comprises of basic three operations, mutation ,crossover and selection. In DE mutation is the most important operation in the DE as it generates new elements for population , which may contain optimum solution for the objective function[35][36]. The DE algorithm is as follows[37].

2.2.1 Initialization :- Generates arbitrary initial population in n dimensional space as follows.

$$x_i(j) = x_j^l + rand(0,1).x_j^u \qquad (1)$$

Where $x_i(j)$denotesjth variable for ith individual, and x_j^l and x_j^u are the lower and upper constraints, rand(0,1) represents uniformly distributed rand value within [0,1].

2.2.2 Mutation :- DE randomly selects two population vectors $x_{p2,}x_{p3}$(p2≠p3) which must be different from each other, then uses the difference between the individuals $x_{p2,}x_{p3}$ by scaling factor η (usually set the value within (0.5 1) [35] to mutate x_{p1} by equation.

$$h_i = x_{p1} + \eta\left(x_{p2} - x_{p3}\right) \qquad (2)$$

2.2.3 Crossover :- The new individual is generated by recombining $x_i$and h_i which is represented as:

$$U_i(j) = \left\{\begin{matrix} h_i(j), if\ rand(0,1) \le CR\ or\ j = j_{rand} \\ x_i(j),\ Otherwise \end{matrix}\right\} \qquad (3)$$

Where rand(0,1) is a random number within [0 1], j_{rand} is a randomly chosen index to ensure thatthe

train vector $U_i$does not duplicate $x_i$and CR is the crossover rate.

2.2.4 Selection :- Greedy algorithm is used to select the better one between the trail individual U_i and the parent vector x_i for the next generation using a fitness/cost function f:

$$x_i = \begin{Bmatrix} U_{i,}\ if\ f(U_i) \geq f(x_i) \\ x_i\ , otherwise \end{Bmatrix} \qquad (4)$$

III. Proposed System : Parameter Free Dbscanalgorithm With Diffrential Evolution

In this section, we present a new clustering algorithm named (PDBSCAN) Parameter free Density Based Spatial Clustering Applications with Noise. Objective of algorithm is, eliminate the parameters required and the limitations which come through while executing traditional Density Based Clustering approaches.The PDBSCAN Parameter freed Density Based Spatial Clustering Applications with noise uses the binary coding technology where each individual or Minpts parameter is being represented by the bit string. The procedure for the PDBSCAN is explained step wise in the following.

Step 1:-Set control parameters and initialize the binary-coded populations randomly.
Step 2:- Calculate the best fitness function in terms of the internal and external clustering indexes for each individual and keep the maximum indexing value as the Best Solution.
Step 3:- Generate the mutated individual value as per equation 3.
Step 4:- Generate the new Trail individual by using the Crossover Function in equation 4.
Step 5:- Applying the limit over the real-coded individuals by equation 6.
Step 6:- Evaluate the target individual and corresponding trail individual, and choose the better one with the maximum value of index (fitness function) to survive the next generation. Also keep the best one into the Best Solution.
Step 7:- If the Terminal conditions are meet , terminate iteration or else got to *step 2*.

Generally the evolution process terminates if the maximum generation is reached or the minimum/ maximum fitness value is satisfied. There are some definitions over the proposed algorithm which are as follows.

1.1 Eps Parameter : As the Eps parameter can largely degrade the value or efficiency of the DBSCAN algorithm[38], the combination of an analytical way of estimating Eps is employed. The tournament selection (TS)[39] method based on its specifications creates more diverse Eps values until an appropriate combinations of MinPts and Eps values be selected. The Eps parameter can be calculated by the analytical way as.

$$Eps = \left(\frac{\left(\prod_i^{\max(x) - \min(x)} i \right) * k * \gamma\,(0.5 + n + 1)}{m * \sqrt{\pi^n}} \right) 1/n \qquad (5)$$

Where x is data matrix by m-objects and n-variables, k is the number of objects in neighborhood of an object, and γ interpolates the fractional function. In each iteration, the Eps parameter is calculated and compared to the stored Eps values from previous iterations through tournament selection method. At first, this initial Eps value has high probability by default in order to be selected in tournament among others. It might be very well suited selection of Eps and Minpts.

1.2 Apply limit over real-coded vectors : Since the standard operators generate real-coded vectors not the bit strings, a simple control is proposed to values get either 0 or 1.

$$Binary - coded = \min(\max(Real - coded, 0)\,, 1) \qquad (6)$$

1.3 Uniform Distribution of Scaling Factor : For scaling factor in mutation operator, a random number with the size of nVar bit strings is generated from the continuous uniform distributions with lower and upper bounds values specified by β_{min} and β_{max} , respectively.

$$\eta = unifrind(\beta_{min}\,, \beta_{max}\,, nVar) \qquad (7)$$

Experimental Results

In this section the experimental consequencesachieved to demonstrate the effectiveness of the algorithms are observed. Processes on varied concentration data sets were carried in this, in directive to check the complication and the proficiency of density based clustering methodology.The experiments were carried on different data sets taken from UCI machine learning repository of varied dimensions. The details of each datasets are present in the Table no 2. In the next part we studied the nature of our proposed algorithm with some benchmark data sets with respect to various clustering validity indexes internal as well as external validity indexes.

All the experiments were performed on an Intel core i3 5th generation processor with 2.5GHz and 4GB RAM running windows 8.1. All programs are accumulated and executed on the MATLAB R2016a GUI framework.

1.4 Experiment 1.

In this we have used some popular datasets are being used to perform clustering. A summary of data sets used is given in table no 2

Table -2Description of datasets used in testing

Dataset	**Instances**
Aggregation	788
Compound	399
Flame	240
Iris	150
Jain	373
Wine	178

1.5 Experiment 2:

In this experiment the scalability for the datasets with noise was checked. Noise plays a very significant role in evaluation of the datasets as due to noise effect the efficiency of the algorithm may effect and this may also strengthen changes in the output. The results were checked with respect to various validity indexes of clustering internal as well as external indexes were used to validate the efficiency of the algorithm

- F-meausre
- Rand Index.

Rand Index :-The Rand index[40] or Rand measure (named after William M. Rand) in statistics, and in specific in data clustering, is a quantity of the resemblance between two data clustering. A method of the Rand index may be defined that is accustomed for the chance alliance of elements, this is the adjusted Rand index. From a mathematical position, Rand index is connected to the accuracy, but is appropriate even when class labels are not used.

The succeeding indexing values for algorithms on several datasets could be witnessed in the Table 3, and Table 4 respectively given below. The comparative graphs for each algorithm in harmony with the specific validity index can also be understood in the graphs given below where we have related and taken the productions of each algorithm and plotted its production according to the standards of the validity indexes which were taken into contemplation and signified graphically in following Figure 1 and Figure 2 respectively.

Table -3Evaluation of Rand Index

Datasets	**Eps**	**Minpts**	**Iterations**	**Rand Index**
Aggregation	1.4905	8	40npop 50	0.9949
Compound	1.4864	4	40npop 50	0.9919
Flame	1.2980	8	30npop 50	0.9762
Iris	0.4057	4	30npop 50	0.9817
Jain	2.0175	10	30npop50	0.9823
Wine	22.312	5	30npop 50	0.9925

Table -4Evaluationof F-measure Index

Datasets	**Eps**	**Minpts**	**Iterations**	**F-measure Index**
Aggregation	1.3917	7	40npop 50	0.9895
Compound	1.4744	3	40npop 50	0.9782
Flame	0.9224	4	30npop 50	0.9550
Iris	0.4014	4	30npop 50	0.9761
Jain	2.0413	10	30npop50	0.9621
Wine	22.312	5	30npop 50	0.9925

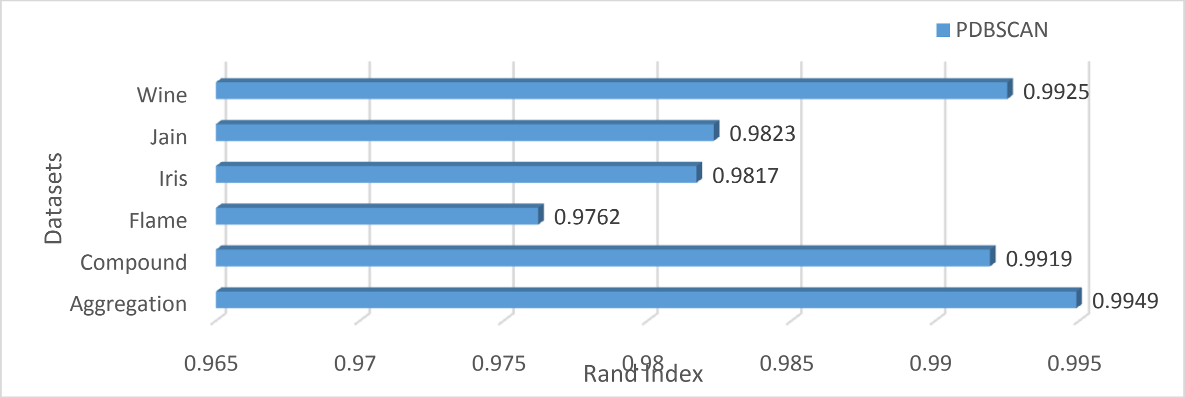

Figure 1:- Comparison of Rand index over various Datasets.

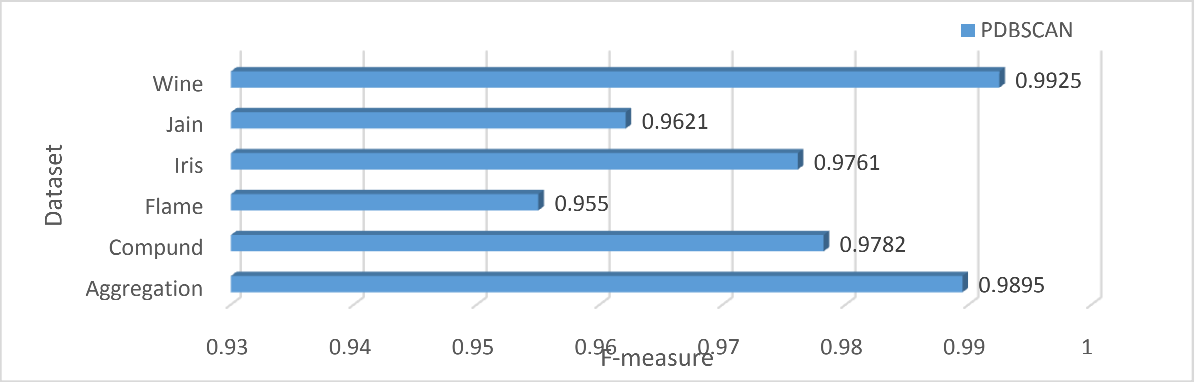

Figure2:-Comparison of F-measure Index over various different Datasets.

IV. Conclusion

In this paper we have presented the Density Based Clustering using Diffrential Evolutions. This algorithm worked in the concept of differential evolution where we tuned the parameters of the DBSCAN algorithm, here we selected the values of the eps and the minpts with the values generated by the population in the Differential Evolutions. And these values of the populations where used as input to the DBSCAN algorithm.The analysisassociatedwith the conventional DBSCAN algorithm which causes some annotations that were evidently observed in this paper. As the analysis was carried out on the basis of some very famous validity indexes which made us to come on a conclusion that the algorithm which we proposed proved efficient than the traditional DBSCAN algorithm and we got better values of the validity indexes.

References

[1]. J. Kutarnia and P. Pedersen, "A Markov random field approach to group-wise registration / mosaicing with application to ultrasound," *Med. Image Anal.*, vol. 24, no. 1, pp. 106–124, 2015.

[2]. D. Huang, J. Lai, and C. Wang, "Ensemble clustering using factor graph," *Pattern Recognit.*, vol. 50, pp. 131–142, 2016.

[3]. A. Hinneburg and D. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise," *Proc. 4th Int. Conf. Knowl. Discov. Data Min. (KDD 98)*, no. October 2001, pp. 58–65, 1998.

[4]. G. Hamerly, "Making k -means even faster," *Proc. 2010 SIAM Int. Conf. data Min. (SDM 2010)*, pp. 130–140, 2010.

[5]. X. Tang and P. Zhu, "Hierarchical Clustering Problems and Analysis of Fuzzy Proximity Relation on Granular Space," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 5, pp. 814–824, 2013.

[6]. C. Liabilities and P. Accounting, "Chapter 11," pp. 259–276, 2002.

[7]. H. Zhang and J. Lu, "Knowledge-Based Systems Semi-supervised fuzzy clustering : A kernel-based approach," *Knowledge-Based Syst.*, vol. 22, no. 6, pp. 477–481, 2009.

[8]. H. Rehioui, A. Idrissi, M. Abourezq, and F. Zegrari, "DENCLUE-IM : A New Approach for Big Data Clustering," *Procedia - Procedia Comput. Sci.*, vol. 83, no. Ant, pp. 560–567, 2016.

[9]. B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*, vol. 14. 2011.

[10]. P. Hansen and B. Jaumard, "Cluster analysis and mathematical programming," *Math. Program.*, vol. 79, pp. 191–215, 1997.

[11]. A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data," *Prentice Hall*, vol. 355. p. 320, 1988.

[12]. L. Kieu, A. Bhaskar, and E. Chung, "A modified Density-Based Scanning Algorithm with Noise for spatial travel pattern analysis from Smart Card AFC data," *Transp. Res. Part C*, vol. 58, pp. 193–207, 2015.

[13]. J. Kuan, "nearest neighbour search. for R-tree family," no. September, pp. 9–12, 1997.

[14]. M. Ester, H. Kriegel, X. Xu, and D.- Miinchen, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, pp. 226–231, 1996.

[15]. E. Bic and D. Yuret, "Locally Scaled Density Based Clustering," *Adapt. Nat. Comput. Algorithms, 8th Int. Conf. ICANNGA 2007, Warsaw, Poland, April 11-14, 2007*, pp. 1–10, 2007.

[16]. D. Birant and A. Kut, “ST-DBSCAN : An algorithm for clustering spatial – temporal data,” *Data Knowl. Eng. ,Intelligent Data Min.*, vol. 60, no. 1, pp. 208–221.

[17]. G. P. Daniel, R. Cleir, and C. Rodrigues, “VDBSCAN *: An efficient and effective spatial data mining algorithm using GPU.”

[18]. B. Borah and D. K. Bhattacharyya, “DDSC: A density differentiated spatial clustering technique,” *J. Comput.*, vol. 3, no. 2, pp. 72–79, 2008.

[19]. A. Ram, A. Sharma, A. S. Jalal, R. Singh, and A. Agrawal, “An enhanced density based spatial clustering of applications with noise,” *2009 IEEE Int. Adv. Comput. Conf. IACC 2009*, no. November, pp. 1475–1478, 2009.

[20]. H. Darong and W. Peng, “Physics Procedia Grid-based DBSCAN Algorithm with Referential Parameters,” *Phys. Procedia*, vol. 24, pp. 1166–1170, 2012.

[21]. A. Amini, H. Saboohi, T. Herawan, and T. Y. Wah, “Journal of Network and Computer Applications MuDi-Stream : A multi density clustering algorithm for evolving data stream,” *J. Netw. Comput. Appl.*, vol. 59, pp. 370–385, 2016.

[22]. J. Zhang, J. Kerekes, S. Member, A. T. Ice, and E. Satellite-, “An Adaptive Density-Based Model for Extracting Surface Returns From Photon-Counting Laser Altimeter Data,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 726–730, 2015.

[23]. M. Ankerst, M. M. Breunig, and H. Kriegel, “OPTICS : Ordering Points To Identify the Clustering Structure,” *SIGMOD ’99 Proc. 1999 ACM SIGMOD Int. Conf. Manag. data*, pp. 49–60, 1999.

[24]. J. Hou, H. Gao, and X. Li, “DSets-DBSCAN : A Parameter-Free,” vol. 25, no. 7, pp. 3182–3193, 2016.

[25]. D. Xu and Y. Tian, “A Comprehensive Survey of Clustering Algorithms,” *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, 2015.

[26]. T. Ali, S. Asghar, N. A. Sajid, and M. Ali, “Critical Analysis of DBSCAN Variations,” 2010.

[27]. K. M. Kumar and A. R. M. Reddy, “Author ’ s Accepted Manuscript,” *Pattern Recognit.*, 2016.

[28]. P. Viswanath and V. S. Babu, “Rough -DBSCAN : A fast hybrid density based clustering method for large data sets,” *Pattern Recognit. Lett.*, vol. 30, no. 16, pp. 1477–1488, 2009.

[29]. S. Mimaroglu and E. Aksehirli, “Improving DBSCAN ’ s execution time by using a pruning technique on bit vectors,” *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1572–1580, 2011.

[30]. G. Andrade, G. Ramos, and D. Madeira, “G-DBSCAN : A GPU Accelerated Algorithm for Density-based Clustering,” *Procedia Comput. Sci.*, vol. 18, pp. 369–378, 2013.

[31]. Y. Lv, T. Ma, M. Tang, J. Cao, and Y. Tian, “Neurocomputing An ef fi cient and scalable density-based clustering algorithm for datasets with complex structures,” *Neurocomputing*, vol. 171, pp. 9–22, 2016.

[32]. R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[33]. D. Zou, J. Wu, L. Gao, and S. Li, “A modified differential evolution algorithm for unconstrained optimization problems,” *Neurocomputing*, vol. 120, pp. 469–481, 2013.

[34]. and F. S. Marco Locatelli, Mirko Maischberger, “No Title,” *Differ. Evol. methods based local searches. Comput. Oper. Res. 43169–180, 2014.*, 2014.

[35]. and Y. W. Yang Lou, Junli Li, “A Binary-Differential Evolution algorithm based on Ordering of individuals,” *IEEE Nat. Comput. (ICNC), 2010 Sixth Int. Conf.*

[36]. Y. A. ; H. Chi, “Experimental Study on Differential Evolution Strategies,” *IEEE Intell. Syst. 2009. GCIS ’09. WRI Glob. Congr.*, 2009.

[37]. M. F. Ling Wang, Xiping Fu, Yunfei Mao, Muhammad Ilyas Menhas, “A novel modified binary differential evolution algorithm and its applications,” *ELSEVIER*, vol. 98, 2012.

[38]. H. Zhou, P. Wang, and H. Li, “Research on Adaptive Parameters Determination in DBSCAN Algorithm,” *J. Inf. Comput. Sci.*, vol. 7, no. July, pp. 1967–1973, 2012.

[39]. M. A. Al-Betar, A. T. Khader, Z. W. Geem, I. A. Doush, and M. A. Awadallah, “An analysis of selection methods in memory consideration for harmony search,” *Appl. Math. Comput.*, vol. 219, no. 22, pp. 10753–10767, 2013.

[40]. W. M. Rand and W. M. Rand, “Objective Criteria for the Evaluation of Clustering Methods Objective Criteria for the Evaluation of Clustering Methods,” vol. 66, no. 336, pp. 846–850, 2016.

Mr. Aakash Kulmitra,. " A Parameter free Clustering of Density Based Algortihm." IOSR Journal of Computer Engineering (IOSR-JCE) 20.5 (2018): 09-16.