Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 7 (2019) pp. 015-024

The Performance Evaluation of Data Stream Mining Algorithm with Shared density graph for Various Classification Techniques, Micro and macro clustering

S.Gopinathan[1], L.Ramesh [2]

[1] Associate Professor,
Department of Computer Science,
University of Madras,
Chennai,600005,India.
gnathans2002@gmail.com

[2] Research Scholar,
Department of Computer Science,
University of Madras,
Chennai, 600005,India
rameshnethaji2012@gmail.com

Abstract: We propose to solve the problem of micro clustering using the integration of data stream mining algorithms. Streaming data are potentially infinite sequence of incoming data at very high speed and may evolve over the time. This causes several challenges in mining large scale high speed data streams in real time. This paper discusses various challenges associated with mining data streams. Several algorithms such as data stream mining algorithms of accuracy and micro clustering are specified along with their key features and significance. The significant performance evaluation of micro and macro clustering relevant in streaming data of shared density graph and clustering are explained and their comparative significance is discussed. The paper illustrates various streaming data computation platforms that are developed and discusses each of them chronologically along with their major capabilities. The performance and analysis are different radius activation functions and various number of radius applied to an data stream mining algorithm with shared density graph for micro clustering and macro clustering. To find the running time of each classification algorithm time clock is used for comparing the running time (in seconds) and calculated the average running time for each classification algorithm.

***Keywords*: stream mining, micro clustering, shared density graph, macro clustering, running time, classification.**

I. Introduction

The data streams are in the real time. They are very high speed and very evolve over the time [1]. Mining of these large scale data streams to perform some kind of machine learning or futuristic predictions regarding data instances have drawn a significant attention of researchers in couple of previous years. The data streams resemble the real time incoming data sequence very well. The source of these data streams can be collected from various sensors situated in medical domain to monitor health conditions of patients, in industrial domain to monitor manufactured products and other sources are to collect from streams on social networking.

E-commerce sites twitter posts, various blogs, web logs, and many more [2-3]. The above mentioned sources not only produce data streams, but they produce them in huge amount (of scale of tera bytes to peta bytes) and at rapid speed. Now, mining huge data in real time raises various challenges and has become the hot area of research recently. These challenges include memory limitation, faster computing requirement *etc.* Apart from these challenges, streaming data has inherent nature of evolution that means that concepts that are being mined evolve and change over the time [4-5]. It makes the traditional data mining algorithms and techniques incapable of appropriately handling data streams and yields the requirement of algorithms suitable for streaming data mining. This may be achieved in two ways; either modify the existing data mining algorithms to make them suitable for stream mining or create new streaming data mining algorithms right from the scratch. Another aspect of this field is the evaluation of the performance of the stream data mining algorithms. Since the performance evaluation is done continuously throughout the mining task and on

partial read data streams, it becomes critical to use suitable performance measures in reference to streaming data mining. Various new performance evaluators have been devised specifically for this purpose [6-8].

II. Related Work

Bakshi and Sonali et al to discuss the stream data mining, Agarwal challenges associated in mining potentially infinite data streams along with various stream mining algorithms for classification and clustering.[3]

Sibson R to discuss The Single-Link method is a commonly used hierarchical clustering method starting with the clustering obtained by placing every object in a unique cluster, in every step the two closest clusters in the current clustering are merged until all points are in one cluster. Other algorithms which in principle produce the same hierarchical structure have also been suggestion. [19]

Hinneburg A., Keim D et al to discuss the density-based algorithm DenClue is proposed. This algorithm uses a grid but is very efficient because it only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure. This algorithm generalizes some other clustering approach'es which, however, results in a large number of input parameters. Also the density- and grid-based clustering technique CLIQUE.[20]

Jain A. K., Dubes R. C et al to discuss the common way to find regions of high-density in the data space is based on grid cell densities. A histogram is constructed by partitioning the data space into a number of non-over lapping regions or cells. Cells containing a relatively large number of objects are potential cluster centers and the boundaries between clusters fall in the "valleys" of the histogram. The success of this method depends on the size of the cells which must be specified by the user. Cells of small volume will give a very "noisy" estimate of the density, whereas large cells tend to overly smooth the density estimate.[21]

Schikuta E et al to discuss the Hierarchical clustering is based on the clustering properties of spatial index structures. The GRID and the BANG clustering apply the same basic algorithm to the data pages of different spatial index structures. A clustering is generated by a clever arrangement of the data pages with respect to their point density. This approach is not well suited for high-dimensional data sets because it is based on the affectivity of these structures as spatial access methods.[22]

B. Brian, M. Datar and R. Motwani et al to discuss the algorithm of Hoeffding tree. Hoeffding trees a decisiontree learning method. Hoeffding trees can be learned in constant time per example while being nearly identical to the trees a conventional batch learner would produce.[10]

Prasad, B. R., Agarwal S et al to discuss the Fast decision tree (FDTA) which is an improvement classification method based on Hoeffding tree. The t is an algorithm parameter has shown an effect on a decision regarding making a split process on the tree or not. So t is generated in order fashion in domain (0-1) instead of treated as a fixed value as known in traditional Hoeffding tree (t known as 0.05). According to the obtained results above, it shows that FDTA is gained highest accuracy than Hoeffding tree, the same things regarding memory space and execution time.[15]

Cao, Feng, Martin Ester et al to discuss the DenStream, an effective and efficient method for clustering an evolving data stream. The method can discover clusters of arbitrary shape in data streams, and it is insensitive to noise. The structures of p-micro-clusters and o-microclusters maintain sufficient information for clustering, and a novel pruning strategy is designed to limit the memory consumption with precision guarantee.[17].

Bakshi Rohit et al to discuss the specifies the need of new algorithms and evaluation measures relevant to this field and mentioned some of them used in stream mining scenario. The various available tools or platform to provide the appropriate framework to deal with large scale data streams along with their key features have also been described in chronical order that helped in understanding the evolvement of the streaming data computing and mining platforms.[18].

Chen, Yixin, and Li Tu construct an algorithm D-Stream for clustering stream data using a density-based approach. The algorithm uses an online component which defines every input data into a grid an offline component which computes the grid density and clusters the grids depends on the density of the cluster. The algorithm can find Clusters of arbitrary shape. The researchers compare the qualities of the clustering results by D-Stream and those by CluStream.[23]

III.BLOCK DIAGRAM OF PROPOSED SYSTEM

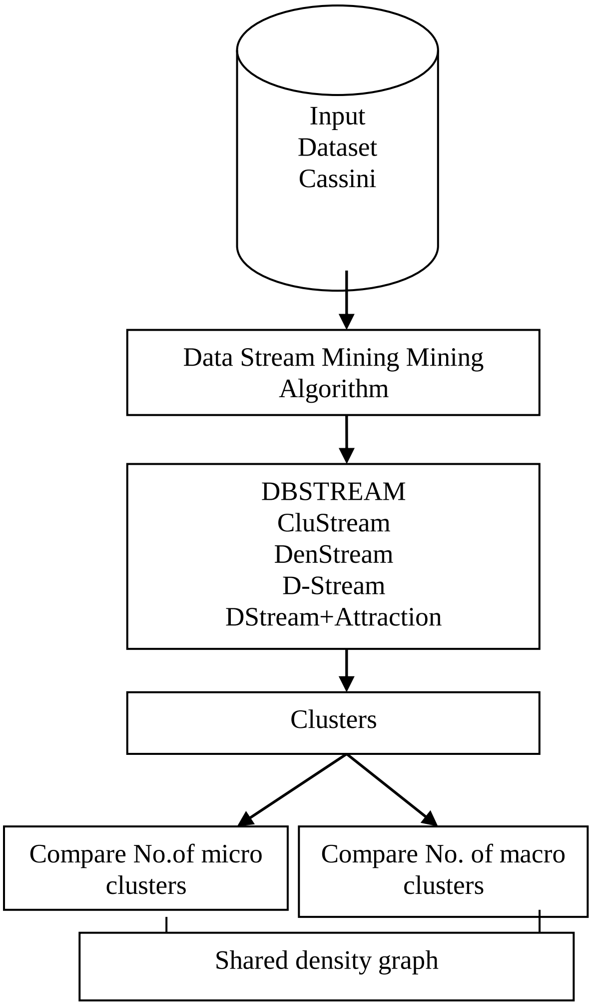

Fig.1.Block diagram of proposed system

The Block diagram explains the proposed method for the Research work. In this frame work for Performance Evaluation of Data Stream Mining Algorithm with Shared density graph for Micro and macro clustering are discussed and explained.

In this framework, the input dataset is feed into the data stream mining algorithm where the DBSTREAM, CluStream,DenStream, D-Stream,D-Stream+Attraction is done according to the given dataset,and it will takes the method which gives the different radius like radius level r=0.01,0.02,0.03,0.04,0.05 to compare the Number of micro clusters and Number of macro clusters and using shared density level is 5 cm.

IV. Algorithm for Proposed System

BEGIN PROPOSED ALGORITHM ( )

1. Initialization
   Input: Read the dataset
   Data: Cassini

END ALGORITHM

BEGIN DATA STREAM MINING ALGORITHM

1.DBStream,
2.Clustream,
3.Denstream,
4.D-Stream
5.DStream +Attraction

END DATA STREAM MINING ALGORITHM

BEGIN START RADIUS FIXATION(radius)

1. Radius fixation for all data stream mining Algorithm like r=0.8, 0.6,0.4, 0.2, 0.1
2. No.of Micro clusters(s)-DBSTREAM without shared density
3. No.of Micro clusters(t)-D-Stream
4. No.of Micro clusters(u)-D-Stream+Attraction
5. No.of Micro clusters(v)-DenStream
6. No.of Micro clusters(w)-CluStream

Set radius of current data stream mining algorithm

END START RADIUS FIXATION

BEGIN UPDATE ON ALL THE DATA STREAM MINING ALGORITHM

1. Compare Number of Micro clusters
2. Compare Number of macro clusters
3. Plot for micro and macro clustering of graph presentation
4. Shared density graph Using Macro cluster and macro clustering
5. Calculate the different running time with the data stream mining Algorithm

End begin

END UPDATE ON DATA STREAM MINING ALGORITHM

V. Experiments and Results

In the experiment part, the performance of the proposed work is implemented with data stream mining algorithm publically available R extension called stream. Our Research work used real data set called CASSINI developed number of data stream mining algorithm and MOA(Massive Online Analysis)used.

Our proposed Algorithm work with best result compare with different method like Michael Hahsler method ,C.Aggarwal method. They are two comparison between micro and macro clusters, like figure 1 show block diagram of proposed work, figure 2 show Cassini data set for micro
Clustering. In the Table 1, 2, 3 and 4 show the comparison with existing method[3], with best accuracy comparison like radius level of r=0.01, 0.02 , 0.03, 0.04, 0.05, to applied all the data stream mining algorithm. In all data stream mining algorithm result accuracy is highlighted in bold. In table 1 DBSTREAM mining Algorithm and figure 3 show that the radius level of 0.01, 0.02, 0.03, 0.04, 0.05 are given there r=0.01 given best micro cluster it show in figure 3. In table 2 DENSTREAM mining Algorithm and figure 4 show that the radius level of 0.01, 0.02, 0.03, 0.04, 0.05 are given their r=0.01 given best micro cluster it show in figure 4. In table 3 CluSTREAM mining Algorithm and figure 3 show that the radius level of 0.01, 0.02, 0.03, 0.04, 0.05 are given their r=0.01 given best micro cluster it show in figure 5. In table 4 D-STREAM mining Algorithm and figure 3 show that the radius level of 0.01, 0.02, 0.03, 0.04, 0.05 are given there r=0.01 given best micro cluster it show in figure 6. In figure,7,8,9,10 and 11 shows the shared density graph for data stream mining algorithm micro and macro cluster.

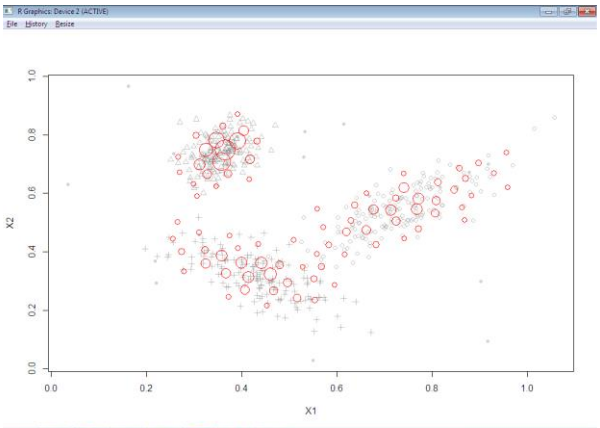

Figure 2: Cassini data set for micro clustering

DBSTREAM Stream mining Algorithm

Sl.No.	Radius level(r)	No.of micro clusters	No.of macro clusters
1	0.01	**52**	22
2	0.02	117	9
3	0.03	95	5
4	0.04	83	**2**
5	0.05	68	**2**

Table 1: DBSTREAM micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, 0.05.

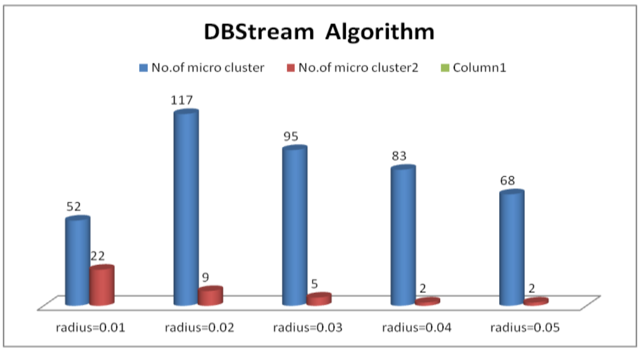

Figure 3: DBSTREAM micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, 0.05.

DENSTREAM Stream mining Algorithm

Sl.No.	Radius level(r)	No.of micro clusters	No.of macro clusters
1	0.01	**26**	2
2	0.02	28	3
3	0.03	37	**1**
4	0.04	60	**1**
5	0.05	47	**1**

Table 2: DEN Stream micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, 0.05

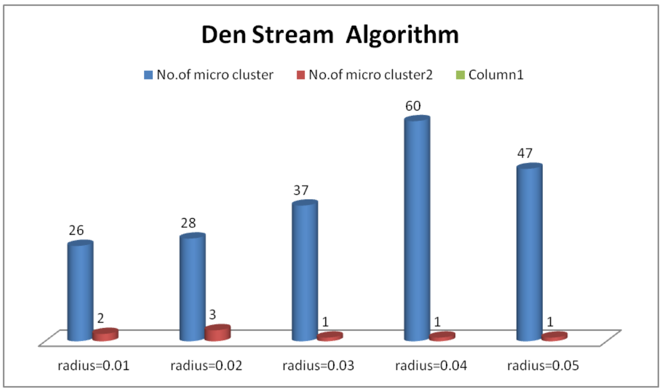

Figure 4: DEN Stream micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, 0.05

Clu STREAM- Stream mining Algorithm

Sl.No.	Radius level(r)	No.of micro clusters	No.of macro clusters
1	0.01	**38**	**1**
2	0.02	43	2
3	0.03	49	**1**
4	0.04	72	2
5	0.05	50	3

Table 3: Clu STREAM micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, 0.05.

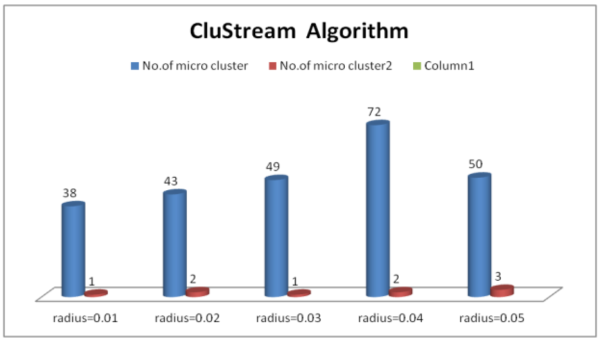

Figure 5: Clu STREAM micro and macro cluster with r=0.01, 0.02, 0.03, 0.04,0.05

D-STREAM -Stream mining Algorithm

Sl.No.	Radius level(r)	No.of micro clusters	No.of macro clusters
1	0.01	**33**	4
2	0.02	41	5
3	0.03	43	**3**
4	0.04	39	4
5	0.05	45	6

Table 4: D-STREAM micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, 0.05.

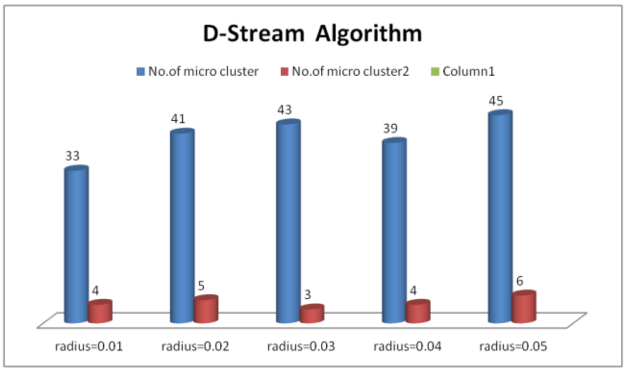

Figure 6: D-STREAM micro and macro cluster with r=0.01, 0.02, 0.03, 0.04, and 0.05.

Shared density graph

Micro-cluster is the share density with each other in a typical clustering. This leads to a very sparse shared density graph. This fact can be exploited for more efficient storage and manipulation of the graph. We represent the sparse graph by a weighted adjacency list in all tha data stream mining algorithms. Furthermore, during clustering we already find all fixed-radius nearest-neighbors.Therefore, obtaining shared weights does not incur any additional increase in searchTime.

Shared density graph for DBStream

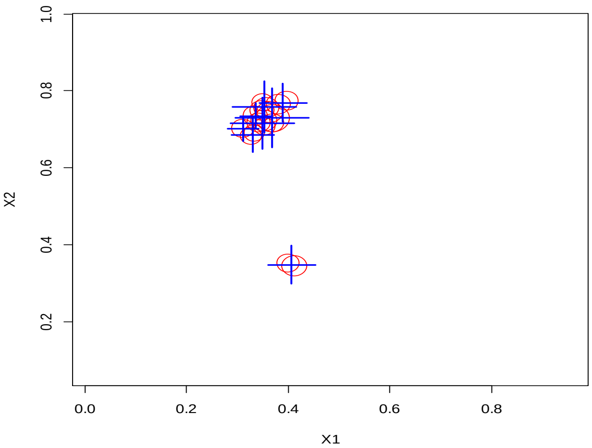

Fig 7:Shared density graph for DBStream Shared density graph=8

Shared density graph for CluStream

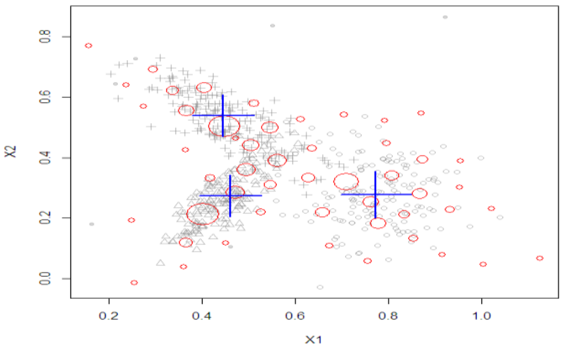

Fig 8: Shared density graph for CluStream Shared density graph=3

Shared density graph for DenStream

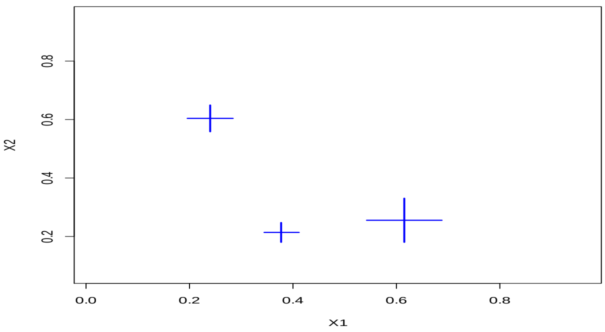

Fig 9: Shared density graph for DenStream Shared density graph=3 Shared density graph for D-Stream

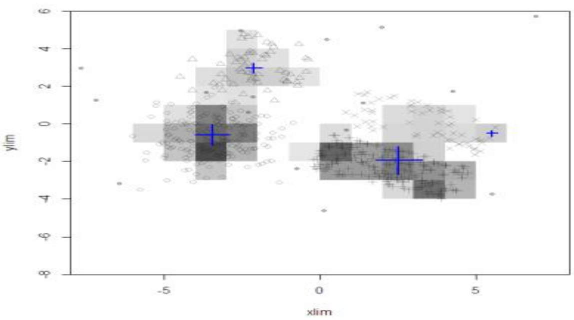

Fig 10: Shared density graph for D-Stream shared density graph=4

Shared density graph for D-Stream + Attraction

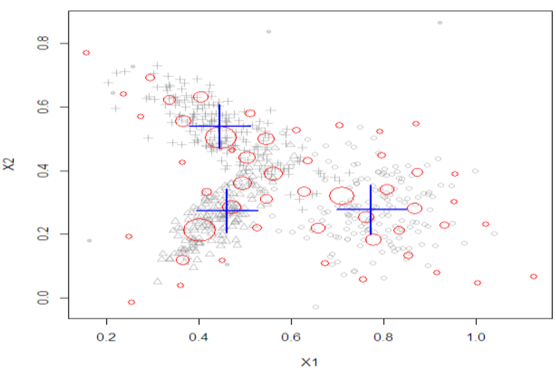

Fig 11: Shared density graph for D-Stream + Attraction shared density graph=3

Sl.No.	Data stream Mining Algorithm	No.of shared density graph
1	DBSTREAM without shared density	8
2	D-Stream	4
3	D-Stream + attraction	3
4	DenStream	3
5	CluStream	3

Table 5: Data Stream Mining algorithms with Shared density graph

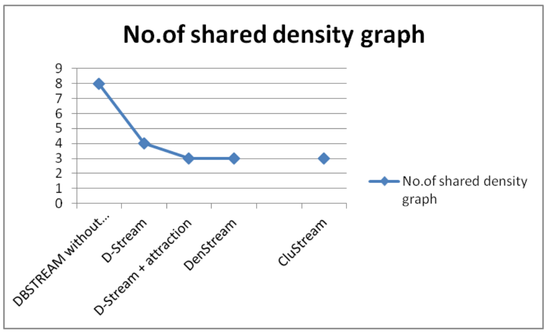

Fig 12:Shared density graph for Data stream mining Algorithm

Data set attributes details

S.No	Attributes Name
1	Pregnancies
2	Glucose
3	BloodPressure
4	SkinThickness
5	Insulin
6	BMI
7	DiabetesPedigreeFunction
8	Age
9	Outcome

Table 6: Dataset attributes details for classification techniques

Preprocess of dataset visualization

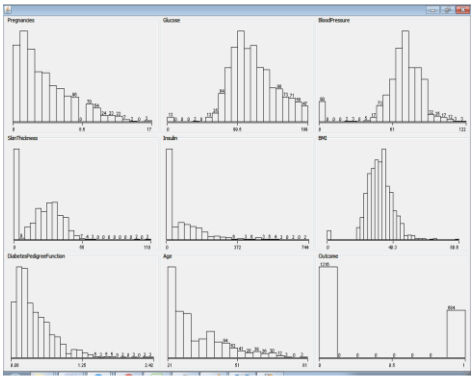

Fig 13: Pre-process of dataset visualization

VI. Classification Techniques

Linear regressions

We further used classification method for improving our work In statistics, **linear regression** is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quintile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine[32].

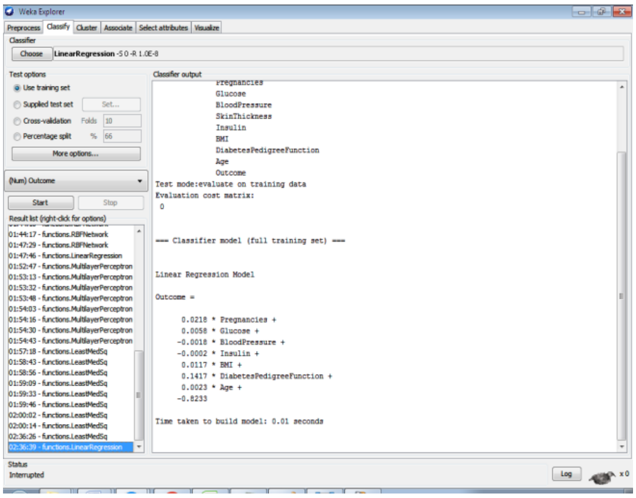

Fig 14: Linear Regression classifier

RBF Network Classifier

In the field of mathematical modeling, a **radial basis function network** is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. Radial basis function networks have many uses, including function approximation, time series prediction, classification, and system control[33].

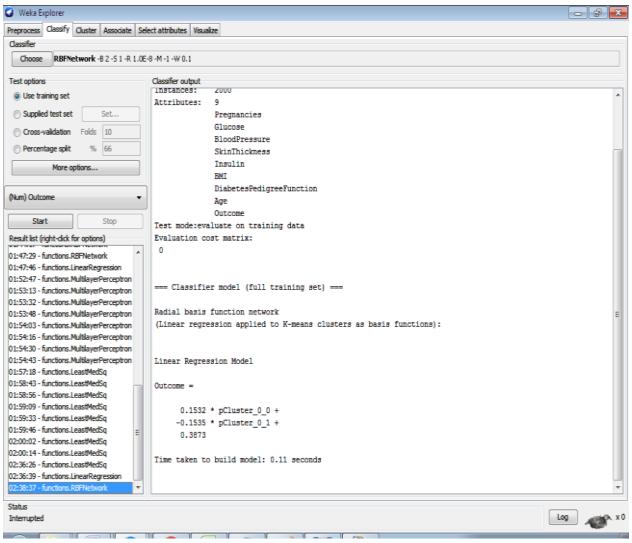

Fig 15: RBF Network classifier

Function approximation

If the purpose is not to perform strict interpolation but instead more general function approximation or classification the optimization is somewhat more complex because there is no obvious choice for the centers. The training is typically done in two phases first fixing the width and centers and then the weights. This can be justified by considering the different nature of the non-linear hidden neurons versus the linear output neuron.

Training the basis function centers

Basis function centers can be randomly sampled among the input instances or obtained by Orthogonal Least Square Learning Algorithm or found by clustering the samples and choosing the cluster means as the centers. The RBF widths are usually all fixed to same value which is proportional to the maximum distance between the chosen centers.

Gradient descent training of the linear weights

Another possible training algorithm is gradient descent. In gradient descent training, the weights are adjusted at each time step by moving them in a direction opposite from the gradient of the objective function (thus allowing the minimum of the objective function to be found.

Multilayer perception Layer

"MLP" is not to be confused with "NLP", which refers to processing. A **multilayer perception** (MLP) is a class of feed forward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perception. It can distinguish data that is not linearly separable. Multilayer perceptions are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer[7].

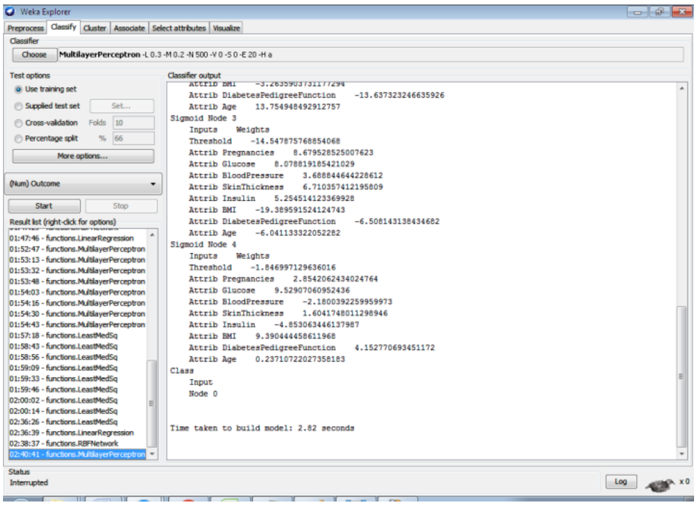

Fig 16: Multilayer Perception classifier

Terminology

The term "multilayer perceptron" does not refer to a single perceptron that has multiple layers. Rather, it contains many perceptrons that are organized into layers. An alternative is "multilayer perceptron network". Moreover, MLP "perceptrons" are not perceptrons in the strictest possible sense. True perceptrons are formally a special case of artificial neurons that use a threshold activation function such as the Heaviside step function. MLP perceptrons can employ arbitrary activation functions. A true perceptron performs binary classification (either this or that), an MLP neuron is free to either perform classification or regression, depending upon its activation function.

The term "multilayer perceptron" later was applied without respect to nature of the nodes/layers, which can be composed of arbitrarily defined artificial neurons, and not perceptrons specifically. This interpretation avoids the loosening of the definition of "perceptron" to mean an artificial neuron in general.

Applications

MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.

MLPs are universal function approximates as showed by Cybenko's theorem, so they can be used to create mathematical models by regression analysis. As classification is a particular case of regression when the response variable is categorical, MLPs make good classifier algorithms.

finding applications in diverse fields such as speech recognition, image recognition, and machine translation software, but thereafter faced strong competition from much simpler and related support vector machines. Interest in back propagation networks returned due to the successes of deep learning[32].

Algorithm	Use training set	Instances	Attributes
Linear Regression	0.01 sec	2000	9
Multilayer Perception	30. 82 sec	2000	9
Leastmedsq Classification	24.01 sec	2000	9
RBF Network classifier	0.42 sec	2000	9

Table: 7 Time complexity for classification Algorithm

It showed in figure 14 show the Linear Regression classifier result, figure 15 RBF show Network classifier, figure 16 show the Multilayer perception layer classifier. It shows the Tables like Table 7 Time complexity for classification Algorithm. There are four classification Algorithm compared. To find the running time of each classification algorithm time clock is used for comparing the running time (in seconds) and calculated the average running time for each classification algorithm and then showed the result with the help of a chart. From the above chart Linear Regression is the most efficient classification algorithm based on the running time.

VII. Conclusion

In the research paper we study the problem of micro clustering using the integration of data stream mining algorithm with shared density graph for cm=5, five different radius functions like r=0.01, 0.02, 0.03, 0.04, 0.05 are input to Cassini dataset. It is found that after Number of micro and macro cluster are compared with the shared density graph functions. Further, we conclude the best micro and macro clustering in the entire data stream mining algorithm mentioned highlighted bold. In this paper various classification algorithms and their comparison are discussed. To find the running time of each classification algorithm time clock is used for comparing the running time (in seconds) and calculated the average running time for each classification algorithm and then showed the result with the help of a chart and figure. From the above chart Linear Regression is the most efficient classification algorithm based on the running time.

References

[1] J. Han,M.Kamber and J. Pei, Data Mining: Concepts and Techniques, 3rd edition, MorganKaufmann, **(2011)**.

[2] J. Gama, Knowledge discovery from data streams, Chapman & Hall CRC, **(2010)**.

[3] S. Agarwal, and B. R. Prasad, High speed streaming data analysis of web generated log streams, In 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS), IEEE, **(2015)**, pp. 413-418.

[4] D. Kifer, S. B. David and J. Gehrke, Detecting

[5] Change in Data Streams. VLDB Conference, **(2004)**.

[6] P. Kranen, H. Kremer, T. Jansen, T. Seidl, A. Bifet, G. Holmes and B. Pfahringer, Clustering Performance on Evolving Data Streams: Assessing Algorithms and Evaluation Measures within MOA, IEEE International Conference on Data Mining - ICDM, **(2010)**, pp. 1400-1403.

[7] A. Bifet, Pitfalls in Benchmarking Data Stream Classification and How to avoid them. Machine Learning and Knowledge Discovery in Databases, pp. 465-479. Springer Berlin Heidelberg (2013)

[8] M. J. Song and L. Zhang, Comparison of cluster representations from partial second- to full fourth-order cross moments for data stream clustering, in ICDM, **(2008)**, pp. 560–569.

[9] K. Philipp, Clustering performance on evolving data streams: Assessing algorithms and evaluation measures within MOA, Data

Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE, (2010).
[10] J. A. Daniel, Aurora: a new model and architecture for data stream managemen, The VLDB Journal The International Journal on Very Large Data Bases, vol. 12, no. 2, **(2003)**, pp. 120-139.
[11] B. Brian, M. Datar and R. Motwani, Load shedding for aggregation queries over data streams." Data Engineering, 2004‖, Proceedings. 20th International Conference on. IEEE, **(2004)**.
[12] D. J. Abadi, The design of the borealis
[13] stream processing engine, in Proceedings of CIDR, 2005.
[14] Wabbit, V. 2007. http://hunch.net/vw
[15] Neumeyer, L., Robbins, B., Nair, A., Kesari, A. 2010. S4: Distributed Stream Computing Platform. In Proc. ICDMW, IEEE Press, 170-177.
[16] Storm, 2011. http://storm-project.net.
[17] Prasad, B. R., Agarwal, S.: Handling Big Data Stream Analytics using SAMOA Framework - A Practical Experience. Int. J. Database Theory and Application. 7, 4, 197-208 (2014)
[18] Bifet, Albert. "Mining Big Data in Real Time", Informatica37, pp:15-20, 2013.
[19] Cao, Feng, Martin Ester, Weining Qian, and Aoying Zhou. "Density-Based Clustering over an Evolving Data Stream with Noise." In SDM, vol. 6, pp. 328-339. 2006.
[20] Bakshi Rohit and Sonali Agarwal "Stream datamiing:Platforms,Algorithms,Performance Evaluatorsand Research Trends "International Journal of Database Theory and Application,Vol-9,(2016),pp-201-218.
[21] Sibson R."SLINK: an optimally efficient algorithm Core the single-link cluster method". The Comp. Journal, Vol. 116, No. 1, 1973, pp. 30-34
[22] Hinneburg A., Keim D.: "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", Proc. 4th Int. Conf. on Knowledge Discovery & Data Mining, New York City, NY, 1998.
[23] Chen, Yixin, and Li Tu. "Density-based clustering for real-time stream data." In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 133-142. ACM, 2007.
[24] Rehm, Frank, Frank Klawonn, and Rudolf Kruse. "A novel approach to noise clustering for outlier detection." Soft Computing 11, no. 5 (2007): 489-494.
[25] Elahi, Manzoor, Kun Li, Wasif Nisar, Xinjie Lv, and Hongan Wang. "Efficient clustering-based outlier detection algorithm for dynamic data stream." In Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08. Fifth International Conference on, vol. 5, pp. 298-304. IEEE, 2008.
[26] Pamula, Rajendra, Jatindra Kumar Deka, and Sukumar Nandi. "An outlier detection method based on clustering." In Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on, pp. 253-256. IEEE, 2011.
[27] Singh Vijendra.Efficient Clustering For High Dimensional Data: Subspace Based Clustering and Density Based Clustering, Information Technology Journal; 2011, 10(6), pp. 1092-1105.
[28] DBreiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. "Classification and Regression Trees". Wadsworth International Group. Belmont, CA: The Wadsworth Statistics/Probability Series1984
[29] Quinlan, J. R. Simplifying "Decision Trees. International Journal of Man-Machine Studies" ;1987, 27:pp. 221-234.
[30] Langley, P. "Induction of Recursive Bayesian Classifiers". In Brazdil P.B. (ed.), Machine Learning: ECML-93;1993, pp. 153-164. Springer,Berlin/Heidelberglew York/Tokyo.
[31] Witten, I. & Frank, E,"Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.ch. 3,4, pp 45-100.
[32] S.Gopinathan and ,M.Pandiyan ,"The Novel Clustering Approach using Extreme Learning Machine with K- Means++ Technique ", International Journal of Innovation and Advancement of Computer Science(ijiacs) (A peer-reviewed journal),volume – 6, issue-11, November 2017, pp-548-551, ISSN-2347-8616.
[33] S.Gopinathan,M.Pandiyan and P.Thangavel , "The New Clustering approach Using Extreme Learning Machine with K-Medoids Techniques", International Journal of Computational Intelligence Research, volume – 13, issue-7, November-2017, pp-1669-1678,ISSN - 0973-1873.

Author Biographies

Dr.S.Gopinathan working as an Associate Professor in the Department of Computer Science, University of Madras, Chennai, India. He has 19 years of teaching experience for post graduate in the field of Computer Science and Research. He has published number of papers. He has produced 11 M. Phil Scholars in the Computer Science, 8 PhD Research Scholars are registered under him. He also has been serving as a panel member for various competitive examinations and universities. His interested area of research is Image Processing, Software Engineering, Data Mining.

L.Ramesh is currently is a Ph.D., Research scholar in the Department of Computer Science, University of Madras. He is Completed Master degree and M.Phil., in Computer Science from University of Madras, Chennai. His area of interest includes Data Mining, Neural Networks.