Load Shedding fo̸r Window Aggregation Queries over Sensor Streams Management

S.Senthamilarasu[1], M.Hemalatha[2]

ABSTRACT

In today information society is becoming a knowledge intensive; mining of knowledge is becoming a very significant task for numerous people. The main issue in stream mining is handling a huge amount of data from wireless sensor Network (WSN) is to deliver rapidly which makes it infeasible to store everything in active storage. To overcome this problem of handling voluminous data we exposed a novel load shedding system using window based aggregate function of the data stream in which we accept those tuples in the stream that meet a criterion. Accepted tuples are conceded to another process as a stream, while further tuples are dropped. This proposed model conceivably segregates the data input stream into windows and probabilistically decides which tuple to drop based on the window function. The result shows that the best window aggregate function used for dropping tuples is identified with the three prediction models used in data ,from these model naïve Bayes shows Better result than the other two prediction models.

Keywords: — Data stream mining, Windows functions, Load Shedding Scheme.

1. INTRODUCTION

A wireless sensor network (WSN) is a special kind of network that has the capacity of sensing and processing, with consumption [25] and delay. While transmitting data through the WSN we can face some problem such as data quality, data reduction and its losses and benefits. There are two main types of applications for WSNs: actuating and monitoring applications. In monitoring applications, the sensing data processed only by the sensor nodes. In actuating applications, nodes can interfere in the monitored environment [27], [28]. In both circumstances, we can apply data stream techniques in the sensor stream for monitoring and we can compose stream queries in actuating case. These applications are concerned with how queries can be answered [29], [30], [31]. Each year, a huge amount of new information is formed. The enormous amount of collected data is increasing with the progression of the fashionable data technology that produces every year and a lot of powerful computers that enable to gather, store, and combine huge amounts of data at a terribly low cost. The progress in database capability has influenced many areas in human's life, from supermarket transaction and credit card usage records in molecular and medical databases[34]. Sensor networks offer a new source of massive, continuous streams of data that may be fed into management systems such as temperature monitoring, precision agriculture, and urban traffic control management. However these largescale networks have additionally to contend with the noise, uncertainty, and asynchrony of the real-world data [35].The complex nature of sensor data has conjointly increased the difficulties and challenges of data mining in sensor networks in terms of data processing, data storage, and model storage necessities[36].

1. Research Scholar, Computer Science, Karpagam University, Coimbatore - 641021, India, s.a6asu@gmail.com
2. Head & Professor, Software Systems, Karpagam University, Coimbatore-641021, India, Hema.bioinf@gmail.com

Recent trends in pervasive computing in a data stream management system (DSMS), together with new (wearable) technology as sensors, and wearable computers strongly support novel kinds of applications [1] such as the military. Agriculture. Medical. Security systems and etc. DSMSs are effective tools for building sensing applications. DSMSs greatly simplify the development of monitoring applications because developers or end-users only need to declaratively express the events they are interested in monitoring and the DSMS takes care of the rest[13]. In general, the area of sensor data stream management and processing is a very challenging but also a timely one for several reasons. Online monitoring or network monitoring domains are which rely on the presence of (hardware and/or software) sensors. These sensors turnout vast amounts of data that must be processed, analyzed and managed online and during a reliable manner. Thus, having an appropriate infrastructure for telemonitoring is crucial. Sensor data and data stream management also greatly affects data which we are collected from source. Data generated by processing streams of sensor information (e.g., after aggregating data over a certain time window or outliers with special semantics which have been detected in a stream) has to be added to a storage. Therefore, the infrastructure for data stream management should consider i.)operators which contend with continuous streams of data and ii.) Discrete operators/services which permit for the interaction with the data stream management [1].

Data stream management systems could be processed the higher input rates and it perform with their available system resources (e.g., CPU, memory). When input rates exceed the resource capacity the system becomes overloaded. So, we get query answers are delayed. Load shedding could be a technique to get rid of excess load from the system in order to keep query processing up with the input arrival rates. As a result of load shedding, the system delivers approximate query answers with reduced latency[10].

The Major Challenges of data stream, the data generation rates might vary some data sources become faster than ever before. This rapid generation of a continuous stream of information has challenging our storage capacity and communication capabilities of the computing system [17]. Another one, cluster validity has high tended the need for determining apposite criteria to validate results. New challenges of Data stream is so far as adaptability becomes more tricky to find what data stream contains noise. It shares the foremost of the difficulties with stream query processing. Discovering the patterns are hidden and much more general than querying and data stream is ability to permanently maintain the accurate decision model. The recent trends of researches within the data stream encompass problem such as understanding climate change and its impacts, electric grid monitoring, disaster preparedness and management, national or homeland security, and the management of critical infrastructures[36] .Data streams generated from sensors and other wireless data sources create a real challenge to transfer these huge amounts of data elements to a central server to be analyzed[16].

II. Related Work

Rather than trying to cover the large body of interesting previous work on data streams, in data stream applications, the unpredicted fluctuation of the arrival rate along with continuous processing of posts queries; is one of the main problems that may result in an overloaded system. Prioritized query shedding technique that handles the overloading problem when considering queries' priorities.Unlike most shedding techniques that assume all queries are equally important, or even prioritized shedding techniques that are based on dropping input tuples according to the regions' priorities, this technique considers the priority of a query as a whole[3].

Zhang Longbo , Many data stream sources are prone to dramatic spikes in volume, and data items arrive in a bursting fashion. Loading and processing all the arrived data items will exceed memory availability. It becomes essential to shed load by dropping some fraction of the unprocessed data items during a spike. The load shedding strategy is to partition the domain of the join attribute into certain subdomains, and filter out certain input tuples based on their join values by maintaining simple data stream statistics [10]. Analyze the behavior of the sketch estimator when computed over a sample of the stream, not the entire data stream, from the size of the join and the self-join size problems. Based on analysis is developed a generic sampling process and instantiate the results of the analysis for all three major types of sampling - Bernoulli sampling which is used for load shedding, sampling with replacement which is used to generate i.i.d. Samples from a distribution, and sampling without replacement which is used by online aggregation engines[4].

The Novel feedback control-based load shedding scheme for data stream processing is to identify system identification to establish a dynamic model to describe a data stream management system (DSMS), which enables us to analyze DSMS quantitatively [5].

To deal with resource constraints by shedding load in the form of dropping tuples from the data streams. Defining the problem space by discussing architectural models for data stream join processing and surveying suitable measures for the quality of an approximation of a set-valued query result and examine in detail a large part of this problem space[6].

Kuen-fang Jea, load Controlled mining system with an Edeficient mining is decided to execute to preserve a fraction of unprocessed data [7].Chao-Wei Li, deals with the overload handling for frequent-pattern mining in online data streams, to deal with the frequent itemsets which need to be enumerated and counted by the mining. Therefore, load shedding scheme involves the maintenance of a smaller set of item sets, so the workload can be conical accordingly [8].Babcock, B, Data streams is often bursty and data characteristics may vary over time. So we focus on aggregation queries that determine at what points in a query plan load shedding should be performed and what amount of load should be shed at each point in order to minimize the degree of inaccuracy introduced into query answer[9].

III. Problem Art

Conceptually, the load should shed whenever Load (Q(I)) > C. The load will be discarded at any point in the query plan. Dropping a load at earlier points avoids wasting work; But, as a result of shared operators in the query plan, an early drop may adversely have an affect on the accuracy of too many query answers. Just enough of the load at the chosen point(s) in the query plan should be shed so that the whole resource demand gets below the obtainable capability with minimal total loss in accuracy. The data items to be discarded got to be chosen based on the approximation model and also the properties of the operators in the query plan.

IV. Dataset Description

The data set used in this paper is from CHART[] which is a joint effort of the Maryland Department of Transportation, Maryland Transportation Authority and the Maryland State Police, in cooperation with other federal, state and local agencies. CHART's mission is to improve "real-time" operations of Maryland's highway system through teamwork and technology. They sponsored several real time data streams in their website[14]. In this paper we have used Traffic Speed Data and this data will automatically refresh every five minutes. The dataset consist of average speed of the vehicle crossed over. The attributes presented are Location, Average Speed and Last Reported. Using this dataset we are performing load shedding based on window based aggregate function.

V. CLASSIFICATION MODELS

5.1Decision Tree

In statistics, data mining and machine learning, uses a decision tree as a predictive model which maps annotations about an entry to conclusions about the entries target value[15]. Decision tree is a classifier in the form of a tree structure – Decision node: specifies a test on a single attribute – Leaf node: indicates the value of the target attribute

– Arc/edge: split of one attribute

– Path: a disjunction of test to make the final decision

5.2 Logistic Regression

In statistics, **logistic regression** is a type of regression analysis used for predicting the outcome of a categorical (a variable that can take on a limited number of categories) criterion variable based on one or more predictor variables. The probabilities describing the possible outcome of a single trial are modelled, as a function of explanatory variables, using a logistic function. In linear regression analysis, one is concerned with partitioning variance via the sum of squares calculations - variance in the criterion is essentially divided into variance accounted for by the predictors and residual variance.

5.3Naive Bayes

The Naive Bayes Classifier technique is based on the Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods[18]. An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters necessary for classification. Because independent variables are assumed only the variances of the variables for each class need to be determined, not the entire covariance matrix.

VI. PROPOSED METHODOLOGY

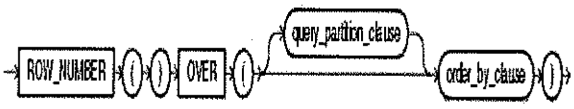
The proposed algorithm has two steps.

- Decide for each query the effectual sampling rates in such a way that will distribute error consistently along with all queries.
- In the data flow diagram find out where load shedding should be performed to attain the suitable rates and satisfy the load equation

The major goal of this proposed work is to overcome the problem of overloading in the datastream. The load shedding process is carried out by dropping tuples based on the four different window aggregate function[19] rank(), row_rank() , dense_rank() and cumulative distance. The incoming data stream is partitioned based on these functions and the best ones are considered for further processing. The resultant dataset is further applied to the prediction processing tocheck the quality of datasets using three different prediction model decision tree, naïve bayes and logistic regression.

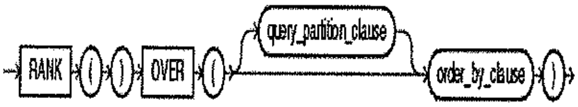
The queries used for dropping tuples are as follows:

row_number() – number of the current row within its partition, counting from 1.



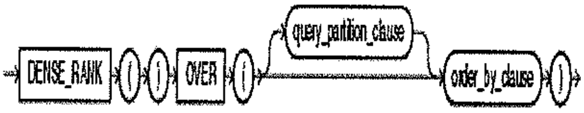
ROW_NUMBER is a logical function. It assigns a distinctive number to each row to which it is functional (either each row in the partition or each row returned by the query), in the ordered sequence of rows specified in the order_by_clause, beginning with 1.

rank() - rank of the current row with gaps;same as row_number of its first peer



This query returns the rank of all rows within the sliding window of a result set. The rank of a row is one plus the number of ranks that come previous to the row in question.

dense_rank() - rank of the current row without gaps; this function counts peer groups

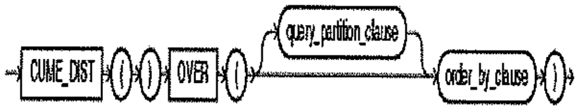


This method returns the rank of rows inside the partition of a result set, without any gaps in the ranking. The rank of a row is one plus the number of distinctive ranks that come before the row in query

As an logical function, DENSE_RANK calculates the rank of each row returned from a query with respect to the further rows, based on the values of the value_exprs in the order_by_clause.

cume_dist() - relative rank of the current row:(number of rows preceding or peer with current row) / (total rows)

CUME_DIST(CD) computes the cumulative distribution of a value in a group of values. The range of values returned by CUME_DIST is between 0 to 1. Tie values always evaluate to the same cumulative distribution value. CUME_DIST calculates the relative position of a precise value in a group of values. For a row r, assuming ascending ordering, the CUME_DIST of r is the quantity of rows with values lower than or equal to the value of r, divided by the quantity of rows being evaluated (the entire query result set or a partition).



VII.EMPRICAL RESULTS

For experiments used PostgreSql of dropping the tuples based on window aggregate function and for analyzing the prediction model rapid miner is used. In this experiment dropping of tuples is performed using four different functions rank(), row_number(), dense_rank() and cume_dsit().After dropping the tuples based on the above discussed window aggregate function . The partition tuples are tested for data quality using three prediction methods Decision Tree, Naïve Bayes and Logistic Regression. The classification models can be evaluated using accuracy, precision and recall of each prediction models. The Naïve bayes algorithm outperforms the remaining ones and the aggregate function best suited of dropping tuples are dense_rank() and cume_dist().

Confusion Matrix

One of the methods to evaluate the performance of a classifier is using confusion matrix the number of correctly classified instances is sum of diagonals in the matrix; all others are incorrectly classified[20]. The following terminology is often used when referring to the counts tabulated in a confusion matrix.

True Positive (TP) [12]**:** The classification model correctly predicting the number of positive examples.

False Negative (FN) [12]**:** corresponds to the number of positive examples wrongly predicted as negative by the classification model.

False Positive (FP) [12]**:** corresponds to the number of negative examples wrongly predicted as positive by the classification model.

True Negative (TN) [12]**:** The classification model correctly predicting the number of nagative examples. The counts are a confusion matrix can also be expressed in terms of percentages. The true positive rate (TPR) or sensitivity is defined as the fraction of positive examples predicted correctly by the model TPR = TP / (TP + FN) Similarly, the true negative rate (TNR) is defined as the fraction of negative examples predicted correctly by the model TNR = TN / (TN + FP) False positive rate (FPR) is defined as the fraction of negative examples predicted as a positive class the model, ie, FPR = FP / (TN + FP) Finally the false negative rate (FNR) is the fraction of positive examples predicted as a negative class. i.e, FNR = FN / (TP + FN)

Recall and Precision:

The recall and Precision values are calculated as follows,

$$\textbf{Precision, p} = \frac{\textbf{TP}}{\textbf{TP + FP}}$$

$$\textbf{Recall, R} = \frac{\textbf{TP}}{\textbf{TP + FP}}$$

TABLE.I Results on Widow aggregation Functions

Window Function	Algorithms								
	Decision Tree			*Logistic regression*			*Naïve Bayes*		
	Acc	*Pre*	*RC*	*Acc*	*Pre*	*RC*	*Acc*	*Pre*	*RC*
Rank wise	50.00	50.00	100	83.67	100	20	100	100	100
Cumulative Disdtance	57.14	0.00	0.0	79.59	0.0	0.0	100	100	100

Window Function	Algorithms								
	Decision Tree			*Logistic regression*			*Naïve Bayes*		
	Acc	*Pre*	*RC*	*Acc*	*Pre*	*RC*	*Acc*	*Pre*	*RC*
Density Rank	57.14	0.00	0.0	79.59	0.0	0.0	100	100	100
Rowwise-Rank	51.02	51.02	100	79.59	0.0	0.0	100	100	100

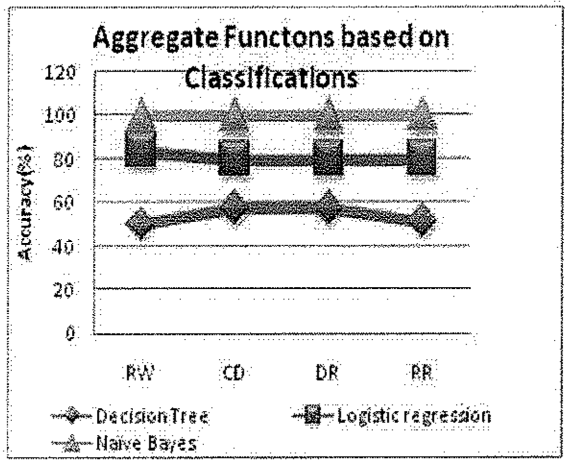


Fig 1: Experimental result based on Accuracy,Precision,Recall

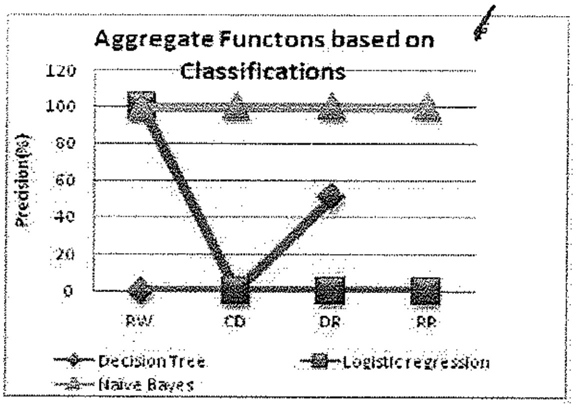


Fig 2: Experimental result based on Precision

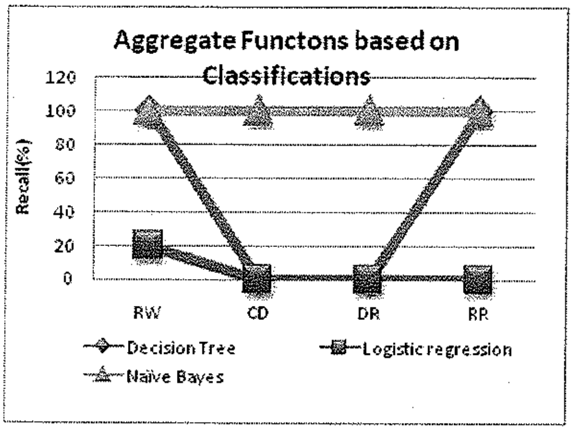


Fig 3: Experimental result based on Precision

From the figures, it is observed that among the four window aggregate functions Rank(),Rowrank(),dense_rank() and cume_dist() performs better in dropping tuples in the datastream to overcome the problem of handling continuous flow of data. The prediction models are applied to find the efficency of each models and the naïve bayes predicted the performance of each function in a significant way.

VIII. Conclusion

In this proposed work the load shedding is based on the concept of dropping tuples during datastream overloading. The tuples are dropped using the different window aggregate function. Then to check the quality of the dataset the prediction models are applied and the result gives the accuracy, precision and recall value of each models. The Naivebayes method outperforms remaning models. The ranking method best performed for tuple dropping is dense_rank() and cumulative distance function.

Acknowledgments.

I Thank the Karpagam University for the motivation and Encouragement for giving me the opportunity to do this research work as successful one.

References

1. Gert Brettlecker, Heiko Schuldt , Peter Fischer , Hans-Jörg

Schek ,: Integration of Reliable Sensor Data Stream Management into Digital Libraries:.

2. Georges HEBRAIL.: Data stream management and mining. In: Mining Massive Data Sets for Security : version 1 - 30 Apr 2010, pages 89-101.
3. Helmy, Y.M. El Zanfaly, D.S. ; Othman, N.A. .:Prioritized query shedding technique for continuous queries over data streams. In: IEEE, Computer Engineering & Systems, 2009. ICCES 2009. International Conference on 14-16 Dec. 2009, Page (s): 418 - 422 .
4. Rusu, F. Dobra, A. .:Sketching Sampled Data Streams. In: Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on March 29 -April 2 2009. Page (s): 381 – 392.
5. Yunyi Zhang , Deyun Zhang ; Chongzheng Huang .: A Novel Adaptive Load Shedding Scheme for Data Stream Processing. In: IEEE, Future Generation Communication and Networking (FGCN 2007) on 6-8 Dec. 2007 Volume: 1 , Page (s): 378 - 384 .
6. Das, A. Gehrke, J. ; Riedewald, M.:Semantic approximation of data stream joins Knowledge and

Data Engineering. In: IEEE Transactions on Jan. 2005 Volume: 17, Issue: 1, Page (s): 44 - 59 .

7. Kuen-Fang Jea, Chao-Wei Li, Chih-Wei Hsu, Ru-Ping Lin, Ssu-Fan Yen.:A load-controllable mining system for frequentpattern discovery in dynamic data streams In: IEEE, Machine Learning and Cybernetics, International Conference on 11-14 July 2010, Vol. IV, Page (s): 2466-2471.
8. Chao-Wei Li, Kuen-Fang Jea, Chih-Wei Hsu, Ru-Ping Lin, Ssu-Fan Yen.:A load shedding scheme for frequent pattern mining in transactional data streams.:IEEE, Fuzzy Systems and Knowledge Discovery, Eighth International Conference on 26-28 July 2011, Vol. II, Page (s): 1294- 1299 .
9. B. Babcock, M. Datar, R. Motwani.:Load shedding for aggregation queries over data streams. In: IEEE, Data Engineering, Proceedings. 20th International Conference on 30 March-2 April 2004, page (s): 350-361.
10. Zhang Longbo,Li Zhanhuai,Wang Zhenyou,Yu Min.:Semantic Load Shedding for Sliding Window Join-Aggregation Queries over Data Streams. In: IEEE, Convergence Information Technology, 2007. International Conference on 21-23 Nov. 2007, Page (s): 2152-2155.
11. Chao-Wei Li, Kuen-Fang Jea, Ru-Ping Lin, Ssu-Fan Yen, Chih-Wei Hsu.: Mining frequent patterns from dynamic data streams with data load Management. In: The Journal of Systems and Software 85 (2012) 1346– 1362.
12. Manganaris S., Christensen M., Zerkle D., Hermiz K.: A data mining analysis of RTID alarms: Computer Networks, 34, 2000, page(s). 571-577.
13. Shengliang Xu ,Magdalena Balazinska.: Sensor Data Stream Exploration for monitoring applications: DMSN'2011.
14. Department of Transporation Coordinated Highways Action Respose Team,Maryland, http://www.chart.state.md.us/default.asp.
15. Carlos H. C. Teixeira, Gustavo H. Orair, Wagner Meira Jr., Srinivasan Parthasarathy.: An Efficient Algorithm for Outlier Detection in High Dimensional Real Databases.
16. Mohamed Medhat Gaber, Shonali Krishnaswamy, Arkady Zaslavsky: Adaptive mining Techniques for Data Stream Using Algorithm Output Granularity.
17. Mohamed Medhat Gaber, Arkady Zaslavsky and Shonali Krishnaswamy .:Mining Data Streams: A Review.In: SIGMOD Record, Vol. 34, No. 2, June 2005,Page(s)18-26.
18. Jia WU, Zhihua CAI.: Attribute Weighting via Differential Evolution Algorithm for Attribute Weighted Naive Bayes (WNB).: Journal of Computational Information Systems 7:5 (2011) page(s) 1672-1679.
19. PostgreSQL Documentation, http:// www.postgresql.org.
20. Knowledge Discovery in Databases- Confusion Matrix, http://www2.cs.uregina.ca.
21. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, .A survey on sensor networks,. IEEE Communications Magazine, vol. 40, no. 8,pp. 102.114, August 2002.
22. T. Arampatzis, J. Lygeros, and S. Manesis, .A survey of applications of wireless sensors and wireless sensor networks,. in Mediterranean Control Conference (Med05), 2005.

23. E. Elnahrawy, .Research directions in sensor data streams: Solutions and challenges,. Rutgers University, Tech. Rep. DCIS-TR- 527, May 2003.

24. E. Elnahrawy and B. Nath. Cleaning and querying noisy sensors. In Submitted for review, 2003.

25. S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, .A taxonomy of wireless micro-sensor network models,. ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, no. 2, pp. 28.36, April 2002.

26. A. Lins, E. F. Nakamura, A. A. Loureiro, and C. J. Coelho Jr.,.Beanwatcher: A tool to generate multimedia monitoring applications for wireless sensor networks,. in Management of Multimedia Networks and Services, ser. Lecture Notes in Computer Science, A. Marshall and N. Agoulmine, Eds., vol. 2839. Belfast, Northern Ireland: Springer-Verlag Heidelberg, September 2003, pp. 128.141.

27. Generating monitoring applications for wireless networks,. In Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2003), Lisbon, Portugal, September 2003.

28. D. J. Abadi, W. Lindner, S. Madden, and J. Schuler, .An integration framework for sensor networks and data stream management systems,.in Proceedings of the Thirtieth International Conference on Very Large Data Bases. VLDB 2004, September 2004, pp. 1361.1364.

29. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, .Models and issues in data stream systems,. in Proceedings of the twenty._rst ACM SIGMOD.SIGACT.SIGART symposium on Principles of database systems, June 2002, pp. 1.16.

30. S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, .Tinydb: An acquisitional query processing system for sensor networks,. ACM Transactions on Database Systems (TODS), vol. 30, no. 1, pp. 122.173, March 2005.

31. Y. Yao and J. Gehrke, .Query processing for sensor networks,. in First Conf. on Innovative Data Systems Research (CIDR), January 2003.

32. T. Arampatzis, J. Lygeros, and S. Manesis, .A survey of applications of wireless sensors and wireless sensor networks,. in Mediterranean Control Conference (Med05), 2005.

33. A. Lins, E. F. Nakamura, A. A. Loureiro, and C. J. Coelho Jr., .Beanwatcher: A tool to generate multimedia monitoring applications for wireless sensor networks,. in Management of Multimedia Networks and Services, ser. Lecture Notes in Computer Science, A. Marshall and N. Agoulmine, Eds., vol. 2839. Belfast, Northern Ireland: Springer-Verlag Heidelberg, September 2003, pp. 128.141.

34. Lior Cohen,Gil Avrahami-Bakish,Mark Last,Abraham Kandel,Oscar Kipersztok,Real-time data mining of nonstationary data streams from sensor networks Volume 9, Issue 3, July 2008, Pages 344–353.

35. D.E. Culler, W. Hong.:Wireless sensor networks: introduction: Communications of the ACM, 47 (6) (2004), pp. 30–33.H. Kargupta, Distributed Data Mining for Sensor Networks, Tutorial, ECML/ PKDD 2004.

36. Varun Chandola, Olufemi A. Omitaomu,Auroop R. Ganguly, Ranga R. Vatsavai,Nitesh V. Chawla, Joao Gama, Mohamed M. Gaber, Knowledge Discovery from Sensor Data (SensorKDD), SIGKDD Explorations Volume 12, Issue 2 P 50-53.

37. Hua-Fu Li , Suh-Yin Lee.: Mining frequent itemsets over data streams using efficient Window sliding techniques: Expert Systems with Applications 36 (2009) page(s).1466- 1477.

Authors Profile

Dr.M.Hemalatha completed MCA, MPhil., PhD in Computer Science and currently working as a Professor and Dept. of Computer Science in Karpagam University. Twelve years of Experience in teaching and published more than hundred papers in International Journals and also presented more than eighty one paper in various national conferences and international conference. Area of research is Data Mining, Software Engineering, Bioinformatics, and Neural Network. She is a Reviewer in several National and International Journals.

S.Senthamilarasu, completed MCA, Pursuing Ph.D Research in computer Science, under the guidance of Dr.M.Hemalatha, Professor and Head, Dept. Software System in Karpagam University, Coimbatore,Tamilnadu. Presented two papers in national conferences. Area of my Research is Data Stream in Data mining.