A SURVEY OF STREAM DATA MINING

Elena Ikonomovska, Suzana Loskovska, Dejan Gjorgjevik

University Ss. Cyril & Methodius, Faculty of Electrical Engineering and Information Technologies, Skopje, Republic of Macedonia, elenai@feit.ukim.edu.mk, suze@feit.ukim.edu.mk, dejan@feit.ukim.edu.mk

***Abstract* – At present a growing number of applications that generate massive streams of data need intelligent data processing and online analysis. Real-time surveillance systems, telecommunication systems, sensor networks and other dynamic environments are such examples. The imminent need for turning such data into useful information and knowledge augments the development of systems, algorithms and frameworks that address streaming challenges. The storage, querying and mining of such data sets are highly computationally challenging tasks. Mining data streams is concerned with extracting knowledge structures represented in models and patterns in non stopping streams of information. In this paper, we present the theoretical foundations of data stream analysis and identify potential directions of future research. Mining data stream techniques are being critically reviewed.**

***Index terms*—data streams, data mining, review**

1. INTRODUCTION

Recently a new class of emerging applications has become widely recognized: applications in which data is generated at very high rates in the form of transient *data streams*. Examples of such applications include financial applications, network monitoring, security, telecommunication data management, web applications, manufacturing, sensor networks, and others. In the data stream model, individual data items may be relational tuples, e.g., network measurements, call records, web page visits, sensor readings, and so on. However, their continuous arrival in multiple, rapid, time-varying, possibly unpredictable and unbound streams open new fundamental research problems. This rapid generation of continuous streams of information has challenged our storage, computation and communication capabilities in computing systems. The vast amounts of data arriving in high speeds need employment of semi-automated interactive techniques, to perform real-time extraction of hidden knowledge and information.

From the last decade, data mining, meaning *extracting useful information or knowledge from large amounts of data*, has become the key technique to analyze and understand data. Typical data mining tasks include association mining, classification, and clustering. These techniques help find interesting patterns, regularities, and anomalies in the data. However, traditional data mining techniques cannot directly apply to data streams. This is because most of them require multiple scans of data to extract the information, which is unrealistic for stream data. The amount of previously happened events is usually overwhelming, so they can be either dropped after processing or archived separately in secondary storage. More importantly, the characteristics of the data stream can change over time and the evolving pattern needs to be captured. Furthermore, we also need to consider the problem of resource allocation in mining data streams. Due to the large volume and the high speed of streaming data, mining algorithms must cope with the effects of system overload. Thus, how to achieve optimum results under various resource constraints becomes a challenging task.

In this paper, we review the theoretical foundations of data stream analysis. Some general issues in stream data mining are discussed. Mining data stream techniques are critically reviewed.

The paper is organized as follows. Section 2 presents the theoretical background of data stream analysis. Mining data stream techniques and systems are reviewed in section 3. Some open research issues are discussed in section 4. Section 5 summarizes this review paper.

2. FOUNDATIONS

The foundations on which stream data mining solutions rely, come from the field of statistics, complexity and computational theory [44]. The online nature of data streams and their potentially

high arrival rates impose high resource requirements on data stream processing systems. In order to deal with resource constraints in a graceful manner, many data summarization techniques have been adopted from the field of statistics. They provide means to examine only a subset of the whole dataset or to transform the data vertically or horizontally to an approximate smaller size data representation so that known data mining techniques can be used. Also, techniques from computational theory have been implemented to achieve time and space efficient solutions. In this section we review these theoretical foundations.

Summarization techniques are often used for producing approximate answers from large databases. They synthesize techniques for data reduction and synopsis construction. Summarization techniques refer to the process of transforming data to a suitable form for stream data analysis. This can be done by summarizing the whole dataset or choosing a subset of the incoming stream to be analyzed. When summarizing the whole dataset techniques such as sampling, sketching and load shedding are used. For choosing a subset of the incoming stream synopsis data structures and aggregation techniques are used. An excellent review of data reduction techniques is presented in [12]. We present the basics of these techniques with examples of their applications in the context of data stream analysis.

Sampling – The idea of representing a large dataset by a small random sample of the data elements goes back to the end of the nineteenth century and has led to the development of a large body of survey-sampling techniques. Sampling is the process of statistically selecting the elements of the incoming stream that would be analyzed [21]. Some previous work on the field of employing sampling techniques for computing approximate frequency counts over data streams is presented in [42]. Domingo *et al.* [19][21] studied sampling approach to tackle decision tree classification and k-means clustering. Sampling plays an important role in developing techniques for clustering data streams [35]. Babcock, Datar, and Motwani [8] studied sampling in the sliding window model. Also, sampling has been often used as a data reduction technique for producing approximate answers for queries over data streams in [1][2].The problem with using sampling in the context of data stream analysis is the unknown dataset size. Therefore the treatment of data stream should follow a special analysis to find the error bounds. Sampling also does not address the problem of fluctuating data rates. When using sampling, it would be worth investigating the relationship among the three parameters: data rate, sampling rate and error bounds. Designing sampling-based algorithms that can produce approximate answers that are provably close to the exact answer is an important and active area of research.

Sketching – Sketching [7][42] involves building a summary of a data stream using a small amount of memory. It is the process of vertically sampling the incoming stream. Sketching has been applied in comparing different data streams and in aggregate queries. Alon et al. in [6] introduced the notion of *randomized sketching* which has been widely used ever since. Dobra et al. in [18] demonstrate that sketching can be generalized to provide approximate answers to complex, multi-join, aggregate SQL queries over streams with explicit and tunable guarantees on the approximation error. Techniques based on sketching are very convenient to distributed computation over multiple streams. The major drawback of sketching is that of accuracy. Principal Component Analysis (PCA) would be a better solution if being applied in streaming applications [39].

Load Shedding – Load shedding refers to the process of eliminating a batch of subsequent elements (randomly or semantically) from being analyzed [15]. It has the same problems of sampling. Load shedding is not a preferred approach with mining algorithms, especially in time series analysis because it drops chunks of data streams that might represent a pattern of interest. Still, it has been successfully used in sliding window aggregate queries [9].

Synopsis Data Structures – Synopsis data structures embody the idea of small space, approximate solution to massive data set problems. Creating synopsis of data refers to the process of applying summarization techniques that are capable of summarizing the incoming stream for further analysis. Wavelet analysis [31], histograms, and frequency moments [7] have been proposed as synopsis data structures.

Wavelets are one of the often-used techniques for providing a summary representation of the data. Wavelets coefficients are projections of the given signal (set of data values) onto an orthogonal set of basis vector. They have the desirable property that the signal reconstructed from the top few wavelet coefficients best approximates the original signal in terms of the L_2 norm. There has been some research done in computing the top wavelet coefficients in the data stream model. The technique of Gilbert et al. [32], gives rise to an easy greedy algorithm to find the best B-term Haar wavelet representation.

Histograms approximate the data in one or more attributes of a relation by grouping attribute values into "buckets" (subsets) and approximating true attribute values and their frequencies in the data based on a summary statistics maintained in each bucket [7]. For most real-world databases, there exist histograms that produce low-error estimates while occupying reasonably small space. Hence, they are the most commonly used form of statistics in practice. There has been some work on using them for approximate query answering [1][2].

Aggregation – Aggregation is the representation of number of elements in one aggregated element using some statistical measure such as means, variance or the average. It is often considered as a data rate adaptation technique in a resource-aware mining [28]. The problem with aggregation is that it does not perform well with highly fluctuating data distributions. Algorithms over data streams that utilize aggregation include approximate quantiles [41][33], V-optimal histograms [36], wavelet based aggregate queries [30][43], and correlated aggregate queries [29]. Merging online aggregation with offline mining has been studied in [1][4][5]. Aggregation has been successfully used in distributed stream data environments and with continuous queries over data streams [11].

Sliding Window – is considered as an advanced technique for producing approximate answers to a data stream query. The idea behind sliding window is to perform detailed analysis over the most recent data items and over summarized versions of the old ones. This idea has been adopted in many techniques in the undergoing comprehensive data stream mining system *MAIDS* [22]. Imposing sliding windows on data streams is a natural method for approximation that has several attractive properties. It is well-defined and easily understood. It is deterministic, so there is no danger that unfortunate random choices will produce a bad approximation. Most importantly, it emphasizes recent data, which in the majority of real-world applications is more important and relevant than old data.

Having discussed the different theoretical approaches to data stream analysis problems, the following section is devoted to stream mining techniques that use the above theoretical approaches in different ways.

3. MINING TECHNIQUES

Mining data streams has been a very attractive field of research for the data mining community for the last few years. The algorithmic ideas above presented have proved powerful for solving a variety of problems in data streams. A number of algorithms have been proposed to deal with the high speed feature in mining data streams using different techniques. In this section, we present the related work in mining data streams, concerning clustering, classification and frequency counting techniques.

3.1 Clustering

Clustering of stream data has been one of the most studied data mining tasks in this growing field. The centre of the attention for many researchers has been the k-median problem, initially posed by Weber [51]. The objective is to minimize the average distance from data points to their closest cluster centers.

A large body of algorithms has been proposed that deal with this problem. Guha et al. [35][37] proposed an algorithm that makes a single pass over the data stream and uses small space. Babcock et al. [10] have used exponential histogram (EH) data structure to improve Guha et al. algorithm [35]. Charikar et al [14] have proposed another k-median algorithm that overcomes the problem of increasing approximation factors in the Guha et al [35] algorithm.

Another algorithm that captured the attention of many scientists is the k-means clustering algorithm. This algorithm has also been studied analytically by Domingos et al. [19][21]. They have proposed a general method for scaling up machine learning algorithms named Very Fast Machine Learning *VFML*. They have applied this method to K-means clustering *VFKM* and decision tree classification *VFDT* techniques. Ordonez [47] has proposed an improved incremental k-means algorithm for clustering binary data streams. O'Challaghan et al. [45] proposed STREAM and LOCALSEARCH algorithms for high quality data stream clustering. Aggarwal et al. [1][5] have proposed a framework for clustering evolving data steams called CluStream algorithm. In [4] they have proposed the HPStream; a projected clustering for high dimensional data streams, which has outperformed CluStream. Stanford's STREAM project has studied the approximate k-median clustering with guaranteed probabilistic bound [14][35][37][46].

3.2 Classification

Several authors have studied the idea of implementing a decision tree technique for classification of stream data. Ding et al. [17] have developed a decision tree based on Peano count tree data structure. Domingos et al. [19][20] have studied the problem of maintaining decision trees over data streams. In [19] they have developed the VFDT system. It is a decision tree learning system based on Hoeffding trees. Ganti et al. [23] have developed analytically two algorithms GEMM and FOCUS for model maintenance and change detection between two data sets in terms of the data mining results they induce. The algorithms have been applied to decision tree models and the frequent itemset model. Techniques such as decision trees are useful for one-pass mining of data streams but these cannot be easily used in the context of an on-demand classifier in an evolving environment.

The concept drifting problem in stream data classification has been addressed by several authors. Wang et al. [50] have proposed a general framework for mining concept drifting data streams. The proposed technique uses weighted classifier ensembles to mine data streams. Last [40] has proposed an online classification system which dynamically adjusts the size of the training window and the number of new examples between model re-constructions to the current rate of concept drift.

Aggarwal et al. in [5] have presented a different view on the data stream classification problem from the perspective of a dynamic approach, in which simultaneous training and testing streams are used for dynamic classification of data sets .

3.3 Frequency Counting

Frequency counting has not attracted much attention among the researchers in this filed, as did clustering and classification. Counting frequent items or itemsets is one of the issues considered in frequency counting. Cormode and Muthukrishnan [16] have developed an algorithm for counting frequent items. The algorithm maintains a small space data structure that monitors the transactions on the relation, and when required, quickly outputs all hot items, without rescanning the relation in the database. Giannella et al. [24] have developed a frequent itemset mining algorithm over data stream. They have proposed the use of tilted windows to calculate the frequent patterns for the most recent transactions. Manku and Motwani [42] have proposed and implemented an incremental algorithm for approximate frequency counting in data streams that uses all the previous historical data to calculate the frequent patterns.

4. SOME RESEARCH ISSUES

Data stream mining is a stimulating field of study that has raised many challenges and research issues that need to be addressed by the machine learning and data mining communities. The characteristics of data streams as pointed out in Section 1 indicate that when developing mining techniques of this kind, there are more issues that need to be considered than in traditional databases. The following is a brief discussion of some crucial open research issues:

- **Memory management**: The first fundamental issue we need to consider is how to optimize the memory space consumed by the mining algorithm. Memory management is a particular challenge when processing streams because many real data streams are irregular in their rate of arrival, exhibiting burstiness and variation of data arrival rate over time. A stream mining algorithm with high memory cost will have difficulty being applied in many situations, such as sensor networks. More research needs to be done in developing new summarization techniques for collecting valuable information from data streams. Fully addressing this issue in the mining algorithm can greatly improve its performance [34].
- **Data pre-processing:** data pre-processing is an important and time consuming phase in the knowledge discovery process and must be taken into consideration when mining data streams. Designing a light-weight preprocessing techniques that can guarantee quality of the mining results is crucial. The challenge here is to automate such a process and integrate it with the mining techniques.
- **Compact data structure:** Due to bounded memory size and the huge amount of data streams coming continuously, efficient and compact data structure is needed to store, update and retrieve the collected information. Failure in developing such a data structure will largely decrease the efficiency of the mining algorithm. Even if we store the information in disks, the additional I/O operations will increase the processing time. Incremental maintaining of the data structure is a necessity since it is not possible to rescan the entire input. Also, novel indexing, storage and querying techniques are required to handle the continual fluctuated flow of information streams.
- **Resource aware:** This is a fundamental issue that considers the problem of how the limited resources, e.g., memory space and computation power, can be well utilized to produce accurate estimates. Stream data mining algorithms must not ignore the problem of nearly consuming the available resources while they are still running. Data will be lost when the memory is used up and this would lead to inaccuracy of the mining results, thus degrade the performance of the mining algorithm. Several authors in [25][26][48] discussed this issue and proposed their solutions for resource-aware mining. Gaber et al. proposed an approach, called AOG, which uses a control parameter to control its output rate according to memory, time constrains and data stream rate [25][26]. Teng et al. proposed an algorithm, that not only reduces the memory required for data storage but also retains good approximation given limited resources like memory space and computation power [49].
- **Visualization of results**: Visualization is a powerful way to facilitate data analysis, but it is crucial that visualization systems explicitly express the presence, nature, and degree of uncertainty to users. Visualization of traditional data mining results on a desktop is still a research issue, while visualization on small screens of a PDA, for example is a real challenge. Streamed data analyzed on a PDA should be efficiently visualized in a way that will enable taking a quick decision.

5. CONCLUSION

The spreading of data stream phenomenon in real life applications has influenced in great manner the development of stream mining algorithms. Mining data streams has raised a number of research challenges for the data mining community. Due to the resource and time constraints many summarization and approximation techniques have been adopted from the fields of statistics and computational theory. Based on these foundations, a number of clustering,

classification and frequency counting techniques have been developed.

Mining data streams is an immature, growing field of study. There are many open issues that need to be addressed. The development of systems that will fully address these issues is crucial for accelerating the science discovery in the fields of physics and astronomy [27], as well as in business and financial applications [39]. This would improve the real-time decision making process in almost every area of our life.

6. REFERENCES

[1] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. Join synopses for approximate query answering. In *Proc. of the 1999 ACM SIGMOD Intl. Conf. on Management of Data*, pages 275–286, June 1999.

[2] S. Acharya, P. B. Gibbons, and V. Poosala. Congressional samples for approximate answering of group-by queries. In *Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data*, pages 487–498, May 2000.

[3] C. Aggarwal, J. Han, J. Wang, P. S. Yu, A Framework for Clustering Evolving Data Streams, Proc. 2003 Int. Conf. on Very Large Data Bases, Berlin, Germany, Sept. 2003.

[4] C. Aggarwal, J. Han, J. Wang, and P. S. Yu, A Framework for Projected Clustering of High Dimensional Data Streams, Proc. 2004 Int. Conf. on Very Large Data Bases, Toronto, Canada, 2004.

[5] C. Aggarwal, J. Han, J. Wang, and P. S. Yu, On Demand Classification of Data Streams, Proc. 2004 Int. Conf. on Knowledge Discovery and Data Mining, Seattle, WA, Aug. 2004.

[6] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proc. of the 1996 Annual ACM Symp. on Theory of Computing*, pages 20–29, 1996.

[7] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In Proceedings of PODS, 2002.

[8] B. Babcock, M. Datar, and R. Motwani. “Sampling from a Moving Window over Streaming Data.” In Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2002, pages 633–634.

[9] B. Babcock, M. Datar, and R. Motwani. Load Shedding Techniques for Data Stream Systems (short paper) In Proc. of the 2003 Workshop on Management and Processing of Data Streams, June 2003.

[10] B. Babcock, M. Datar, R. Motwani, L. O'Callaghan: Maintaining Variance and k-Medians over Data Stream Windows, Proceedings of the 22nd Symposium on Principles of Database Systems, 2003.

[11] S. Babu, and J. Widom Continuous queries over data streams. SIGMOD Record, 30:109-120, 2001.

[12] D. Barbara et al. The New Jersey data reduction report. Bull. Technical Committee on Data Engineering, 20:3-45, Dec. 1997.

[13] Y. D. Cai, D. Clutter, G. Pape, J. Han, M. Welge, L. Auvil. MAIDS: Mining Alarming Incidents from Data Streams. Proceedings of the 23rd ACM SIGMOD International Conference on Management of Data, June 13-18, 2004, Paris, France.

[14] M. Charikar, L. O'Callaghan, and R. Panigrahy. Better streaming algorithms for clustering problems In Proc. of 35th ACM Symposium on Theory of Computing, 2003.

[15] Y. Chi, H. Wang and P.S. Yu. Loadstar : Load Shedding in Data Stream Mining. In Proc. The 31st VLDB Conf., Trondheim, Norway, 2005, pp. 1302—1305.

[16] G. Cormode, S. Muthukrishnan What's hot and what's not: tracking most frequent items dynamically. PODS 2003: 296-306

[17] Q. Ding, Q. Ding, and W. Perrizo, Decision Tree Classification of Spatial Data Streams Using Peano Count Trees, Proceedings of the ACM Symposium on Applied Computing, Madrid, Spain, March 2002.

[18] A. Dobra, M. Garofalakis, J. Gehrke, R. Rastogi. Processing Complex Aggregate Queries Over Data Streams. In Proceedings of SIGMOD, 2002.

[19] P. Domingos and G. Hulten. Mining High-Speed Data Streams. In Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining, 2000.

[20] P. Domingos, G. Hulten, and L. Spencer. Mining time-changing data streams. In *Proc. of the 2001ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 97–106, 2001.

[21] P. Domingos and G. Hulten, A General Method for Scaling Up Machine Learning Algorithms and its Application to Clustering, Proceedings of the Eighteenth International Conference on Machine Learning, 2001, Williamstown, MA, Morgan Kaufmann.

[22] G. Dong, J. Han, L.V.S. Lakshmanan, J. Pei, H. Wang and P.S. Yu. Online mining of changes from data streams: Research problems and preliminary results, In Proceedings of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams. In cooperation with the 2003 ACM-SIGMOD International Conference on Management of Data, San Diego, CA, June 8, 2003.

[23] V. Ganti, J. Gehrke, and R. Ramakrishnan: Mining Data Streams under Block Evolution. SIGKDD Explorations 3(2), 2002.

[24] C. Giannella, J. Han, J. Pei, X. Yan, and P.S. Yu, Mining Frequent Patterns in Data Streams at Multiple Time Granularities, in H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), Next Generation Data Mining, AAAI/MIT, 2003.

[25] M. M. Gaber, S. Krishnaswamy, A. Zaslavsky; Adaptive Mining Techniques for Data Streams Using Algorithm Output Granularity; The Australasian Data Mining Workshop; December 2003.

[26] M. M. Gaber, A. Zaslavsky S. Krishnaswamy; Resource- Aware Knowledge Discovery in Data Streams; Int'l Workshop on Knowledge Discovery in Data Streams; September 2004.

[27] M. M. Gaber, A. Zaslavsky, S. Krishnaswamy; Towards an Adaptive Approach for Mining Data Streams in Resource Constrained Environments, the Proceedings of Sixth International Conference on Data Warehousing and Knowledge Discovery – Industry Track (DaWak 2004), Zaragoza, Spain, 30 August – 3 September, Lecture Notes in Computer Science (LNCS), Springer Verlag.

[28] M. M. Gaber, S. Krishnaswamy and A. Zaslavsky. Resource-aware mining of data streams. Journal of Universal Computer Science, vol. 11, no. 8 (2005), 1440-1453 submitted: 10/3/05, accepted: 5/5/05, appeared: 28/8/05 © J.UCS.

[29] J. Gehrke, F. Korn, and D. Srivastava. On computing correlated aggregates over continual data streams. In *Proc. of 2001 ACM SIGMOD*, pages 13–24, 2001.

[30] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *Proc. of 27th Intl. Conf. on Very Large Data Bases*, 2001.

[31] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, M. Strauss: One-Pass Wavelet Decompositions of Data Streams. TKDE 15(3), 2003.

[32] A. Gilbert et al. Fast, small-space algorithms for approximate histogram maintenance. In Proc. of the 2002 Annual ACM Symp. on Theory of Computing, 2002.

[33] M. Greenwald, S. Khanna. Space-efficient online computation of quantile summaries. In *Proc. of 2001 ACM SIGMOD*, pages 58–66, 2001.

[34] L. Golab and M. T. Ozsu. Issues in Data Stream Management. In SIGMOD Record, Volume 32, Number 2, June 2003.

[35] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In Proceedings of the Annual Symposium on Foundations of Computer Science. IEEE, November 2000.

[36] S. Guha, N. Koudas, and K. Shim. Data-streams and histograms. In *Proc. of 33rd Annual ACM Symp. on Theory of Computing*, pages 471–475, July 2001.

[37] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, Clustering Data Streams: Theory and Practice TKDE special issue on clustering, vol. 15, 2003.

[38] J. Hellerstein, P. Haas, and H. Wang. Online aggregation. In Proc. of the 1997 ACM SIGMOD Intl. Conf. on Management of Data, pages 171–182, May 1997.

[39] H. Kargupta et al. VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring, Proceedings of SIAM International Conference on Data Mining, 2004.

[40] M. Last, Online Classification of Nonstationary Data Streams, Intelligent Data Analysis, Vol. 6, No. 2, pp. 129-147, 2002.

[41] G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proc. of 1999 ACM SIGMOD*, pages 251–262, 1999.

[42] G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, August 2002.

[43] Y. Matias, J. S. Vitter, and M. Wang. Dynamic maintenance of wavelet-based histograms. In *Proc. Of 26th Intl. Conf. on Very Large Data Bases*, pages 101–110, 2000.

[44] S. Muthukrishnan, Data streams: algorithms and applications. Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms 2003.

[45] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. Streaming-data algorithms for high-quality clustering. Proceedings of IEEE International Conference on Data Engineering, March 2002.

[46] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. High-performance clustering of streams and large data sets. In *Proc. of the 2002 Intl. Conference on Data Engineering (ICDE 2002)*, Feb 2002.

[47] C. Ordonez. Clustering Binary Data Streams with K-means ACM DMKD 2003.

[48] S. Papadimitriou, C. Faloutsos, and A. Brockwell, Adaptive, Hands-Off Stream Mining, 29th International Conference on Very Large Data Bases VLDB, 2003.

[49] W. Teng, M. Chen, and P. S. Yu; Resource-Aware Mining with Variable Granularities in Data Streams; SIAM Int'l Conf. on Data Mining; 2004.

[50] H. Wang, W. Fan, P. Yu and J. Han; Mining Concept-Drifting Data Streams using Ensemble Classifiers, in the 9th ACM International Conference on Knowledge Discovery and Data

Mining (SIGKDD), Aug. 2003, Washington DC,USA.

[51] G. Wesolowsky. The Weber problem: History and perspective. *Location Science,* 1:5-23, 1993.