Mining Data Streams: A Review

Mohamed Medhat Gaber, Arkady Zaslavsky and Shonali Krishnaswamy
Centre for Distributed Systems and Software Engineering, Monash University
900 Dandenong Rd, Caulfield East, VIC3145, Australia
{Mohamed.Medhat.Gaber, Arkady.Zaslavsky, Shonali.Krishnaswamy} @infotech.monash.edu.au

Abstract

The recent advances in hardware and software have enabled the capture of different measurements of data in a wide range of fields. These measurements are generated continuously and in a very high fluctuating data rates. Examples include sensor networks, web logs, and computer network traffic. 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. The research in data stream mining has gained a high attraction due to the importance of its applications and the increasing generation of streaming information. Applications of data stream analysis can vary from critical scientific and astronomical applications to important business and financial ones. Algorithms, systems and frameworks that address streaming challenges have been developed over the past three years. In this review paper, we present the state-of-the-art in this growing vital field.

1- Introduction

The intelligent data analysis has passed through a number of stages. Each stage addresses novel research issues that have arisen. Statistical exploratory data analysis represents the first stage. The goal was to explore the available data in order to test a specific hypothesis. With the advances in computing power, machine learning field has arisen. The objective was to find computationally efficient solutions to data analysis problems. Along with the progress in machine learning research, new data analysis problems have been addressed. Due to the increase in database sizes, new algorithms have been proposed to deal with the scalability issue. Moreover machine learning and statistical analysis techniques have been adopted and modified in order to address the problem of very large databases. Data mining is that interdisciplinary field of study that can extract models and patterns from large amounts of information stored in data repositories [30, 31, 34].

Advances in networking and parallel computation have lead to the introduction of distributed and parallel data mining. The goal was how to extract knowledge from different subsets of a dataset and integrate these generated knowledge structures in order to gain a global model of the whole dataset. Client/server, mobile agent based and hybrid models have been proposed to address the communication overhead issue. Different variations of algorithms have been developed in order to increase the accuracy of the generated global model. More details about distributed data mining could be found in [47].

Recently, the data generation rates in some data sources become faster than ever before. This rapid generation of continuous streams of information has challenged our storage, computation and communication capabilities in computing systems. Systems, models and techniques have been proposed and developed over the past few years to address these challenges [5, 44].

In this paper, we review the theoretical foundations of data stream analysis. Mining data stream systems, techniques are critically reviewed. Finally, we outline and discuss research problems in streaming mining field of study. These research issues should be addressed in order to realize robust systems that are capable of fulfilling the needs of data stream mining applications.

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 sections 3 and 4 respectively. Open and addressed research issues in this growing field are discussed in section 5. Finally section 6 summarizes this review paper.

2- Theoretical Foundations

Research problems and challenges that have been arisen in mining data streams have its solutions using well-established statistical and computational approaches. We can categorize these solutions to data-based and task-based ones. In data-based solutions, the idea is to examine only a subset of the whole dataset or to transform the data vertically or horizontally to an approximate smaller size data representation. At the other hand, in task-based solutions, techniques from computational theory have been adopted to achieve time

and space efficient solutions. In this section we review these theoretical foundations.

2.1 Data-based Techniques

Data-based techniques refer to summarizing the whole dataset or choosing a subset of the incoming stream to be analyzed. Sampling, load shedding and sketching techniques represent the former one. Synopsis data structures and aggregation represent the later one. Here is an outline of the basics of these techniques with pointers to its applications in the context of data stream analysis.

2.1.1 Sampling

Sampling refers to the process of probabilistic choice of a data item to be processed or not. Sampling is an old statistical technique that has been used for a long time. Boundaries of the error rate of the computation are given as a function of the sampling rate. Very Fast Machine Learning techniques [16] have used Hoeffding bound to measure the sample size according to some derived loss functions.

The problem with using sampling in the context of data stream analysis is the unknown dataset size. Thus the treatment of data stream should follow a special analysis to find the error bounds. Another problem with sampling is that it would be important to check for anomalies for surveillance analysis as an application in mining data streams. Sampling may not be the right choice for such an application. Sampling also does not address the problem of fluctuating data rates. It would be worth investigating the relationship among the three parameters: data rate, sampling rate and error bounds.

2.1.2 Load Shedding

Load shedding refers [6, 52] to the process of dropping a sequence of data streams. Load shedding has been used successfully in querying data streams. It has the same problems of sampling. Load shedding is difficult to be used with mining algorithms because it drops chunks of data streams that could be used in the structuring of the generated models or it might represent a pattern of interest in time series analysis.

2.1.3 Sketching

Sketching [5, 44] is the process of randomly project a subset of the features. It is the process of vertically sample the incoming stream. Sketching has been applied in comparing different data streams and in aggregate queries. The major drawback of sketching is that of accuracy. It is hard to use it in the context of data stream mining. Principal Component Analysis (PCA) would be a better solution that has been applied in streaming applications [38].

2.1.4 Synopsis Data Structures

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 [25], histograms, quantiles and frequency moments [5] have been proposed as synopsis data structures. Since synopsis of data does not represent all the characteristics of the dataset, approximate answers are produced when using such data structures.

2.1.5 Aggregation

Aggregation is the process of computing statistical measures such as means and variance that summarize the incoming stream. Using this aggregated data could be used by the mining algorithm. The problem with aggregation is that it does not perform well with highly fluctuating data distributions. Merging online aggregation with offline mining has been studies in [1, 2, 3].

2.2 Task-based Techniques

Task-based techniques are those methods that modify existing techniques or invent new ones in order to address the computational challenges of data stream processing. Approximation algorithms, sliding window and algorithm output granularity represent this category. In the following subsections, we examine each of these techniques and its application in the context of data stream analysis.

2.2.1 Approximation algorithms

Approximation algorithms [44] have their roots in algorithm design. It is concerned with design algorithms for computationally hard problems. These algorithms can result in an approximate solution with error bounds. The idea is that mining algorithms are considered hard computational problems given its features of continuality and speed and the generating environment that is featured by being resource constrained. Approximation algorithms have attracted researchers as a direct solution to data stream mining problems. However, the problem of data rates with regard with the available resources could not be solved using approximation algorithms. Other tools should be used along with these algorithms in order to adapt to the

available resources. Approximation algorithms have been used in [13]

2.2.2 Sliding Window

The inspiration behind sliding window is that the user is more concerned with the analysis of most recent data streams. Thus the detailed analysis is done over the most recent data items and summarized versions of the old ones. This idea has been adopted in many techniques in the undergoing comprehensive data stream mining system *MAIDS* [17].

2.2.3 Algorithm Output Granularity

The algorithm output granularity (AOG) [21, 22, 23] introduces the first resource-aware data analysis approach that can cope with fluctuating very high data rates according to the available memory and the processing speed represented in time constraints. The AOG performs the local data analysis on a resource constrained device that generates or receive streams of information. AOG has three main stages. Mining followed by adaptation to resources and data stream rates represent the first two stages. Merging the generated knowledge structures when running out of memory represents the last stage. AOG has been used in clustering, classification and frequency counting [21].

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 attracted the attention of data mining community for the last three years. A number of algorithms have been proposed for extracting knowledge from streaming information. In this section, we review clustering, classification, frequency counting and time series analysis techniques.

3.1 Clustering

Guha et al. [27, 28] have studied analytically clustering data streams using K-median technique. The proposed algorithm makes a single pass over the data stream and uses small space. It requires O(nk) time and O(nε) space where "k" is the number of centers, "n" is the number of points and $\varepsilon<1$. They have proved that any k-median algorithm that achieves a constant factor approximation can not achieve a better run time than O(nk). The algorithm starts by clustering a calculated size sample according to the available memory into 2k, and then at a second level, the algorithm clusters the above points for a number of samples into 2k and this process is repeated to a number of levels, and finally it clusters the 2k clusters into k clusters.

Babcock et al. [7] have used exponential histogram (EH) data structure to improve Guha et al. algorithm [27]. They use the same method described above, however they address the problem of merging clusters when the two sets of cluster centers to be merged are far apart by maintaining the EH data structure. They have studied their proposed algorithm analytically.

Charikar et al [11] have proposed another k-median algorithm that overcomes the problem of increasing approximation factors in the Guha et al [27] algorithm with the increase in the number of levels used to result in the final solution of the divide and conquer algorithm. The algorithm has also been studied analytically

Domingos et al. [15, 16, 35] have proposed a general method for scaling up machine learning algorithms. They have termed this approach Very Fast Machine Learning *VFML*. This method depends on determining an upper bound for the learner's loss as a function in number of data items to be examined in each step of the algorithm. They have applied this method to K-means clustering *VFKM* and decision tree classification *VFDT* techniques. These algorithms have been implemented and evaluated using synthetic data sets as well as real web data streams. VFKM uses the Hoeffding bound to determine the number of examples needed in each step of K-means algorithm. The VFKM runs as a sequence of K-means executions with each run uses more examples than the previous one until a calculated statistical bound (Hoeffding bound) is satisfied.

Ordonez [46] has proposed several improvements to k-means algorithm to cluster binary data streams. He has developed an incremental k-means algorithm. The experiments were conducted on real data sets as well as synthetic ones. He has demonstrated experimentally that the proposed algorithm outperforms the scalable k-means in the majority of cases. The proposed algorithm is a one pass algorithm in O(Tkn) complexity, where T is the average transaction size, n is number of transactions and k is number of centers. The use of binary data simplifies the manipulation of categorical data and eliminates the need for data normalization. The main idea behind the proposed algorithm is that it updates the cluster centers and weights after examining a batch of transactions which equalizes square root of the number of transactions rather than updating them one by one.

O'Challaghan et al. [45] have proposed STREAM and LOCALSEARCH algorithms for high quality data stream clustering. The STREAM algorithm

starts by determining the size of the sample and then applies the LOCALSEARCH algorithm if the sample size is larger than a pre-specified equation result. This process is repeated for each data chunk. Finally, the LOCALSEARCH algorithm is applied to the cluster centers generated in the previous iterations.

Aggarwal et al. [1] have proposed a framework for clustering data steams called CluStream algorithm. The proposed technique divides the clustering process into two components. The online component stores summarized statistics about the data streams and the offline one performs clustering on the summarized data according to a number of user preferences such as the time frame and the number of clusters. A number of experiments on real datasets have been conducted to prove the accuracy and efficiency of the proposed algorithm. They [2] have recently proposed HPStream; a projected clustering for high dimensional data streams. HPStream has outperformed CluStream in recent results.

Keogh et al [39] have proved empirically that most highly cited clustering of time series data streams algorithms proposed so far in the literature come out with meaningless results in subsequence clustering. They have proposed a solution approach using k-motif to choose the subsequences that the algorithm can work on to produce meaningful results.

Gaber et al. [21] have developed Lightweight Clustering *LWC*. It is an AOG-based algorithm. AOG has been discussed in section 2. The algorithm adjusts a threshold that represents the minimum distance measure between data items in different clusters. This adjustment is done regularly according to a pre-specified time frame. It is done according to the available resources by monitoring the input-output rate. This process is followed by merging clusters when the memory is full.

3.2 Classification

Wang et al. [53] have proposed a general framework for mining concept drifting data streams. They have observed that data stream mining algorithms proposed so far have not addressed the concept of drifting in the evolving data. The proposed technique uses weighted classifier ensembles to mine data streams. The expiration of old data in their model depends on the data distribution. They use synthetic and real life data streams to test their algorithm and compare between the single classifier and classifier ensembles. The proposed algorithm combines multiple classifiers weighted by their expected prediction accuracy. Also the selection of number of classifiers instead of using all is an option in the proposed framework without loosing accuracy in the classification process.

Ganti et al. [18] have developed analytically an algorithm for model maintenance under insertion and deletion of blocks of data records. This algorithm can be applied to any incremental data mining model. They have also described a generic framework for change detection between two data sets in terms of the data mining results they induce. They formalize the above two techniques into two general algorithms: GEMM and FOCUS. The algorithms have been applied to decision tree models and the frequent itemset model. GEMM algorithm accepts a class of models and an incremental model maintenance algorithm for the unrestricted window option, and outputs a model maintenance algorithm for both window-independent and window-dependent block selection sequence. FOCUS framework uses the difference between data mining models as the deviation in data sets.

Domingos et al. [15] have developed VFDT. It is a decision tree learning systems based on Hoeffding trees. It splits the tree using the current best attribute taking into consideration that the number of examined data items used satisfies a statistical measure which is Hoeffding bound. The algorithm also deactivates the least promising leaves and drops the non-potential attributes.

Papadimitriou et al. [48] have proposed AWSOM (Arbitrary Window Stream mOdeling Method) for interesting pattern discovery from sensors. They developed a one-pass algorithm to incrementally update the patterns. Their method requires only O(log N) memory where N is the length of the sequence. They conducted experiments with real and synthetic data sets. They use wavelet coefficients as compact information representation and correlation structure detection, and then apply a linear regression model in the wavelet domain.

Aggarwal et al. have adopted the idea of micro-clusters introduced in CluStream in On-Demand classification [3] and it shows a high accuracy. The technique uses clustering results to classify data using statistics of class distribution in each cluster.

Last [41] has proposed an online classification system that can adapt to concept drift. The system re-builds the classification model with the most recent examples. Using the error rate as a guide to concept drift, the frequency of model building and the window size are adjusted. The system uses info-fuzzy techniques for model building and information theory to calculate the window size.

Ding et al. [14] have developed a decision tree based on Peano count tree data structure. It has been shown experimentally that it is a fast building algorithm that is suitable for streaming applications.

Gaber et al. [21] have developed Lightweight Classification *LWClass*. It is a variation of *LWC*. It is also an AOG-based technique. The idea is to use K-nearest neighbors with updating the frequency of class occurrence given the data stream features. In case of contradiction between the incoming stream and the

stored summary of the cases, the frequency is reduced. In case of the frequency is equalized to zero, all the cases represented by this class is released from the memory.

3.3 Frequency Counting

Giannella et al. [20] have developed a frequent itemsets mining algorithm over data stream. They have proposed the use of tilted windows to calculate the frequent patterns for the most recent transactions based on the fact that users are more interested in the most recent transactions. They use an incremental algorithm to maintain the FP-stream which is a tree data structure to represent the frequent itemsets. They conducted a number of experiments to prove the algorithm efficiency.

Manku and Motwani [43] have proposed and implemented an approximate frequency counts in data streams. The implemented algorithm uses all the previous historical data to calculate the frequent patterns incrementally.

Cormode and Muthukrishnan [13] have developed an algorithm for counting frequent items. The algorithm uses group testing to find the hottest k items. The algorithm is used with the turnstile data stream model which allows addition as well as deletion of data items. An approximation randomized algorithm has been used to approximately count the most frequent items. It is worth mentioning that this data stream model is the hardest to analyze. Time series and cash register models are computationally easier. The former does not allow increments and decrements and the later one allows only increments.

Gaber et al. [21] have developed one more AOG-based algorithm: Lightweight frequency counting *LWF*. It has the ability to find an approximate solution to the most frequent items in the incoming stream using adaptation and releasing the least frequent items regularly in order to count the more frequent ones.

3.4 Time Series Analysis

Indyk et al. [36] have proposed approximate solutions with probabilistic error bounding to two problems in time series analysis: relaxed periods and average trends. The algorithms use dimensionality reduction sketching techniques. The process starts with computing the sketches over an arbitrarily chosen time window and creating what so called sketch pool. Using this pool of sketches, relaxed periods and average trends are computed. The algorithms have shown experimentally efficiency in running time and accuracy.

Perlman and Java [49] have proposed a two phase approach to mine astronomical time series streams. The first phase clusters sliding window patterns of each time series. Using the created clusters, an association rule discovery technique is used to create affinity analysis results among the created clusters of time series.

Zhu and Shasha [54] have proposed techniques to compute some statistical measures over time series data streams. The proposed techniques use discrete Fourier transform. The system is called StatStream and is able to compute approximate error bounded correlations and inner products. The system works over an arbitrarily chosen sliding window.

Lin et al. [42] have proposed the use of symbolic representation of time series data streams. This representation allows dimensionality/numerosity reduction. They have demonstrated the applicability of the proposed representation by applying it to clustering, classification, indexing and anomaly detection. The approach has two main stages. The first one is the transformation of time series data to Piecewise Aggregate Approximation followed by transforming the output to discrete string symbols in the second stage.

Chen et al. [12] have proposed the application of what so called regression cubes for data streams. Due to the success of OLAP technology in the application of static stored data, it has been proposed to use multidimensional regression analysis to create a compact cube that could be used for answering aggregate queries over the incoming streams. This research has been extended to be adopted in an undergoing project Mining Alarming Incidents in Data Streams *MAIDS*.

Himberg et al. [33] have presented and analyzed randomized variations of segmenting time series data streams generated onboard mobile phone sensors. One of the applications of clustering time series discussed: Changing the user interface of mobile phone screen according to the user context. It has been proven in this study that Global Iterative Replacement provides approximately an optimal solution with high efficiency in running time.

Guralnik and Srivastava [29] have developed a generic event detection approach of time series streams. They have developed techniques for batch and online incremental processing of time series data. The techniques have proven efficiency with real and synthetic data sets.

4- Systems

Several applications have stimulated the development of robust streaming analysis systems. The following represents a list of such applications.

- Burl et al. [9] have developed *Diamond Eye* for NASA and JPL. The aim of this project to enable remote computing systems as well as observing

scientists to extract patterns from spatial objects in real time image streams. The success of this project will enable "a new era of exploration using highly autonomous spacecraft, rovers, and sensors" [9]. This project represents an early development in streaming analysis applications.

- Kargupta et al. [37] have developed the first ubiquitous data stream mining system: *MobiMine*. It is a client/server PDA-based distributed data stream mining application for stock market data. It should be pointed out that the mining component is located at the server side rather than the PDA. There are different interactions between the server and PDA till the results finally displayed on the PDA screen. The tendency to perform data mining at the server side has been changed with the increase of the computational power of small devices.
- Kargupta et al. [38] have developed Vehicle Data Stream Mining System (*VEDAS*). It is a ubiquitous data mining system that allows continuous monitoring and pattern extraction from data streams generated on-board a moving vehicle. The mining component is located at the PDA on-board the moving vehicle. Clustering has been used for analyzing the driver behavior.
- Tanner et al. [51] have developed EnVironment for On-Board Processing (*EVE*). The system mines data streams continuously generated from measurements of different on-board sensors in astronomical applications. Only interesting patterns are transferred to the ground stations for further analysis preserving the limited bandwidth. This system represents the typical case for astronomical applications. Huge amounts of data are generated and there is a need to analyze this streaming information at real time.
- Srivastava and Stroeve [50] have developed a NASA project for onboard detection of geophysical processes represented in snow, ice and clouds using kernel clustering methods. These techniques are used for data compression. The motivation of the project is to preserve the limited bandwidth needed to send image streams to the ground centers. The kernel methods have been chosen due to its low computational complexity in such resource-constrained environment.

5- Research Issues

Data stream mining is a stimulating field of study that has raised challenges and research issues to be addressed by the database and data mining communities. The following is a discussion of both addressed and open research issues [17, 21, 26, 37]. The following is a brief discussion of previously addressed issues:

Handling the continuous flow of data streams: this is a data management issue. Traditional database management systems are not capable of dealing with such continuous high data rate. Novel indexing, storage and querying techniques are required to handle this non-stopping fluctuated flow of information streams.

Minimizing energy consumption of the mobile device: Large amounts of data streams are generated in resource-constrained environments. Senor networks represent a typical example. These devices have short-life batteries. The design of techniques that are energy efficient is a crucial issue given that sending all the generated stream to a central site is energy inefficient in addition to its lack of scalability problem [8].

Unbounded memory requirements due to the continuous flow of data streams: machine learning techniques represent the main source of data mining algorithms. Most of machine learning methods require data to be resident in memory while executing the analysis algorithm. Due to the huge amounts of the generated streams, it is absolutely a very important concern to deign space efficient techniques that can have only one look or less over the incoming stream.

Required result accuracy: design a space and time efficient techniques should be accompanied with acceptable result accuracy. Approximation algorithms as mentioned earlier can guarantee error bounds. Also sampling techniques adopt the same concept as it has been used in *VFML* [16].

Transferring data mining results over a wireless network with a limited bandwidth: knowledge structure representation is another essential research problem. After extracting models and patterns locally from data stream generators, it is essential to transfer these structures to the user. Kargupta et al. [37] have addressed this problem by using Fourier transformations to efficiently send mining results over limited bandwidth links.

Modeling changes of mining results over time: in some cases, the user is not interested in mining data stream results, but how these results are changed over time. If the number of clusters generated for example is changed, it might represent some changes in the dynamics of the arriving stream. Dynamics of data streams using changes in the knowledge structures generated would benefit many temporal-based analysis applications.

Developing algorithms for mining results' changes: this is related to the previous issue. Traditional data mining algorithms do not produce any results that show the change of the results over time. This issue has been addressed in *MAIDS* [10].

Visualization of data mining results on small screens of mobile devices: visualization of traditional data mining results on a desktop is still a research issue. Visualization in small screens of a PDA for example is a real challenge. Imagine a businessman and data are being streamed and analyzed on his PDA. Such results should be efficiently visualized in a way that enables

him to take a quick decision. This issue has been addressed in [37]

The above are the addressed research issues in mining data streams. Open Issues in the field are discussed in the following:

Interactive mining environment to satisfy user requirements: mining data streams is a highly application oriented field. The user requirements are considered a vital research problem to be addressed.

The integration between data stream management systems [4, 40] and the ubiquitous data stream mining approaches: it is an essential issue that should be addressed to realize a fully functioning ubiquitous mining. The integration among storage, querying, mining and reasoning over streaming information would realize robust streaming systems that could be used in different applications. Current database management systems have achieved this goal over static stored datasets.

The needs of real world applications: the relationship between the proposed techniques and the needs of the real world applications is another important issue. Some of the proposed techniques attempt to improve computational complexity of the mining algorithms with some margin error without taking care to the real needs of the applications that will use the proposed approach. Since data mining is an applied scientific discipline, the requirements of the applications should be stated clearly in order to achieve the analysis objectives.

Data stream pre-processing: the data pre-processing in the stream mining process should also be taken into consideration. That is how to design a light-weight pre-processing techniques that can guarantee quality of the mining results. Data pre-processing consumes most of the time in the knowledge discovery process. The challenge here is to automate such a process and integrate it with the mining techniques.

Model overfitting: the overfitting problem in data stream has not been addressed so far in the literature. Using some techniques such as cross validation is very costly in the case of data streams. Novel techniques are required to avoid model overfitting.

Data stream mining technology: the technological issue of mining data streams is also an important one. How to represent the data in such an environment in a compressed way? And which platforms are best to suit such special real-time applications? Hardware issues are of special concerns. Small devices are not designed for complex computations. Currently emulators are used to do this task and it is a real burden over data stream mining applications that run in resource-constrained environments. Novel hardware solutions are required to address this issue

The formalization of real-time accuracy evaluation: that is to provide the user by a feedback by the current achieved accuracy with relation to the available resources and being able to adjust according to the available resources.

The data stream computing formalization: mining of data streams is required to be formalized within a theory of data stream computation [32]. This formalization would facilitate the design and development of algorithms based on a concrete mathematical foundation. Approximation techniques and statistical learning theory represent the potential basis for such a theory. Approximation techniques could provide the solution, and using statistical learning theory would provide the loss function of the mining problem.

The above issues represent the grand challenges to the data mining community in this essential field. There is a real need inspired by the potential applications in astronomy and scientific laboratories [23] as well as business applications to address the above research problems.

6- Summary

The dissemination of data stream phenomenon has necessitated the development of stream mining algorithms. The area has attracted the attention of data mining community. The proposed techniques have their roots in statistics and theoretical computer science. Data-based and task-based techniques are the two categories of data stream mining algorithms. Based on these two categories, a number of clustering, classification, frequency counting and time series analysis have been developed. Systems have been implemented to use these techniques in real applications.

Mining data streams is still in its infancy state. Addressed along with open issues in data stream mining are discussed in this paper. Further developments would be realized over the next few years to address these problems. Having these systems that address the above research issues developed, that would accelerate the science discovery in physical and astronomical applications [23], in addition to business and financial ones [38] that would improve the real-time decision making process.

References

[1] 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.

[2] 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.

[3] 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.
[4] A. Arasu, B. Babcock. S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein, and J. Widom. STREAM: The Stanford Stream Data Manager Demonstration description - short overview of system status and plans; in Proc. of the ACM Intl Conf. on Management of Data, June 2003.
[5] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In Proceedings of PODS, 2002.
[6] 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
[7] 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
[8] R. Bhargava, H. Kargupta, and M. Powers, Energy Consumption in Data Analysis for On-board and Distributed Applications, Proceedings of the ICML'03 workshop on Machine Learning Technologies for Autonomous Space Applications, 2003.
[9] M. Burl, Ch. Fowlkes, J. Roden, A. Stechert, and S. Mukhtar, Diamond Eye: A distributed architecture for image data mining, in SPIE DMKD, Orlando, April 1999.
[10] 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.
[11] 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.
[12] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-Dimensional Regression Analysis of Time-Series Data Streams In VLDB Conference, 2002.
[13] G. Cormode, S. Muthukrishnan What's hot and what's not: tracking most frequent items dynamically. PODS 2003: 296-306
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] V. Ganti, Johannes Gehrke, Raghu Ramakrishnan: Mining Data Streams under Block Evolution. SIGKDD Explorations 3(2), 2002.
[19] M. Garofalakis, Johannes Gehrke, Rajeev Rastogi: Querying and mining data streams: you only get one look a tutorial. SIGMOD Conference 2002: 635
[20] 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.
[21] Gaber, M, M., Krishnaswamy, S., and Zaslavsky, A., On-board Mining of Data Streams in Sensor Networks, Accepted as a chapter in the forthcoming book Advanced Methods of Knowledge Discovery from Complex Data, (Eds.) Sanghamitra Badhyopadhyay, Ujjwal Maulik, Lawrence Holder and Diane Cook, Springer Verlag, to appear
[22] Gaber, M, M., Zaslavsky, A., and Krishnaswamy, S., A Cost-Efficient Model for Ubiquitous Data Stream Mining, the Tenth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Perugia Italy, July 4-9.
[23] Gaber, M, M., Zaslavsky, A., and Krishnaswamy, S., 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.
[24] Gaber, M, M., Zaslavsky, A., and Krishnaswamy, S., Resource-Aware Knowledge Discovery in Data Streams, the Proceedings of First International Workshop on Knowledge Discovery in Data Streams, to be held in conjunction with the 15th European Conference on Machine Learning and the 8th European Conference on the Principals and Practice of Knowledge Discovery in Databases, Pisa, Italy, 2004.
[25] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, M. Strauss: One-Pass Wavelet Decompositions of Data Streams. TKDE 15(3), 2003
[26] L. Golab and M. T. Ozsu. Issues in Data Stream Management. In SIGMOD Record, Volume 32, Number 2, June 2003.
[27] 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.
[28] 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.
[29] V. Guralnik and J. Srivastava. Event detection from time series data. In ACM KDD, 1999.
[30] D. J. Hand, Statistics and Data Mining: Intersecting Disciplines *ACM SIGKDD Explorations*, 1, 1, pp. 16-19, June 1999.
[31]Hand D.J., Mannila H., and Smyth P. (2001) *Principles of data mining*, MIT Press.
[32] M. Henzinger, P. Raghavan and S. Rajagopalan, Computing on data streams , Technical Note 1998-011, Digital Systems Research Center, Palo Alto, CA, May 1998
[33] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmki, and H.T.T. Toivonen. Time series segmentation for context recognition in mobile devices. In Proceedings of the 2001 IEEE International Conference on Data Mining, pp. 203-210, San Jos, California, USA, 2001.
[34] Hoffmann F., Hand D.J., Adams N., Fisher D., and Guimaraes G. (eds) (2001) *Advances in Intelligent Data Analysis*. Springer.
[35] G. Hulten, L. Spencer, and P. Domingos. Mining Time-Changing Data Streams. ACM SIGKDD 2001.
[36] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying Representative Trends in Massive Time Series Data Sets Using Sketches. In Proc. of the 26th Int. Conf. on Very Large Data Bases, Cairo, Egypt, 2000.
[37] Kargupta, H., Park, B., Pittie, S., Liu, L., Kushraj, D. and Sarkar, K. (2002). MobiMine: Monitoring the Stock Market from a PDA. ACM SIGKDD Explorations. January 2002. Volume 3, Issue 2. Pages 37-46. ACM Press.
[38] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa, and D. Handy, VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring, Proceedings of SIAM International Conference on Data Mining, 2004.
[39] E. Keogh, J. Lin, and W. Truppel. Clustering of Time Series Subsequences is Meaningless: Implications for Past and Future Research. In proceedings of the 3rd IEEE International Conference on Data Mining. Melbourne, FL. Nov 19-22, 2003.
[40] S. Krishnamurthy, S. Chandrasekaran, O. Cooper, A. Deshpande, M. Franklin, J. Hellerstein, W. Hong, S. Madden, V. Raman, F. Reiss, and M. Shah. TelegraphCQ: An Architectural Status Report. IEEE Data Engineering Bulletin, Vol 26(1), March 2003.
[41] M. Last, Online Classification of Nonstationary Data Streams, Intelligent Data Analysis, Vol. 6, No. 2, pp. 129-147, 2002.
[42] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13, 2003.
[43] 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.
[44] S. Muthukrishnan (2003), Data streams: algorithms and applications. Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms.
[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] C. Ordonez. Clustering Binary Data Streams with K-means ACM DMKD 2003.
[47] B. Park and H. Kargupta. Distributed Data Mining: Algorithms, Systems, and Applications. To be published in the Data Mining Handbook. Editor: Nong Ye. 2002.
[48] S. Papadimitriou, C. Faloutsos, and A. Brockwell, Adaptive, Hands-Off Stream Mining, 29th International Conference on Very Large Data Bases VLDB, 2003.
[49] E. Perlman and A. Java. Predictive Mining of Time Series Data in Astronomy. In ASP Conf. Ser. 295: Astronomical Data Analysis Software and Systems XII, 2003.
[50] A. Srivastava and J. Stroeve, Onboard Detection of Snow, Ice, Clouds and Other Geophysical Processes Using Kernel Methods, Proceedings of the ICML'03 workshop on Machine Learning Technologies for Autonomous Space Applications
[51] S. Tanner, M. Alshayeb, E. Criswell, M. Iyer, A. McDowell, M. McEniry, K. Regner, EVE: On-Board Process Planning and Execution, Earth Science Technology Conference, Pasadena, CA, Jun. 11 - 14, 2002
[52] N. Tatbul, U. Cetintemel, S. Zdonik, M. Cherniack, M. Stonebraker. Load Shedding on Data Streams, In Proceedings of the Workshop on Management and Processing of Data Streams, San Diego, CA, USA, June 8, 2003.
[53] 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.
[54] Y. Zhu and D. Shasha. StatStream: Statistical monitoring of thousands of data streams in real time. In VLDB 2002, pages 358-369.