MINING TECHNIQUES FOR STREAMING DATA

ManaL Mansour and Manal Abdullah

Department of Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT

The huge explosion in using real time technology leads to infinite flow of data which known as data streams. The characteristics of streaming data require different techniques for processing due its volume, velocity and volatility, beside issues related to the limited storage capabilities. Hence, this research highlights the significant aspects to consider when building a framework for mining data streams. It reviews the methods for data stream summarizing and creating synopsis, and the approaches of processing these data synopses. The goal is to present a model for mining the streaming data which describes the main phases of data stream manipulation.

KEYWORDS

Data Stream, Data Mining, Data Synopsis, Micro-clusters.

1. INTRODUCTION

The amounts of the existed data globally almost reached to 64 zettabytes by the end of 2020, and the forecasts indicate that it will grow to 175 zettabytes by the end of 2025 [1].This is encouraging the data mining to become an integral part in many sectors such as businesses, financial, medical, manufacturing, media, and many others. It helps to extract knowledge that improve several processes like decision making, goals defining, customer personalizing, decreasing risk and fraud.

The great interest in data mining leads to produce new data mining techniques that could process huge infinite data flows that come from different sources such as sensors, network traffic, call center records, social media posts, ATM and credit card operations, web searches, etc. All these sources produce continuous sequences of massive data that generated in high speed rate and known as data stream [2].

A data stream consists of infinite sequence of elements that keep flowing with high speed rate. Every element of a stream is a pair of (Xi,Ti), where Xi is a dimensional vector arrives at time stamp Ti. The time stamp Identifies when the element occurred [3]. Mining such type of data streams is a continuous process of extracting knowledge and valuable insights from infinite and rapid data signals. The main goal is to predict the class or the value of the new elements using machine learning techniques such as clustering, classification, regression, etc [2].

Data stream is much related to the Big Data concept, since the big data comes as result of streaming the data. The continuous flow of fast data causes too large volume of data which is known as big data. It is important to clarify that streaming data growth and produce big data, but it is not necessary that all big data comes from data stream. Big data usually reflects data at rest mode even if it was petabytes while streaming data reflects data that in motion mode which need a different technology to handle it [3].

DOI:10.5121/ijdkp.2022.12201

Data mining techniques work differently with streaming data because of their different characteristics such as data volume which keep increasing continually, data volatility which reflects the different types of data values, and data velocity that reach very high ranges. Therefore, the system requires specific methods to summarize, storing, and processing the data stream [2]. Table 1 shows a comparison of mining traditional and streaming data [2].

Table 1. Comparison of Mining Traditional and Streaming data

Comparison Aspects	**Traditional Data**	**Streaming Data**
Data vitality	Static data	Dynamic data
Data size	Fixed size	Unlimited size
Number of data access	Multiple access	Single~few access
Processing procedure	Batch processing: Process all points in the data set	Online Processing: Process samples of data stream
Processing time	Unlimited time	Restricted time
Memory usage	Unlimited storage	Restricted storage
Results accuracy	Accurate results	Approximated results

2. DATA SYNOPSIS

Data streams can be generated using different types of artificial generators. Artificial generatoris an embedded component in the mining model used to create a sequence of data simulating the properties of the real data streams such speed, noise, density, etc. There are many existed data streams generators such as: Random RBF Generator [4], SEA Concepts Generator [4], AGRAWAL Generator [5], LED Generator [4], Hyperplane Generator [4], and Random Tree Generator [5].

The characteristics of streaming data make it difficult to store all the data in a limited memory. Instead, mining frameworks construct synopsis as a summary of the data stream. These synopses can be constructed in different ways: randomly, by average, by trends, by most recent, etc. Constructing the synopsis is a periodic online process which selects a subgroup of the data from the running stream. Despite that using synopsis decreases the amount of processed data and the analysis costs, it increases the possible errors. This problem makes it critical to choose the suitable method for constructing the synopsis [6]. Next sections will review some of the main methods of constructing the synopsis from the data stream.

2.1. Sampling

Sampling is one of the simplest methods for constructing synopsis from data streams. The process of sampling the data stream depends on taking a subgroup of data elements that estimate the whole statistical factors for streaming data, like mean, variance, probability distribution, etc[2]. Figure 1 shows an example of sampling the streaming data with a simple rule of choosing the data every four received data points.

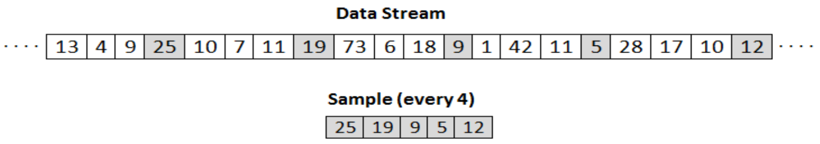
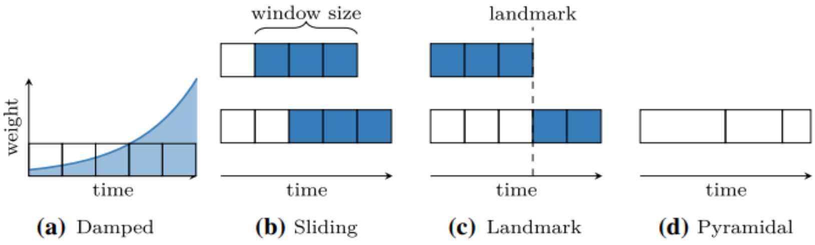


Figure 1. Data Stream Sampling

There are many sampling methods and the challenge is to choose the one which produces a sample that reflect the statistics of the original data. Random sampling is the simplest form, where all data instances have equal probability to be included in the sample. One technique for random sampling is the reservoir algorithm [7] which randomly collects data sample of size k and keep updating this sample. Every new instance in the data flow has a probability p=1/n to replace an old instance, where n is the number of elements that been generated from the stream. Other sampling techniques include systematic sampling [8], stratified sampling [9], and cluster sampling [10].

2.2. Windowing

Data stream windowing is a process of segmenting the data stream into small packets at certain time t with specific size w [2]. Windowing approach helps to deals with concept drift which reflects the difference in the data stream nature through time. The content of the window continues to change and gives updated results whereas the element that arrives at time t is rejected at time t+w [11]. There are four main types of time windows [12]: damped window, sliding window, landmark window, and tilted window (pyramidal). Figure 2 describe the windows types and how it works. All windows commonly update the data periodically, but they differ in the updating procedure as described next sections.



(a) Damped **(b)** Sliding **(c)** Landmark **(d)** Pyramidal

Figure 2. Types of Data Stream Windowing [11]

2.2.1. Damped Window

This type of windows gives specific weight value for each arrived data element based on time. New data elements have higher weight values which keep decreasing over time based on ageing function. Ageing function used to update the window by using fading factor (λ) which controls the rate of weight decreasing. All the data are involved in the processing, but with different weight values which means different importance levels. Equation 1 presents the exponential fading function, where the current time is denoted by tc and the point creation time is denoted by t0 [12].

$$f(t) = 2\lambda\ (tc-t0) \qquad (1)$$

2.2.2. Sliding Window

This type of windows is more suitable for applications that interested in the newest information because it considers only the recent data of the stream [12]. The window works based on the principle (First-In-First-Out), where the old data are removed to add the new data. All the data elements in the updated window have the same weight in processing which mean having the same importance. The small size windows give better accuracy in high changing environment with risks of data drifts, while the larger size works better in the stable data environment [11].

2.2.3. Landmark Window

This type of windows works by segmenting the data stream into chunks based on landmarks. Landmark could reflect the data events or the passed time. For example, the landmark defined as the starting point that could be (every 100 elements) which means updating the window after receiving 100 new data points and removed all the previous 100 data points. The window includes a summary of all data points that came after the landmark and gives them the same level of importance [11] [12].

2.2.4. Pyramidal Window

This type of windows also known as tilted windows and it includes data from different times with different detailing levels. The new coming data points are included to the window and aggregated with the points received before. In contrast of sliding window, titled window does not completely discard the old data, but integrate all the data and give more importance to the recent data [11] [12].

2.3. Histograms

Histogram is a popular method for constructing synopsis of data stream which depend on capturing the frequency distribution of the data values received from the stream. Instead of store the frequency for each data value, the domain of data values are divided into subsets called buckets. Then, the average frequency is computed and stored for each bucket [2].

Different types of data stream histograms are developed with different manners of dividing the data domain into buckets. The histogram with less variance is better, but it requires more space capabilities to implants. The histogram variance defined as the weighted sum of the original values for each bucket where the weight of each bucket equals the number of its included values [13] [14]. There are five types of histograms used to store data distributions which presented in figure 3.

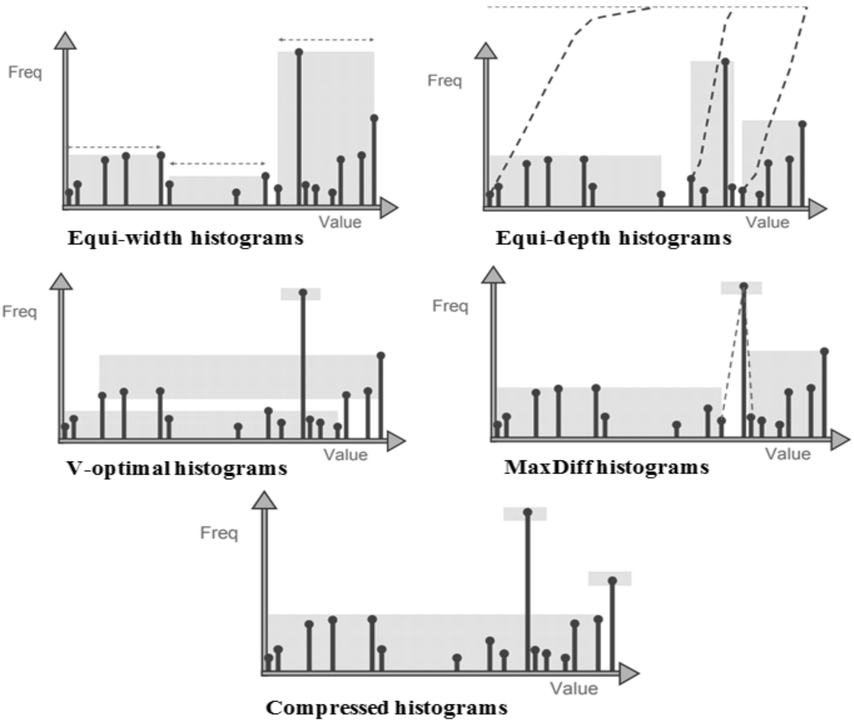


Figure 3. Types of Data stream Histograms [14]

2.3.1. Equi-width histograms

This type works by dividing the data domain into equal width buckets which means that every bucket has the same range of data pints numbers. It reduces the storage requirements and works well with simple skewed data distribution, but it is difficult to estimate the errors [13].

2.3.2. Equi-depth Histograms

This type works by dividing the data domain into similar height buckets which mean that data points in certain bucket have almost the same frequencies. It requires the same storage of Equi-width histograms, but it is more complex to maintain. It Works well with low skewed data distribution and gives better accuracy than Equi-width histograms, but variance within a bucket may still high [13].

2.3.3. Variance-optimal Histograms

This type works by dividing the data domain into buckets where the data frequencies in every bucket is either all greater than or all less than the frequencies of other buckets. It works well with high skewed data distributions and gives the minimum histogram variance, but it requires high memory capacities [14].

2.3.4. MaxDiff Histograms

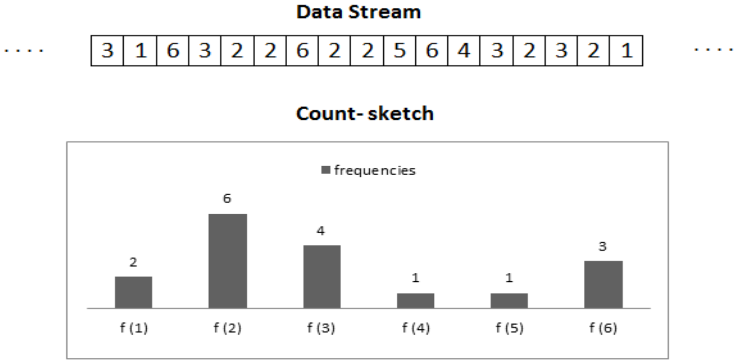
This type works by dividing the data points with the highest frequencies and lowest frequencies in individual separated buckets, while the data with middle frequencies are all combined with one bucket. It stores only the frequencies of the attribute values in the individual buckets which require smaller storage capacities than Variance-optimal histograms [14].

4.3.5. Compressed Histograms

This type works by dividing the data points with high frequencies in singleton buckets while the rest data is divided as equi-depth histogram. It gives great accuracy with skewed data distributions [14].

2.4. Sketching

The data sketching is a process of creating a structure of the data stream. The data sketch includes some information about data samples that help to response some predefined questions [15]. The stored information could be the number of the received values appears in the stream without repetition like introduced in Flajolet Martin Algorithm (FM Sketch) [16], the most frequent elements seen in the stream like introduced in Count Sketch Algorithm [17], or frequencies of data events like introduced in Count–Min Algorithm (CM Sketch) [18]. The task of recording values frequencies is shared in CM sketching and histograms techniques discussed before, but histogram considers the frequencies of specific data sample, while CM sketching updates the frequencies of a data sample over the stream [15]. Figure 4 shows an example of count sketching in top view.



Data Stream

....	3	1	6	3	2	2	6	2	2	5	6	4	3	2	3	2	1	....

Count- sketch

Figure 4. Count Sketching for Data Stream

Sketching is much related to sampling technique since it uses some initials samples as basis to build the sketch. Every coming data point will be discarded after being used to update the sketch which means not storing all the data dimensions, but only the interested dimensions. This technique presents the sketching as a powerful method to compress data stream with high dimensions [15].

3. DATA STREAM PROCESSING

Data stream processing is a continuous operational techniques applied on the live data to give direct results [3]. Figure 5 shows the basic approaches of processing data stream: supervised learning, unsupervised learning. The main difference is about using labelled data or not. All approaches that depend on labelled data are assumed as supervised where each input is assigned to a predefined label based on knowledge comes from training data examples. In contrast, the approaches that depend on unlabelled data are assumed as unsupervised where the main tasks are about finding the similarities between data points. Some approaches assumed as semi supervised since it uses both types: some labelled data at the early stages and then continue the process with unlabeled data [19].

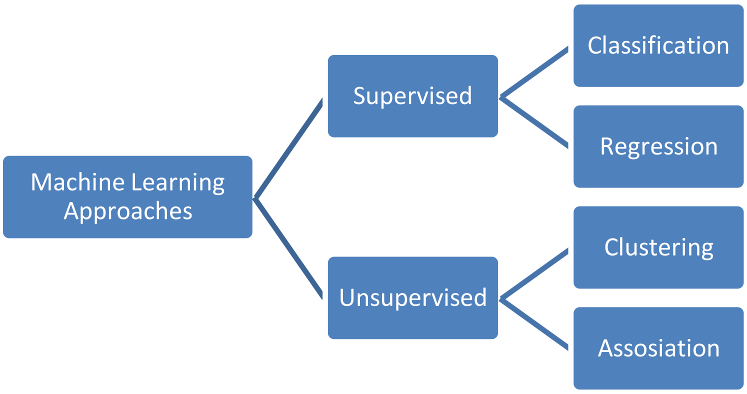
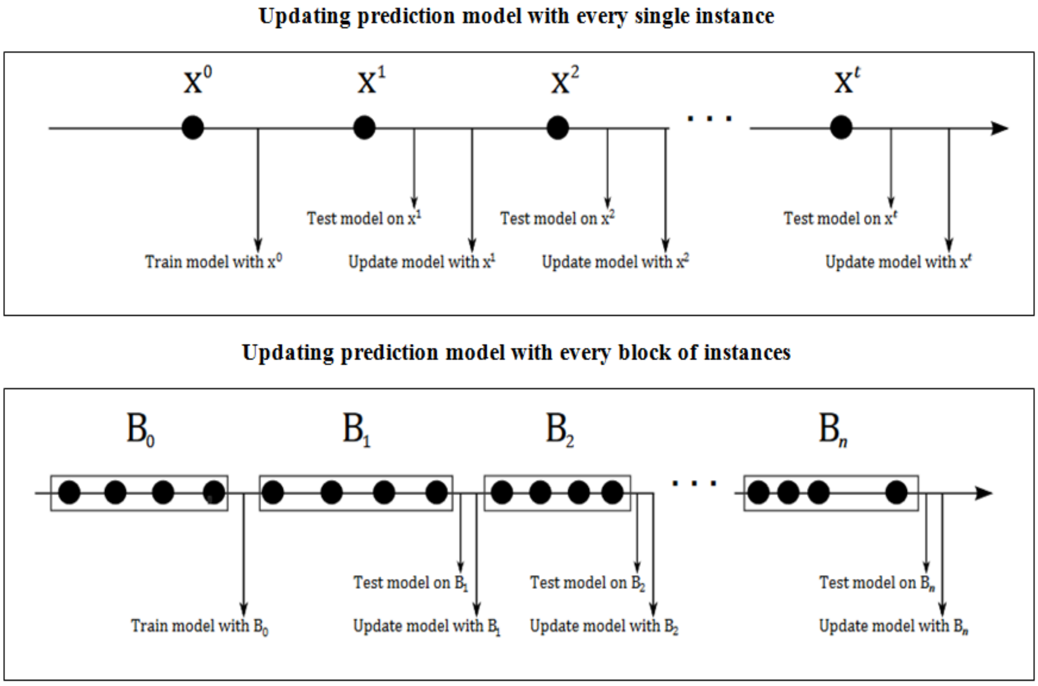


Figure 5. Machine Learning Approaches

3.1. Classification

Classification is a process of data organizing and grouping based on predefined classes which also called labels, For example, predicting the type of fruit. The process consists of two phases: training and prediction which are clearly separated in case of batch processing for regular and static data. In training phase, the dataset is divided into two sets: training and testing to ensure the accuracy. Then, the prediction phase starts the actual classification process.

In contrast of static data, classifying streaming data doesn't have a clear cut between training and prediction phases. Training, evaluating, and testing are overlapped since more data is arriving continuously and the prediction phase starts before receiving all data. The algorithm predicts the labels of the received data based on prediction model and continues to update and adjust that model while receiving new data [3]. Figure 6 shows how some approaches update the model with every single received data instance, while some others update the model with every block of data points [20].



Updating prediction model with every single instance

Updating prediction model with every block of instances

Figure 6. Updating Prediction Model of Data Stream [20]

Several classifiers were developed to solve the problem of predicting the data stream such as: Lazy classifier [21]. Decision Trees [22]which works by building a tree structure using training examples that used to predict the labels of other coming examples, and Naïve Bays [23] which is a simple classifier works based on the use of the Bayes' theorem.

3.2. Regression

The regression classifier aims for predicting the label value of an example where data labels is not yet known, and then classify the data to the closest label. The predicted label is a real value instead of being one of the discrete set of predefined values, For example, predicting the house price which will assign each house to the closet price label. Some classification algorithms include natural counterparts for regression, such as: lazy learning and decision trees [3].

3.3. Clustering

Clustering is a process of finding homogeneous groups or clusters among unlabeled data. It is a useful technique in some problems such as segmenting the customers to find the targeted groups, and finding communities in social networks [2].

Clustering approaches for data streams consist of two steps: live data phase and offline data phase. In the live data phase, the algorithm performs one pass over some current data streams to computes a specific number of micro-clusters, store it in memory, and update it using new subsets of the coming new data. In the offline phase, the algorithm has several passes only over the stored micro-clusters to processes them using traditional batch clustering techniques [3]. Next subsections will describe some of the common algorithms for clustering the streaming data.

3.3.1. Hierarchical Clustering

This approach works by grouping the data into a tree of clusters such as Clustree Algorithm [24], and Brich Algorithm [25].

3.3.2. Partitioning Clustering

This approach works by grouping the data into number of partitions which reflects the clusters. It works based on distance functions like k-means which used in StreamKM++ Algorithm [26].

3.3.3. Density Based Clustering

This approach is based on density functions which assume dense areas of data points as clusters. It requires two values for the cluster: the radius size and the minimal number of data points to include in the cluster. DenSteam Algorithm [27] is a popular example of this approach which works based on summarizing clusters using dense micro-clusters called core-micro-cluster.

3.3.4. Grid Based Clustering

This approach is based on computing the grid density. D-Stream Algorithm [28] is an example which works by segmenting the data space into finite number of cells assumed as grid structure. Then, mapping the recorded data to that grid which will be clustered based on their density.

3.3.5. Growing Neural Gas Clustering

The growing neural gas is a clustering approach works by devising the data space into units depending on the densest areas. It produces a two-dimensional presentation for the clusters and the data patterns [29].

3.4. Association

Finding association rules is a process of finding frequent pattern among data such as trees, graphs, item sets, etc. It is useful to describe the structure of the data and explore its features which could enhance other processes of classification or clustering. Most algorithms of finding frequent items for data streams includes batch miner and streaming techniques which produce approximate results [2]. IncMine algorithm [30] is one of the algorithms developed to find the frequent items among streaming data. It works by updating the model incrementally over a sliding window.

4. REAL APPLICATIONS AND TOOLS

The importance of data stream analytics increased as a result of turning most organizations and business towards digital work and applications which produce real time data. This is encourages to develop mining tools to help get benefits from streaming data.

Table 2 shows several real applications of mining streaming data such as clinical decision support system which helps in disease diagnosis, choosing medicines and improving the patient care [31]. Applications also include the biological field such as using the satellite systems to study the earth nature. Moderate Resolution Imaging Spectroradiometer (MODIS) [32] is an example which processes the satellite signals to study the long-term change of vegetation.

Many other applications related to the social fields are developed such as mining the data stream of social media like Twitter, LinkedIn, Facebook, etc. All these types of platforms produce unstructured data streams which needs machine learning techniques to analyze users profiles, identify posts patterns, trends, and others insights. Novel metaheuristic method (CSK) proposed in [33] is an example of social applications which developed for sentiment analysis of Twitter. It works based on K-means method combined with cuckoo search technique to provide reviews conclusion for any social issues.

Marketing is one of the biggest beneficiaries' fields which use data stream analysis. Most markets are interested in studying the customers' behavior which could be done by segmenting the customers into groups based on their purchases baskets. Studying the customers' behavior helps to customize special offers for them. Sales markets also interested to study the relationships between different product items which help in designing the items catalog, deciding the profitable offers [3]. Other markets applications include Stock market prediction. One example of the applications used in on this field is using Neural Network and Decision Tree which proposed in [34]. This algorithm use data of markets sales, and the customers' interests to predict the future value of the companies' stock.

Table 2. Real Applications of Data Stream Mining

Application field	**Streaming data**	**Mining Objective**
Medical [31]	Health status records	Disease diagnosis, and improving the patient care
Biological [32]	Satellite signals	Study the long-term change of vegetation
Social [33]	Twitter posts	Provide reviews conclusion for any social issues
Marketing [34]	Sales data	Prediction of Stock Market

Many frameworks appear to process data streams which make it critical to choose the suitable one that meets the problem requirements. Waikato Environment for Knowledge Analysis (WEKA) [35], Massive Online Analysis (MOA) [3], Hadoop [36], Spark [37], Storm [38], and Flink [39] are the some popular processing frameworks for mining data streams.

Existing tools for mining data stream differ in their models wither it works with batch, real time processing, or both. Batch processing collects the data over specific period of time like payroll and billing data. In contrast, real time processing receives continual data like bank services and social media messages. Table 3 reviews the processing model, programing languages, and main functions of some existing tools for mining the data stream.

.

Table 3. Tools for Mining Data Stream

Tool	**Processing Model**	**Program language**	**Functions**
WEKA [35]	Hybrid: batch and real-time	Java-Python	Includes a large set of machine learning algorithms used in data mining operations
MOA [3]	Real time only	java	includes a several types of data streams generators, algorithms of machine learning , and tools for results evaluation
Hadoop [36]	Batch only	Java-Python and Scala	Provides distributed storage for big data processing based on MapReduce programming model.

Spark [37]	Hybrid: batch and real-time	Java-Python and Scala	General-purpose processing engine for both big and steaming data
Storm [38]	Real time only	Any programming language	Distributed framework for stream processing computation. Very fast and capable to integrate with Hadoop for immediate analytics
Flink [39]	Hybrid: batch and real-time	Java, Scala, Python and SQL	Distributed processing engine and a scalable data analytics framework. includes data stream API

5. Related Work

The data stream analysis is a rich field and always required for further improvements. Several contributions were adopted to create, develop, and improve the algorithms of mining data stream. The contributions in data stream algorithms may be through changing the initialization methods such proposed in [40], changing the method of constructing the synopsis of data stream or merging two methods such suggested in [41], integrating multiple methods of data stream processing as proposed in [42], or implementing the algorithm in a new field such produced [43]. Other related work and contributions were reviewed in table 4.

Table 4. Algorithms for Data Stream Mining

Algorithm	**Synopses Type**	**Processing Approach**	**Main Ideas**
Very Fast Decision Tree (VFDT) [31]	Samples	Classification	Using The VFDT with pointers to enhance the decision tree
Hyper-Ellipsoidal Clustering for Evolving data Stream (HECES) [40]	Sliding window	Clustering	Using covariance shrinkage for data estimation, and heuristic technique for the initial clusters
A Privacy-Preserving Distinct Counting Scheme for Mobile Sensing (PPDC) [41]	FM Sketches	Clustering	Integrating homomorphic data encrypting with Flajolet-Martin sketching
Scale-free social network (SFNClassifier) [42]	Sliding window	Classification	Integrating Adwin Algorithm with ensemble-based approaches
Clustering Algorithm for geographical data streams (GeoDenStream) [43]	Damped window	Clustering	Enhancing DenStream Algorithm to apply entity-based clustering on geographical spaces
Uncertain Data Stream Frequent Itemsets Mining (UFIM) [44]	Sliding window	Association	Using a global GT-tree to process and update the frequent items in the sliding windows
Adaptive Random Forest Algorithm (ARF) [45]	Samples	Classification	Implementing resampling method for detecting data drifts and other techniques like warning detection and background trees
Parsimonious Learning Machine Algorithm (PALM) [46]	Samples	Regression	Using upgraded rule of fuzzy method which developed based on the idea of hyperplane clustering
Augmented Sketch Algorithm (ASketch) [47]	Sketches	Association	Improving the Count-Min sketch by implementing a pre-filtering stage for data stream

6. Conclusion and Findings

Today's digital world emphasizes the important role of data streams analytics which helps to develop the operations and improve the performance in many fields. Despite the great evolution in data mining techniques, it still have difficulties to deal with the huge explosion of data comes from real time applications. Many applications were developed for mining the streaming data, and there are common phases in their architectures which are: data generating, sampling, initializing, and processing. These phases combined to define the general data streams model presented in Figure 7.

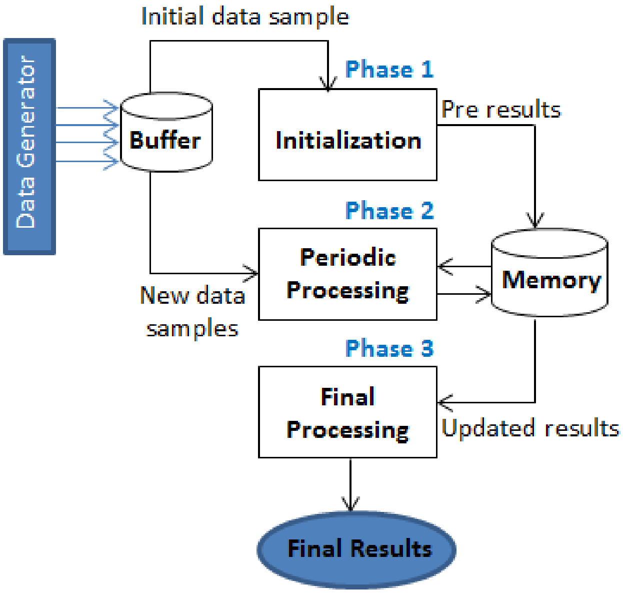


Figure 7. General Model for mining data streams

The general model presents the process of data streams manipulation following the common steps of the existing models. First, streaming data flows to the system continuously in massive volumes and high velocity. Since the system is unable to process the whole stream, it takes some initial data, processes it, produce approximated results as a summary, and store it in memory. Then, the system takes different samples from the new coming data periodically, and updates the initial results stored in memory. Finally, only the updated summaries will be processed to introduce the final results instead of using the raw data comes from the stream.

These Three phases offer good flexibility to explore the evolution nature of the data over different time periods which means handling the data streams over time rather than mining the whole stream at one time.

7. Recommendations

The process of mining the data stream keeps developing to enhance the results quality. There are some suggestions that are recommended to extend the work:

- Applying the model with real and artificial data generators
- Evaluating the model while raising the data stream speed and noise

- Enhancing the model by using other methods for data summarizing and creating synopsis.

REFERENCES

[1] "How Much Data Is Created Every Day," 28 October 2021. [Online]. Available: https://seedscientific.com/how-much-data-is-created-every-day/. [Accessed 18 12 2021].

[2] M. Garofalakis, J. Gehrke and R. Rastogi, in Data Stream Management: Processing High-Speed Data Streams, Springer, 2016.

[3] A. Bifet, R. Gavalda, G. Holmes and B. Pfahringer, Machine Learning for Data Streams:with Practical Examples in MOA, MIT Press, 2018.

[4] P. K. SRIMANI and M. M. PATIL, "Mining data streams with concept drift in massive online analysis frame work," WSEAS Trans. Comput, vol. 15, pp. 133-142, 2016.

[5] O. Wu, Y. S. Koh, G. Dobbie and T. Lacombe, "Transfer Learning with Adaptive Online TrAdaBoost for Data Streams," in Asian Conference on Machine Learning, 2021.

[6] Z. Shah, A. N. Mahmood, Z. Tari and A. Zomaya, "A Technique for Efficient Query Estimation," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, vol. 28, pp. 2770-2783, 2017.

[7] Gaber, Mohamed Medhat; Gama, Jo˜ao ; Krishnaswamy, Shonali; Gomes, Jo˜ao B´ artolo ; Stahl, Frederic;, "Data stream mining in ubiquitous environments: state-of-the-art and current directions," WIREs: Data Mining & Knowledge Discovery, vol. 4, no. 2, pp. 116-138, 2014.

[8] D. J. Brus and N. Saby, "Approximating the variance of estimated means for systematic random sampling, illustrated with data of the French Soil Monitoring Network," Elsevier, vol. 279, pp. 77-86, 2016.

[9] S. S. Ramkrishna and . P. S. Housila, "Efficient classes of estimators in stratified random," Statistical Papers, vol. 56, no. 1, pp. 83-103, 2015.

[10] W. Li , "Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph," IEEE transactions on multimedia, vol. 19, no. 2, pp. 367-381, 2017.

[11] M. Carnein and H. Trautmann , "Optimizing data stream representation: An extensive survey on stream clustering algorithms," Business & Information Systems Engineering, vol. 61, no. 3, p. 277–297, 2019.

[12] E. Ntoutsi, N. Pelekis and Y. Theodoridis, "An evaluation of data stream clustering algorithms," Statistical Analysis and Data Mining, vol. 11, no. 4, pp. 167-187, 2018.

[13] Y. Ioannidis, "The history of histograms (abridged)," Proceedings, pp. 19-30, 2003.

[14] J. Gama and T. Mendonça, "Constructing fading histograms from data streams," PROGRESS IN ARTIFICIAL INTELLIGENCE, vol. 3, no. 1, pp. 15-28, 2014.

[15] R. Jayaram, "Sketching and Sampling Algorithms for," Carnegie Mellon University Pittsburgh, Pittsburgh, 2021.

[16] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," Journal of computer and system sciences, vol. 31, no. 2, pp. 182-209, 1985.

[17] M. Charikar, K. Chen and M. Farach-Colton, "Finding frequent items in data streams," International Colloquium on Automata, Languages, and Programming, vol. 2380, pp. 693-703, 2002.

[18] D. Ting, "Count-Min: Optimal Estimation and Tight Error Bounds using Empirical Error Distributions," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.

[19] A. C. Gorgônio, A. . M. d. P. Canuto, K. M. O. Vale and F. . L. Gorgônio, "A semi-supervised based framework for data stream classification in non-stationary environments," International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2020.

[20] D. Brzezinski and J. Stefanowski, "Stream Classification.," in Encyclopedia of Machine Learning, Springer, 2016, pp. 1191-1199.

[21] J. Yuan, Z. Wang, Y. Sun, W. Zhang and J. Jiang, "An effective pattern-based Bayesian classifier for evolving data stream," Neurocomputing, vol. 295, pp. 17-28, 2018.

[22] D. Jankowski, K. Jackowski and B. Cyganek, "Learning Decision Trees from Data Streams with Concept Drift," Procedia computer science, vol. 80, pp. 1682-1691, 2016.

[23] M. Vadivukarassi, N. Puviarasan and P. Aruna, "Sentimental Analysis of Tweets Using Naive Bayes Algorithm," World Applied Sciences Journal, vol. 35 (1), pp. 54-59, 2017.

[24] J. Zgraja and M. Woźniak, “Drifted data stream clustering based on ClusTree algorithm,” International Conference on Hybrid Artificial Intelligence Systems, vol. 10870, pp. 338-349, 2018.

[25] G. Pitolli, L. Aniello, G. Laurenza, L. Querzoni and R. Baldoni, “Malware family identification with BIRCH clustering,” International Carnahan Conference on Security Technology , pp. 1-6, 2017.

[26] M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen and C. Sohler, “StreamKM++: A Clustering Algorithms for Data Streams,” Journal of Experimental Algorithmics (JEA), vol. 17, pp. 2-1, 2012.

[27] F. Cao, M. Estert, W. Qian and A. Zhou, “Density-Based Clustering over an Evolving Data Stream with Noise,” in Proceedings of the 2006 SIAM International Conference on Data Mining (SDM), 2006.

[28] M. H. Ali, A. Sundus, W. Qaiser, Z. Ahmed and Z. Halim, “Applicative implementation of D-stream clustering algorithm for the real-time data of telecom sector,” nternational conference on computer networks and information technology, pp. 293-297, 2011.

[29] M. Ghesmoune, M. Lebbah and H. Azzag, “A new Growing Neural Gas for clustering data streams,” Neural Networks, vol. 78, pp. 36-50, 2016.

[30] J. Cheng, Y. Ke and W. Ng, “Maintaining frequent closed itemsets over a sliding window,” Journal of Intelligent Information Systems, vol. 31, p. 191–215, 2007.

[31] Z. Yang , S. Fong, J. Fiaidhi and S. Mohammed, “Real-Time Clinical Decision Support System with Data Stream Mining.,” Journal of Biomedicine & Biotechnology, pp. p1-8, 8p, 2012.

[32] J. J. Maynard, J. W. Karl and D. M. Browning, “Effect of spatial image support in detecting long-term vegetation change from satellite time-series,” Landscape Ecology, vol. 31, no. 9, pp. p2045, 18 p, 2016.

[33] C. P. Avinash , “Twitter sentiment analysis using hybrid cuckoo search method,” Information processing & management, vol. 53, no. 4, p. 764, 2017.

[34] S. Tiwari and A. Gulati, “Prediction of Stock Market from Stream Data Time Series Pattern using Neural Network and Decision Tree,” Citeseer, 2011.

[35] E. Frank, . M. A. Hall and I. H, “Weka 3: Machine Learning Software in Java,” University of Waikato, 2016. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/. [Accessed 2022].

[36] A. Hadoop®, “What Is Apache Hadoop?,” hadoop.apache.org, 18 10 2018. [Online]. Available: http://hadoop.apache.org/. [Accessed 2 12 2018].

[37] “Apache Spark™ - Lightning-Fast Cluster Computing,” The Apache Software Foundation, 2 3 2017. [Online]. Available: https://spark.apache.org/. [Accessed 2 12 2018].

[38] “Apache Storm,” Apache Software Foundation, 3 6 2017. [Online]. Available: http://storm.apache.org/. [Accessed 2 12 2018].

[39] “Apache Flink: Introduction to Apache Flink®,” The Apache Software Foundation, 7 2 2017. [Online]. Available: https://flink.apache.org/introduction.html. [Accessed 2 12 2018].

[40] Z.-u. R. Muhammad, . L. Tianrui, Y. Yan and W. Hongjun , “Hyper-ellipsoidal clustering technique for evolving data stream,” Knowledge-Based Systems, vol. 3, no. 14, pp. 3-14, 2014.

[41] X. Yang , M. Xu, S. Fu and Y. Luo , “PPDC: A Privacy-Preserving Distinct Counting Scheme for Mobile Sensing,” Applied Sciences, vol. 9(18), p. 3695, 2019.

[42] J. P. Barddal, H. M. Gomes and F. Enembreck, “SFNClassifier: A scale-free social network method to handle concept drift,” in Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014.

[43] M. Li, A. Croitoru and S. Yue, “GeoDenStream: An improved DenStream clustering method for managing entity data within geographical data streams,” Computers & Geosciences, vol. 144, p. 104563, 2020.

[44] H. Liu, . K. Zhou, P. Zhao and S. Yao, “Mining frequent itemsets over uncertain data streams,” nternational Journal of High Performance Computing and Networking, vol. 11, pp. 312-321, 2018.

[45] H. . M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck and B. Pfharinger, “Adaptive random forests for evolving data stream classification,” Machine Learning, vol. 106, pp. 1469-1495, 2017.

[46] M. M. Ferdaus, M. Pratama, S. . G. Anavatti and M. A. Garratt, “Palm: An incremental construction of hyperplanes for data stream regression,” IEEE Transactions on Fuzzy Systems, vol. 27, no. 11, pp. 2115-2129., 2019.

[47] P. Roy, A. Khan and G. Alonso, “Augmented sketch: Faster and more accurate stream processing,” in Proceedings of the 2016 International Conference on Management of Data, 2016.