Frequent Pattern Mining over Streaming Data: From models to research challenges

Asmaa S.Abdo
Information System Department
Faculty of Computers and Artificial Intelligence
University of Sadat City
El Menoufia, Egypt
asmaa.saad@fcai.usc.edu.eg

Rashed K.Salem
Information System Department
Faculty of Computers and Information
Menoufia University
El Menoufia, Egypt
rsalem@ci.menofia.edu.eg

Hatem M.Abdul-Kader
Information System Department
Faculty of Computers and Information
Menoufia University
El Menoufia, Egypt
hatem.abdelkader@ci.menofia.edu

***Abstract*— Research in frequent pattern mining from streaming data becomes a pioneer in the field of information systems. The data stream is a continuous flow of data generated from different sources. Extracting frequent patterns from streaming data raises new challenges for the data mining community. We present an overview of the growing field of data streams. Many applications handle streaming data such as sensor networks, traffic management, log data, telephone call records, and social networks. These applications generate high volumes of streaming data with velocity, which is difficult to handle with traditional data mining techniques. This paper mainly reviewed different research algorithms, scientific practices, and methods that have been developed for mining frequent patterns from streaming data. In addition, it discusses well-known open-source software and tools for data stream mining, which are developing to handle streaming data. Finally, it summarizes the open issues and challenges to current existing approaches while handling and processing data streams in real-world applications.**

Keywords*— *Data streams, Frequent pattern mining, Stream data mining, Concept Drift, Window models.

I. Introduction

Data stream is an unbounded and continuous sequence of data received at a high-speed rate. Streaming data is considered one of the main sources of big data [1][2]. There is a wide range of applications on streaming data such as telecommunication calling records, web logs, network monitoring, video streams, etc [3]. Algorithms involved in processing these data sources have to deal with issues of streaming data, because existing traditional data mining algorithms are not designed to face them efficiently [4].

Data streams are classified into two types, online and offline data stream [5].

- **offline data streams**: It is generated by abruption or regular bulk comings, such as a bulk addition of recent instances in a data warehouse system.
- **online data streams**: It is generated from data updated in real-time, which comes one by one in time, such as transactions that are constantly generated from the network monitoring system.

The process of extracting interesting patterns from a sequence of transactions that arrive continuously with high speed is known as data stream mining. Mining frequent patterns from the streaming data have increased the number of research challenges in data mining society [6][7]. There are three main challenges in developing new algorithms to handle streaming data [8][9].

- First, design a fast-mining technique to process streaming data.
- Second, design an efficient algorithm that uses limited memory space in mining streaming data.
- Third, detect the distribution of data and changing concept drifts in a dynamic environment [10][11][12]. This phenomenon is commonly known as concept drift indicated in Fig. 1.

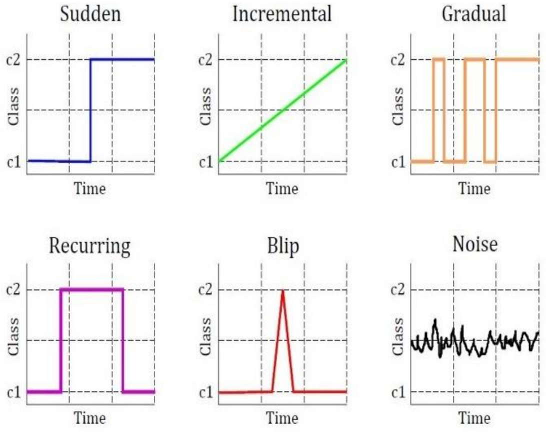

Fig. 1 Types of changes in streaming data [13].

There are many big challenges in mining streaming data in real applications. It is difficult to develop an application model for a specific purpose. The processing of data stream to find knowledge has been given great interest recently [8].

The general data stream mining procedure is indicated in Fig. 2. Data stream generators arise from different sources of applications. This procedure includes select some parts of stream data, preprocessing, incremental learning, and knowledge extraction in one pass. Results from the mining data stream are the knowledge that can support making and accepting intelligent decisions. Processing time and memory space present a trade-off challenge to choose which efficient algorithms are selected from computation theory.

This paper provides an overview of frequent pattern mining from streaming data. It discusses the main research issues in the mining from the data stream. It explains the main data stream window models, their advantages, and drawbacks in comparison. Furthermore, it presents a comparative study of the state-of-the-art approaches and indicates evaluation measures for those algorithms. Also, review the open-source stream mining stream and tools suggested in the literature. Finally, research challenges and major research directions are presented.

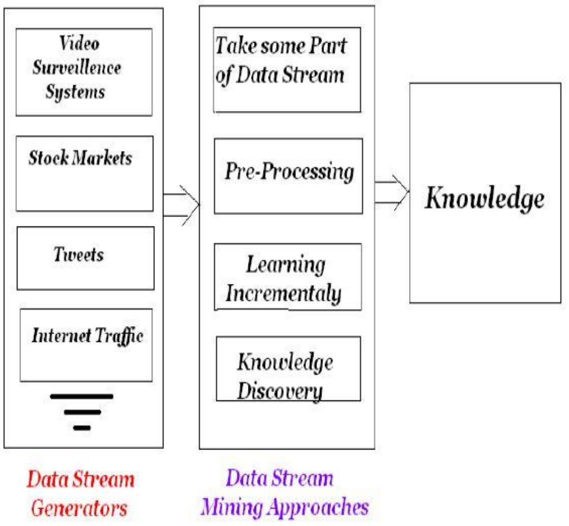

Fig. 2 General procedure for data stream mining [8].

The rest of the paper is organized as follows. Section II presents research issues in mining frequent patterns from streaming data. Section III presents approaches for mining frequent patterns from streaming data. In section IV presents a comparison of mining frequent pattern algorithms and evaluation measures. Section V discusses open-source software and tools. Section VI presents research challenges and future directions. Finally, section VII is the conclusion of the paper.

II. Research issues in Mining Frequent Patterns from Streaming Data

Although, the data stream is an emergent research field. Data stream mining presents challenges in the community of data mining. Several challenges such as bounded memory, velocity, concept drift, and enormous volume of data streams [14]. These issues that arise while mining data streams need to be addressed [15]. As indicated from Fig. 3, many research issues in mining from the data stream.

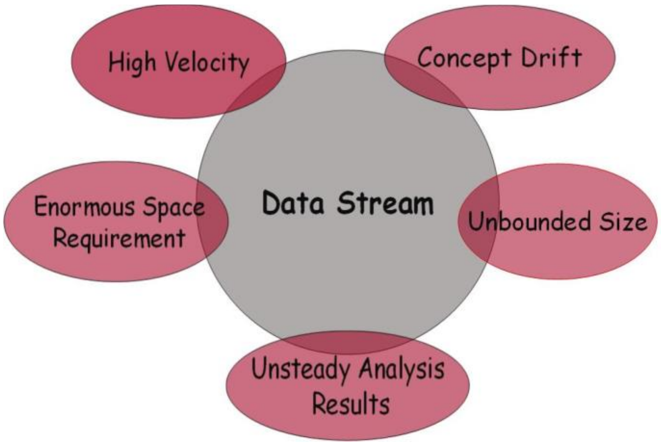

Fig. 3 Research issues in mining freuent patterns from streaming data.

- **High Velocity**: data streams are generated from online and offline sources at high speed. This becomes impractical to scan data several times as in traditional data mining and needs to be enhanced [16].
- **Concept Drift:** is a phenomenon in the data stream used to describe deviations in the distribution of data over time. The process of generating patterns changes over time because the nature of data streams is dynamic [17].
- **Unbounded Size**: data streams are generated from different stream sources with unbounded size. Also, the size of data streams is unknown in advance, unlike stationary data [18].
- **Enormous Space Requirement:** The volume of data generated from streaming applications is enormous. The limited memory may not be enough to store the data stream [19].
- **Unsteady Analysis Results:** The high speed and varying distribution of data may affect the evaluated output. To handle the updating of new data streams during the process, mining from the data stream has to be an incremental process [13].

A summary of the main differences between a traditional database and data stream processing is present in Table 1. The traditional database uses unlimited secondary storage to store the data. A traditional database is used when there is little or no time requirement. In a data stream, the nature of data is volatile, and sequential data access is performed using DSMS. It operates on continuous queries and gives the exact/approximate output for that query. DBMS uses limited main memory to store the data and the data update rate is very high. DSMS is used when there is a real-time requirement [20] [21].

Table 1 traditional database vs data stream Processing.

No.	Parameter	Databases	Data streams
1	Data access	Random	Sequential
2	Available memory	Flexible	Limited memory
3	Processing time	Unlimited	Restricted
4	Data spread	Need not be	Distributed
5	Computation results	Accurate	Approximate findings
6	Data scan	Flexible	Single scan
7	Data schema	Static	Dynamic

III. Approaches of Mining Frequent Patterns from Streaming Data

There are many algorithms for mining frequent patterns from data streams. All those algorithms fall into one of the window models of data stream mining. A window can be either time-based or count-based [22] [23]. Several data stream window models are proposed in the literature. The most common are landmark window model, sliding window model, damped window model, and titled time window model [22][15][24]. A comparison between data stream window models is presented in Table 2.

A. Landmark Window Model

A landmark data model considers the data stream from an initial fixed point in time (usually the time the system starts) to the current time for mining. All the streaming data handle equally. The landmark window retains all batches after the increasing number of batches. Typically, it uses when analysis of historical data. All frequent patterns generated from the whole data stream are considered [22][15], [23][25].

In [19], an algorithm called lossy counting is presented. It computes frequency counts over a stream of single scan transactions. The disadvantage of this algorithm is inefficient in utilizing memory space.

According to [26], an algorithm called the extended TUF-streaming algorithm is proposed. This algorithm finds frequent patterns of uncertain data streams using single scan transactions. Also, it reduces the amount of space required when mining frequent patterns from streaming data. The disadvantage of this algorithm is that it takes longer to visit each node in the UF-stream.

B. Sliding Window Model

This type of window model is concerned with the most recent time points. It has been widely applied because it captures recent changes in patterns, by only taking advantage of recent mining transactions [22].

Two types of sliding window models are fixed size window, variable sliding window. Most sliding window-based algorithms are based on a fixed size window. Some variable sliding window-based algorithms are proposed in [27].

The moment algorithm was first proposed in [28]. It mine closed frequent patterns in streaming data. Also, it uses a count-based with a fixed size of the sliding window. This algorithm consumes more time because it scans the streaming data multiple times.

The MSWTP algorithm was proposed in [29]. This algorithm extracts top-k frequent patterns from streaming data.

DSM-Miner algorithm proposed in [30]. This algorithm work on mine maximal frequent patterns from streaming data. It maintains the following tree called SWM-Tree to mine recent frequent maximal patterns. DSM-Miner is efficient in time performance and memory usage.

In [31], the paper proposes an improved approach called RMFIsM for mining maximal frequent patterns instead of mining all frequent patterns. It also scans stream data with a single scan. The limitation of this approach is using a two-dimensional matrix structure algorithm, which is costly in time efficiency.

C. Damped Window Model

In this model, weights associated with data in the stream depending on its arrival time. This model focuses further on the currently come transactions and gives higher weights to recent data than those in the past [15]. Each transaction of the data stream has a corresponding value, and the value gradually decreases with the increasing time. Therefore, preservation and reduction of information related to historical data need to be considered [23].

In [32] estDec algorithm is proposed as a data mining method for finding recently frequent patterns over an online streaming data. This technique examines every single transaction in the stream without candidate generation. The major limitation of this algorithm is difficult to execute after many transactions, since the size of its prefix tree turns larger than the limited memory space.

In [33] estDec+ algorithm is proposed, It maintains a compressed prefix tree known as CP-tree that is effectively used in finding either frequent or maximal frequent patterns over an online stream. Using CP-tree reduces memory usage compared to the previous estDec algorithm. The main limitation of this algorithm is the increase in average processing time.

In [34], the authors proposed a new mining algorithm called MPM. This algorithm mines significant maximal utility patterns from streaming data.

D. Tilted Time Window Model

This model generates frequent patterns on a set of windows. It assigns a tilted time window for each frequent pattern. The FP-Stream algorithm uses this window model. FP-Stream uses two tree data structures. FP-Tree is firstly designed to store transactions in the current time window. Second, the tree pattern structure stores the frequent patterns of previous windows [15].

TABLE 2 DATA STREAM WINDOW MODELS COMPARISON

	Definition	**Advantages**	**Disadvantages**
Landmark window model	Process all frequent patterns from an initial point of time to the current time for mining	Suitable for analysis of historical data	All instances have the same importance
Sliding window model	Concerned about recently received transactions.	Suitable during recent instances are particular importance	Ignores the history of pattern frequencies
Damped window model	Associate weights with transactions in the data stream	Appropriate when the old transactions may affect results	unlimited time window
Title time window model	Finding frequent patterns over a set of windows	Time-sensitive, and suitable in working with different windows. It assign different weights to each of them	Different windows with different weights may affect time and space efficiency

IV. COMPARATIVE REVIEW AMONG THE STATE OF THE ART ALGORITHMS

Table 3 provides a comparison of frequent pattern algorithms on the data stream. These algorithms are discussed in detail according to the window model used in the previous section. In the table below, FI stands for frequent itemset, CFI stands for closed frequent itemset, MFI stands for maximal frequent itemset.

TABLE 3 COMPARISON REVIEW BETWEEN DATASTREAM ALGORITHMS

Data stream Mining algorithm	Window model	Mining FI/CFI / MFI	Database scan	Tuple/Batch processing
Lossy Counting [19]	Landmark window	FI	Single	Tuple
TUF-streaming [26]	Landmark window	FI	Single	Batch
DSM-FI [35]	Landmark window	FI	Single	Tuple
Moment [28]	Sliding window	CFI	Multiple	Tuple
MSWTP [29]	Sliding window	FI	Single	Tuple
DSM-Miner [30]	Sliding window	MFI	Single	Tuple
RMFIsM [31]	Sliding window	MFI	Single	Tuple
estDec [32]	Damped window	FI	Multiple	Batch
estDec$^+$ [33]	Damped window	MFI	Single	Tuple
MPM [34]	Damped window	MFI	Single	Tuple
FP-Stream [15]	Titled time window	FI	Single	Tuple

A. Evaluation Measures

In this section, we present performance measures for each algorithm in data stream mining. These measures should be examined and taken into consideration in the evaluation of streaming algorithms [14]: -

- **Memory space consumption**: it is essential to examine the average memory consumption of every algorithm.
- **Update time:** the time required for the algorithm toward update its structure and confirm new transactions from the stream. It is recommended the update time be smaller than the arrival time of a new transactions.
- **Decision time:** The time of algorithm takes to decide on new transactions coming from the stream.

V. OPENSOURCE SOFTWARE AND TOOLS

Several streaming data mining frameworks have been intended in the literature. The available open-source software frameworks contain a lot of algorithms. Those algorithms can be extended to suggest innovative approaches or to compare against them. Also, a lot of tools have been proposed for streaming data mining [36][24].

Massive Online Analysis (MOA): is a free, open-source software project for handling evolving massive data streams. MOA is written in java and can be used via the command line, GUI, or Java API [37][38]. It includes a collection of offline and online stream mining algorithms. It also contains tools for evaluation methods and statistics to evaluate the performance of algorithms [39][9].

Scalable Advanced Massive Online Analysis (SAMOA): A framework was written in Java programming for mining big streaming data. It has a set of distributed stream and stream analyses for the most common data mining and machine learning tasks [40]. The goal of SAMOA is to fill this gap, as in Fig. 4.

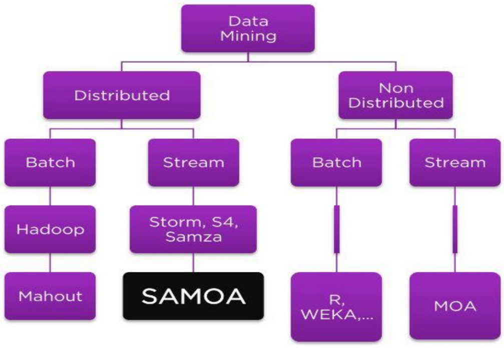

Fig. 4 Data mining tools [40].

Advanced Data Mining in Spark Streaming (StreamDM): It is an open-source framework that uses a spark stream to enable stream processing from a variety of sources. Furthermore, it can handle scalable data streams because of using spark streaming API [41].

A Multi-output Streaming Framework (Scikit-multiflow): is a new open-source framework for learning from streaming data. It is written in python. Like MOA, it also contains stream generators and algorithms for streaming data [42]. The source code is available at https://github.com/scikit-multiflow/scikit-multiflow.

VI. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

This section summarize the main challenges and future research directions for mining frequent patterns from streaming data [13]. In Fig. 5 presents research challenges in streaming data. These challenges are summarized as stream preprocessing, timing and availability of information, analyzing event data, making models simpler, dealing with legacy systems, protecting privacy and confidentiality, and evaluation of stream mining algorithms [43].

Open research directions in mining frequent patterns from streaming data are the following: -

- Requirements to develop data preprocessing methods for more complex data stream types.
- Concept drift detection and adaptation of model in the online data stream. Dealing with concept phenomena can affect multidisciplinary areas.
- Privacy preservation in mining streaming data. The purpose is to develop data stream mining approaches that do not reveal information or frequent patterns. These algorithms should promise confidentiality and privacy.
- Evaluation techniques and measures of stream mining algorithms. There is a need for evaluation

strategies and methods that are quite well-defined.

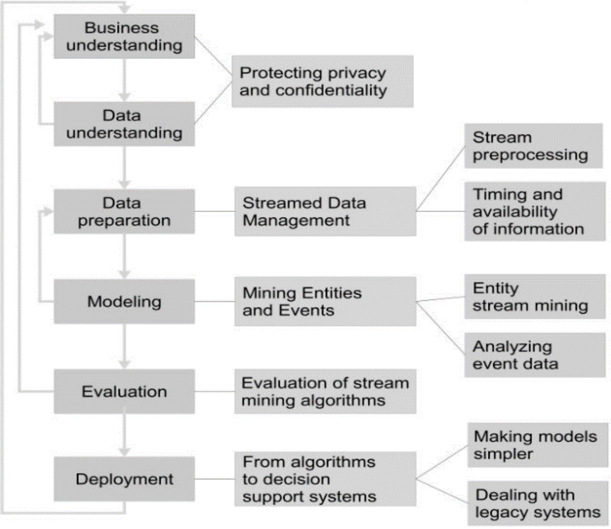
Business understanding
Data understanding
Protecting privacy and confidentiality
Data preparation
Streamed Data Management
Stream preprocessing
Timing and availability of information
Modeling
Mining Entities and Events
Entity stream mining
Analyzing event data
Evaluation
Evaluation of stream mining algorithms
Deployment
From algorithms to decision support systems
Making models simpler
Dealing with legacy systems

Fig. 5 Research challenges in streaming data [43].

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments. We also thank the authors of all references for helping us set up the paper. Special thanks to Dr.Engy el-shafeiy (University of Sadat City) for support and motivation during this research.

VII. CONCOLUSIONS

Data stream mining is an effective area of research and is rapidly growing. In this paper, we introduce recent research efforts in mining frequent patterns from streaming data. There are still various challenges in data stream mining arising from real-world applications. These challenges include not only the study of memory space consumption, but also the time complexity of data stream mining algorithms. This paper presents a reference to researchers in introducing the topic of mining frequent patterns from streaming data. It discusses window models, and recent data stream mining algorithms. Finally, we discuss research challenges for data streams mining from real-world applications. We conclude that mining frequent patterns from streaming data is still a significant direction.

REFERENCES

[1] S. J. Fong and R. C. Millham, *Bio-inspired Algorithms for Data Streaming and Visualization, Big Data Management, and Fog Computing*. 2021.

[2] S. Bagui and P. Stanley, “Mining frequent itemsets from streaming transaction data using genetic algorithms,” *Journal of Big Data*, vol. 7, no. 1, 2020.

[3] D. Leite, I. Škrjanc, and F. Gomide, “An overview on evolving systems and learning from stream data,” *Evolving Systems*, vol. 11, no. 2, pp. 181–198, 2020.

[4] L. Bustio-Martínez, A. Muñoz-Briseño, R. Cumplido, R. Hernández-León, and C. Feregrino-Uribe, “A novel multi-core algorithm for frequent itemsets mining in data streams,” *Pattern Recognition Letters*, vol. 125, pp. 241–248, 2019.

[5] H. F. Li, C. C. Ho, M. K. Shan, and S. Y. Lee, “Efficient maintenance and mining of frequent itemsets over online data streams with a sliding window,” *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2672–2677, 2006.

[6] S. Nasreen, M. A. Azam, K. Shehzad, U. Naeem, and M. A. Ghazanfar, “Frequent pattern mining algorithms for finding associated frequent patterns for data streams: A survey,” *Procedia Computer Science*, vol. 37, no. November, pp. 109–116, 2014.

[7] U. Kokate, A. Deshpande, P. Mahalle, and P. Patil, “Data stream clustering techniques, applications, and models: Comparative analysis and discussion,” *Big Data and Cognitive Computing*, vol. 2, no. 4, pp. 1–30, 2018.

[8] V. S. Reddy, T. V Rao, and A. Govardhan, “Data mining techniques for data streams mining,” *Review of Computer Engineering Studies*, vol. 4, no. 1, pp. 31–35, 2017.

[9] A. Bifet *et al.*, “Extremely fast decision tree mining for evolving data streams,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F1296, pp. 1733–1742, 2017.

[10] M. M. W. Yan, “Accurate detecting concept drift in evolving data streams,” *ICT Express*, vol. 6, no. 4, pp. 332–338, 2020.

[11] H. Mehmood, P. Kostakos, M. Cortes, T. Anagnostopoulos, S. Pirttikangas, and E. Gilman, “Concept Drift Adaptation Techniques in Distributed Environment for Real-World Data Streams,” *Smart Cities*, vol. 4, no. 1, pp. 349–371, 2021.

[12] F. G. Da Costa, R. A. Rios, and R. F. De Mello, “Using dynamical systems tools to detect concept drift in data streams,” *Expert Systems with Applications*, vol. 60, pp. 39–50, 2016.

[13] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, “A survey on data preprocessing for data stream mining: Current status and future directions,” *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[14] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” *Information Fusion*, vol. 37, pp. 132–156, 2017.

[15] A. Borah and B. Nath, “Mining patterns from data streams: An overview,” *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, pp. 371–376, 2017.

[16] B. Denham, R. Pears, and M. A. Naeem, “HDSM: A distributed data mining approach to classifying vertically distributed data streams,” *Knowledge-Based Systems*, vol. 189, p. 105114, 2020.

[17] A. Haque, L. Khan, M. Baron, B. Thuraisingham, and C. Aggarwal, “Efficient handling of concept drift and concept evolution over Stream Data,” *2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016*, pp. 481–492, 2016.

[18] S. Dawar, V. Sharma, and V. Goyal, “Mining top-k high-utility itemsets from a data stream under sliding window model,” *Applied Intelligence*, vol. 47, no. 4, pp. 1240–1255, 2017.

[19] G. S. Manku and R. Motwani, “Approximate frequency counts over data streams,” *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 1699–1699, 2012.

[20] M. S. B. PhridviRaj and C. V. GuruRao, “Data Mining – Past, Present and Future – A Typical Survey on Data Streams,” *Procedia Technology*, vol. 12, pp. 255–263, 2014.

[21] Simarpreet and V. Singla, “A Review: Frequent Pattern Mining Techniques in Static and Stream Data Environment,” *Indian*

Journal of Science and Technology, vol. 9, no. 45, 2016.

[22] H. Mohamed, A. Sharaf, and M. Abd, "Mining Frequent Itemsets from Online Data Streams: Comparative Study," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 7, pp. 117–125, 2013.

[23] M. J. Sutha and F. R. Dhanaseelan, "Mining frequent, maximal and closed frequent itemsets over data stream - A review," *International Journal of Data Analysis Techniques and Strategies*, vol. 9, no. 1, pp. 46–62, 2017.

[24] M. Bahri, A. Bifet, J. Gama, H. M. Gomes, and S. Maniu, "Data stream analysis: Foundations, major tasks and tools," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 3, pp. 1–17, 2021, doi: 10.1002/widm.1405.

[25] P. Goyal, J. S. Challa, S. Shrivastava, and N. Goyal, "Anytime Frequent Itemset Mining of Transactional Data Streams," *Big Data Research*, vol. 21, p. 100146, 2020.

[26] C. K. S. Leung, A. Cuzzocrea, and F. Jiang, "Discovering frequent patterns from uncertain data streams with time-fading and landmark models," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7790 LNCS, pp. 174–196, 2013.

[27] H. Li and L. Wang, "A variable size sliding window based frequent itemsets mining algorithm in data stream," in *AIP Conference Proceedings*, May 2017, vol. 1839.

[28] Y. Chi, H. Wang, P. S. Yu, and R. R. Muntz, "Moment: Maintaining closed frequent itemsets over a stream sliding window," *Proceedings - Fourth IEEE International Conference on Data Mining, ICDM 2004*, pp. 59–66, 2004.

[29] H. Chen, "Mining top-k frequent patterns over data streams sliding window," *Journal of Intelligent Information Systems*, vol. 42, no. 1, pp. 111–131, 2014.

[30] J. Yang, Y. Wei, and F. Zhou, "An efficient algorithm for mining maximal frequent patterns over data streams," *Proceedings - 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2015*, vol. 2, pp. 444–447, 2015.

[31] S. Cai, S. Hao, R. Sun, and G. Wu, "Mining recent maximal frequent itemsets over data streams with sliding window," *International Arab Journal of Information Technology*, vol. 16, no. 6, pp. 961–969, 2019.

[32] J. H. Chang and W. S. Lee, "Finding recent frequent itemsets adaptively over online data streams," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 487–492, 2003.

[33] S. J. Shin, D. S. Lee, and W. S. Lee, "CP-tree: An adaptive synopsis structure for compressing frequent itemsets over online data streams," *Information Sciences*, vol. 278, pp. 559–576, 2014.

[34] U. Yun, D. Kim, E. Yoon, and H. Fujita, "Damped window based high average utility pattern mining over data streams," *Knowledge-Based Systems*, vol. 144, pp. 188–205, 2018.

[35] H. F. Li, M. K. Shan, and S. Y. Lee, "DSM-FI: An efficient algorithm for mining frequent itemsets in data streams," *Knowledge and Information Systems*, vol. 17, no. 1, pp. 79–97, 2008, doi: 10.1007/s10115-007-0112-4.

[36] D. Brzezinski, "Mining Data Streams with Concept Drift," 2010, [Online]. Available: http://www.cs.put.poznan.pl/dbrzezinski/publications/ConceptDrift.pdf.

[37] B. G. Alhogbi, *MACHINE LEARNING FOR DATA STREAMS with Practical Examples in MOA*, vol. 53, no. 9. 2017.

[38] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.

[39] P. Kranen *et al.*, "Stream data mining using the MOA framework," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7239 LNCS, no. PART 2, pp. 309–313, 2012.

[40] G. De Francisci Morales and A. Bifet, "SAMOA: Scalable advanced massive online analysis," *Journal of Machine Learning Research*, vol. 16, pp. 149–153, 2015.

[41] A. Bifet, S. Maniu, J. Qian, G. Tian, C. He, and W. Fan, "StreamDM: Advanced Data Mining in Spark Streaming," *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, pp. 1608–1611, 2015.

[42] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A Multi-output Streaming Framework," *Journal of Machine Learning Research*, vol. 19, pp. 1–5, 2018.

[43] G. Krempl *et al.*, "Open challenges for data stream mining research," *ACM SIGKDD Explorations Newsletter*, vol. 16, no. 1, pp. 1–10, 2014.