International Journal of Engineering & Technology, 7 (4) (2018) 2166-2171

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.140819
Research paper

Critical evaluation of classifiers in data stream mining

Lalit Agrawal [1] *, Dattatraya Adane [1]

[1] Shri Ramdeobaba College of Engineering and Management, Nagpur, India
**Corresponding author E-mail:agrawalls@rknec.edu*

Abstract

Over past decade there has been a significant increase in the volume of online data. Extracting meaningful knowledge from this high volume data is considered as important aspect of research. It is very difficult to completely store full data, because of its perpetual nature. Therefore, analysis is needed while the "data is moving". This moving data is known as data stream and analyzing it without storing it completely is termed as data stream mining. In recent years, many new techniques have been proposed to overcome the challenges of data stream mining. In this paper, we review the operation of popular streaming algorithms highlighting their strength and weaknesses. We also evaluate the classifiers used in these algorithms against two popular benchmark datasets namely (a) forest cover (forest) and (b) german credit available at UCI repository. Finally, we present our critical observation and draw conclusions on the basis of our analysis.

Keywords: *Classification; Clustering; Data Stream; Random Forest; Stream Mining.*

1. Introduction

Traditional data mining systems are suitable for basic and organized information collection. Data should be arranged in predefined pattern and stored completely is one of the important prerequisite of data mining system. There is a fast and continuous improvement in making new database frameworks and data collection technologies to handle the huge data getting generated by internet users, sensor applications etc. Data of this kind has different and complex structures making it difficult to store in predefined manner in relational database for analysis. Mining of such complex information while the data is moving turns into a critical constraint for standard data mining techniques. To overcome this problem, researchers have put forward new data stream mining methods to handle the difficulties of storage and analysis of continuously generated data [1-2].

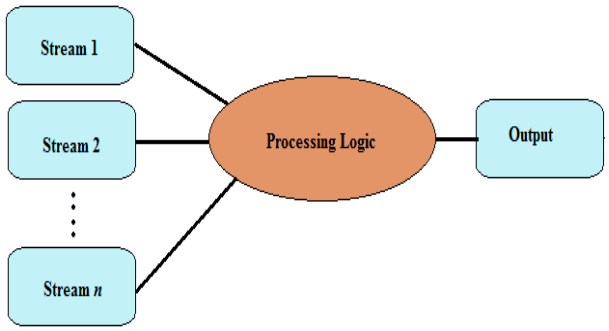

Fig. 1: Stream Processing Overview.

Data stream is viewed as a continuous flow of data with rich information generated from variety of heterogeneous sources sent for storing or processing purposes [3]. Consider a system like social network where number of users creating data into the network. The information is in terabytes, transient, changing quickly and unbounded. These features introduce many challenging issues in data stream mining. In the past, traditional online analytical processing (OLAP) and data mining techniques commonly require multiple passes over the data and are subsequently infeasible for stream applications [4].

Data stream is continuous and endless in nature. This makes storage of the stream data in a centralized database is very difficult. One of the strategies to tackle the problem is "divide and conquer" strategy. One of the best methods for such big data problem is the map reduce. It is a software framework developed in java for data management and processing. The core is made up of distributed file system (DFS) and map reduces [5]. The DFS has extensive datasets which are repetitively stored over multiple machines. It guarantees adaptation to internal failure and information accessibility for parallel processing of huge datasets. For this reason, DFS isolates a substantial document into little pieces called as information hubs which work in parallel.

The Huge data being exceptionally enormous poses a lot of challenges in processing. Although parallelism, adaptation capabilities are plausible, statistical techniques are yet to be investigated on such high dimensional substantial large stream of data. Because of the incomprehensible collection of online information, a number of articulated techniques are required which takes into account the data stream [6]. Since data streams are portrayed by a persistently high rate of generating and approaching data, the stream comes to the processing frameworks consistently with fluctuating entry rates, which is not at all like traditional data warehouses. Along these lines, there are few issues [7] [8] in handling and mining of the data stream as below:

a) The data received in the stream is continuous
b) Proper order of data attributes cannot be ascertained in data stream
c) The stream of data is infinite in size
d) The storage of the stream data is also one of the constraints.

Therefore, data has to be discarded or archived in some format for future reference. In this paper, we present the study of various stream mining algorithms and their analysis. We also present the experimental study of various static and stream mining algorithm and the future scope in the field of data stream mining.

2. Related techniques

During last decade, research community has shifted focus from standard data mining to stream data mining to address the issues in data stream classification, clustering etc. we classify the techniques into five broad categories and their detailed analysis is presented in this paper. We have also carried out experiments using WEKA [39] and MOA [41] tools to see the working of various algorithms on standard data and stream data. Their comparative analysis is also presented in this paper along with future research direction.

2.1. Decision tree classifier

Many researchers have proposed new techniques based on the decision trees for giving better classification accuracy. The decision tree has initially provided the solution for handling various challenges of the classification in static data that can be completely stored. But because of unique characteristics of stream data, traditional classification algorithm based on simple decision tree has degraded the performance.

Xue-Gang Hu et., al. [9] proposed semi-random multiple decision tree for data stream (SRMTDS) incremental algorithm based on random decision trees. This approach is benefitted by the combination of hoeffding bounds, heuristic methods to calculate information gain and naive bayes algorithm. This algorithm has better efficiency in comparison with VFDT [30]. It is suitable for the applications working in distributed environment. It doesn't classify the noise and concept drift viably.

Incremental technique with multiple semi-random decision trees (MSRT) [37] makes use of two different windows for training and testing. To differentiate between concept drift in data stream and noise, it uses inequality of hoeffding bounds. This technique has advantages over CVFDT[38] in terms of better classification accuracy and noise reduction. Data stream with skewed distribution poses challeges for this algorithm.

Classes of an instance changes over time resulting into a situation called as concept drift. A decision tree based, Sensitive concept drift probing decision tree algorithm (SCRIPT) [10] based on X^2 statistical test is proposed to handle concept drift in data stream. The system helps in reducing the unnecessary system cost to get a stable data stream. This method is strong enough to rebuild the classifier through unstable data streams. The system is feasible for applications in which accurate detection is needed. This technique is based upon the assumptions that all the drifting attributes should be collected in some specific portion. If this is not ascertained then these attributes are treated as noise.

Sattar Hashemi et. al. [11] proposed a decision tree based algorithm named as adapted one-versus-all decision tree for data stream classification. In this system, the *k* individual binary classifiers are utilized to classify a new instance. The classifier with the best confidence value is chosen when the *k* numbers of the classifiers are executed. The adapted one-versus-all method is chosen because of its low error correlation. This results into high classification accuracy. Adapted version has several advantages over basic version in terms of handling of concept change, not all instances are fed to the component classifier reducing overhead. Future scope of this method includes handling of imbalanced class distribution.

Zahra et. al., proposed the incremental decision tree based algorithm evolving fuzzy min max decision tree (EFMMDT) [12] in which every internal node has dynamic splitting logic. This logic is self sufficient to train itself based upon the data arrival. In normal decision tree, one attribute is selected as a split criteria in internal node whereas in this method the every internal node has a trainable function based upon multiple attributes giving better efficiency especially in the case of concept drift. Future work of this method is getting alternative split test criteria for better classification prediction.

Peng Zhang et.al., [13] proposed Ensemble-Tree indexing pattern to store the ensembles for better prediction. The researchers optimized the ensembles as spatial database formerly applied an indexing technique which implies a height balanced structure named R-tree which reduces the delay from liner to sub-liner complexity. The technique can be automatically and continuously updated by integrating new classifiers and also it discards the old classifiers that are not trending currently. This ability helps in adapting to new trends with pattern data streams. This method can further be extended to handle spatial data.

2.2. Concept drift technique

Concepts underlying data keeps on changing over the period of time. Data stream processing system should be capable of recognizing the change in data pattern, adapt suitable analyzing and handling mechanism. This is difficult to perform because data is continuously moving with high speed and analysis has to be done on-the fly in single pass without storing it completely. This change in data may be sudden or gradual. Many researchers have focused on this aspect of data stream processing. Few popular techniques are summarized below:

Gama et.al, [14] proposed a Drift Detection Method (DDM) to detect the decision classes with the help of every iteration of the online classifier which can be either true or false. Bernoulli`s trials are used to calculate errors. They maintain two variables namely pmin and smin. History of error rate is stored in pmin and smin is used for every data stream they have maintained. pmin and smin are used to detect the warning level and alarm level conditions. The examples are remembered in the separate window whenever the warning level is reached. A new classifier is adapted from the example stored in separate warning window and the previously learned classifier is dropped.

Tatsuya Minegishi et.al, [15] Proposed feature evolution and feature selection method with an online decision tree. Their online decision tree consists of very fast decision tree learner algorithm because of which the performance is increased and the classification is more accurate compared to other methods. This approach focuses more on classification accuracy and in turn, selects fewer features which show the way to simplify learning tasks for a huge amount of data.

Mohammad M Masud et.al, [16] designed an algorithm which aimed at enabling automatic recognition of novel classes even before when the true labels of novel class instances arrive in the presence of concept drift. Also, one of the most important aspects of stream data was covered by this algorithm i.e. arrival of a novel class but they could not address classification problem under dynamic feature sets.

2.3. Novel class detection

One of the important concepts in data stream mining is novel class detection. Fixed number of classes cannot be detected in a data stream classification because in the real environment new classes can arrive at any time and old classes may vanish in due time. During the evolution of new classes, the data classification technique should detect the arrival of novel class.

Mohammad M Masud ET. al. [17] proposed a stream data classification in which each classifier consists of a detector to detect novel class. It also addresses the concept evolution and concept drift. It also consists of feature set homogenization technique for feature evolution. More than one novel class is detected by the enhanced novel class detection module which makes it more robust and improves the performance in terms of time, space and accuracy.

Amit Biswas ET. al. [18] proposed a decision tree based approach to detect multiple novel classes. First, decision tree is constructed and then the calculation of the percentage of data points on each leaf node is done. Then, based on similarity, clustering is applied on each leaf node. The calculated percentage is referred as the arrival of novel class if the number of data points in a leaf node of tree increases. Multiple novel classes are detected in order to make a graph where the total number of connected components determines the number of novel class arrived and it becomes very easy and efficient approach for detection of novel class.

Huan Liu et al. [19] explained a feature selection process. This feature selection is very effective for reduction of dimensions and it is a very important to get a successful data mining application. Challenges which occurred due to high dimensional data are answered by this process. The benefits of dimensionality reduction help in creating simpler and more comprehensible models. It improves data performance and also helps to clean, prepare and understand data.
G. Divya et al. [20] created a hybrid approach to classify and detect the novel class in the feature evolving data streams. The unwanted data is present in the data stream is removed by the outlier detection method. Also, for novel class detection, this methodology uses Naive Bayes classifier and Nearest Neighbor algorithm. When a new feature appears the long-standing feature fades out and new one occupies the space. In order to get the accurate data the outlier detection techniques are used to remove the unwanted data from the data streams.

2.4. Clustering

If we have one variable that we have to process as a component of a few known factors then these issues are called managed learning issues. However, numerous a times, we may be made a request to investigate the examples inside given information with no objective quality. Such issues are called unsupervised learning issues. Clustering is an example of unsupervised learning. Popular streaming algorithms are explored here.
Stream algorithm focuses on batches of points that fit in the main memory to process the data stream [21]. The concept of the local search algorithm is used which runs in linear time in proportion to the number of points. It has higher time but it provides compact information. This algorithm works in two phases i.e. offline and online phases which are followed by divide and conquer approach. The data stream is in the form of buckets and then it finds k cluster for each of the buckets by performing the k-median algorithm. During this point, the cluster centers are weighed and stored depending on the total data points to resemble a particular cluster and then the data points are discarded. Later, the weighed centers are clustered in few smaller clusters. Main advantage is its lower time consumption and lower space complexity. The biggest disadvantage is stream data adoption to concept evolution.
Aggrawal proposed an algorithm called as Clustream [22]. It is used for clustering evolving data stream which is based on k-mean technique. This is designed by combining ideas of both BIRCH [23] and STREAM [21]. In this algorithm, the clustering process is divided into offline and online components and these components use the micro and macro clustering. In online component, the data summary is stored in the form of micro-cluster by using CF-vector. There is a concept of clustering feature of BIRCH which is a temporal extension is a micro cluster. Clustream stores summarized data in form of snapshots manner this helps the user to identify the time interval required for clustering of micro-clusters. The next phase consists of offline components that carry out k-mean to cluster micro clusters into bigger size clusters. For this purpose pyramid architecture is used. The most important advantage of this algorithm is its acceptable higher efficiency and accuracy.
Aggrawal et al. proposed an algorithm called as HPStream [24] used to cluster a high dimensional data stream. A fading cluster structure stores the summarization of the fading data. The recent most data gets more privilege and hence the past data is discarded. The original data stream is projected with high dimension for the selection of a subset of dimensions. Every cluster has distinct dimensions and number of dimensions. The number of dimensions is highly scalable and it is incrementally updatable. The clusters of arbitrary shapes cannot be found and in-depth knowledge is required for giving a number of clusters and parameters of projected dimensions.
DenStream [25] is a density-based algorithm proposed by Cao F. et. al. It is an extension of DBSCAN. This algorithm is divided into online and offline phases. Online phase maintains the micro-clusters and these maintained micro-clusters are generated to final clusters in the offline phase. It can handle outliers and also provide arbitrary shapes to the clusters.
Density grid cluster forms the basis of D-Stream [26]. A complete data space is divided into a grid like structure. This algorithm has online and offline phase. In online phase, the mapping of receiving data points to the corresponding grid is carried out whereas in the offline phase the density of each grid is calculated and the leftover data points are removed. A fading function is used to decrease the density of the grid with respect to time. If the fading function goes below the threshold value then the grid is discarded. With the increase in the number of dimensions, the number of grid increases exponentially which makes it not scalable on a number of data dimensions.
For proper consumption of resources, the data streams should be processed in the batches of some predefined m size [27]. As the main memory receives the data bucket, this technique groups the data into k clusters depending on the respective centers which are weighed by the number of data points present in each cluster. It summarizes the bucket information i.e., the weighted center information of each of k clusters and the data points or elements used are discarded. For each m-data points, this process is repeated. This technique makes use of single pass approach. But the major disadvantage of this technique is the possibility of the older data stream to become dominating and there is no concept of time granularity [28].

2.5. Classification

We already know about techniques that group the data points or items based on their differences. This is a supervised technique with a set of training examples which are in the form of (i, j). wherein i represents the vector of n number of attributes and j is the discrete class label which aims at making a model of the form of j=f(a), where f(a) should be able to accurately predict the class j for all the future examples.
One of the most effective ways of classifying the data points is the decision tree learning algorithm. It consists of test nodes, the root node and leaf nodes. Here classification [29] of the streaming data is the major requirement where the examples are read-in only with one pass and processed in even less amount of time. When the data is streaming in, it integrates the data into the tree. Even if the model is built incrementally, it can be used to classify data. Along with these advantages of Hoeffding learning tree, there are several disadvantages as well. Identical or near to identical splitting qualities of the attributes can have ties among them. It takes time in order to assume the superiority of the attribute. The time varying data does not have a solution which means concept drift handling is impossible here because after creating a root node, changing it will require complete tree restructuring.
One of the extensions of Hoeffding learning decision tree is very fast decision tree (VFDT) [30]. There was a need to shift towards VFDT to resolve the ties between the identical and near to identical splitting attributes which possess closer split evaluation values. This is termed as a better technique because of the time consumption and memory. However, it still has the issue of concept drift.
Static training data is used to build the classification model which stores the current state only, so if there are any changes in the current data, it learns a completely new model which consumes a lot of resources when we consider time-varying streaming data. The concept drift is the process where the time varying data enforces the changes in the target classification model over time. Another extension of VFDT is the concept adaptive-very fast decision tree(C-VFDT) [31]. It is used to address the concept drift problem in the time-varying data streams. It is similar to VFDT in terms of speed and time but the difference is in the fact that it detects and responds to changes that happen in the data without constructing a new target classification model every time. It handles this by including alternate subtrees. The problem occurs when the attribute near to the root does not pass through the Hoeffding bound which results in large portions of the subtree.

This is an improvement of VFDT which is based on the idea of using the data about different chunks of data streams to train the group of classifiers. For examples, different chunks or windows of data are used to train n number of independently grown trees. Further, n different class predictions are produced by each of the trees. The prediction of the whole ensemble is the highest voted class. Further, on the basis of the accuracy in the time-varying environment, each individual tree`s prediction is assigned a weight. The decision is based on the highest weighted votes of the selected top k-classifiers [32]. The aim of combining all the classifiers was to attain higher accuracy. The least accurate classifiers are discarded here.

The Random Forests algorithm [33] is a classification technique developed by Breiman. Superficially, random forests are similar to binary decision tree sets. Suppose a data set contains n records, each with m attributes. We develop a set of decision trees, each of a subset of the n records, chosen from the random dataset with replacement. Therefore, the training data set for each tree contains several copies of the original records. The random selection with replacement ensures that approximately n / 3 of the records are not included in the training set and are therefore available as a set of tests to evaluate the performance of each tree.

The construction of a single tree uses a variant of the standard decision tree algorithm. In decision tree, the set of attributes considered in a node is the complete set of attributes that have not yet been used in the main parent nodes. On the contrary, in the random forest algorithm, the set of attributes considered in each internal node is a randomly selected subset of attributes, size m.

3. Theoretical analysis

In this section of paper, we have compared techniques briefly explained in earlier sections. As observed from the above many existing Streaming algorithms are designed to handle two-class classification problem. Few multi-class classification algorithms also exist but with reduced classification accuracy. Parameters (Number of trees, the size of tree etc.) should be changed dynamically at run time. Most of the Streaming algorithms handle numerical data only. However, categorical data is most common in real time. Stream Classification on Skewed distribution needs to be handled. Stream Classification Algorithm should deal with missing data. Should be able to handle concept drift (generation of new classes and extinction [if any] of old classes).

4. Experimental analysis

Based on the notable performance reported in the literature, four different classifiers are chosen to perform the comparative study in case of static data and stream data. These classifiers are decision stump, Random tree, Naive Bayes and Random forest.

A decision stump is a machine learning model consisting of a one-level decision tree. That is, it is a decision tree with one root node immediately connected to the leaf nodes. A decision stump forecast is based on the value of just a single input feature. Sometimes they are also called as 1-rule classifier. Random Hoeffding tree has a unique feature that it takes very few attributes to choose optimal node splitting point. It uses hoeffding bound for the same. Naive Bayes is probabilistic approach for machine learning. It is based on Bayes theorem. Random forest is a collection of multiple decision trees and final decision is taken by considering the score generated from all the individual trees.

Waikato Environment for Knowledge Analysis (Weka) [38] which is a popular suite of machine learning is used for carrying out experiments on static data. It supports variety of tools and algorithms for predictive data analysis. We have considered the benchmark datasets namely German credit and forest cover (forest) available on UCI repository to study classification accuracy. Similar experiments were carried on streaming data to study their behavior in streaming context. For experiment, we have chosen massive online analysis (MOA) framework [40] which is a popular tool to analyze streaming data.

German credit data available at UCI repository [39] is widely used by researchers for testing their approaches on stream data. This multivariate dataset classifies people as good or bad on attribute credit risk. Number of instances are 1000 and attribute count is 20. Attributes are of type categorical and integer both.

Forest Covertype dataset available at UCI repository is used to predict the forest cover information based on 54 attributes. Number of instances recorded in this dataset are 581012.

For experimentation purpose, dataset is randomly divided into training and testing sets. Based upon popular split criteria, 66% of records are randomly used for training and remaining for testing. Cross validation parameters are set to 5 folds. For static data, the dataset is divided into 5 equal subsamples where one is used as validation data and remaining are used to train the model. During 5 folds, every subsample is once used as a validation data and then their results are averaged to get the final result. As we do not have a choice of revisiting the dataset in case of streaming data, analysis is single folded.

Table 1: Performance on Static Data

Classifier	Classification Accuracy using German Credit data	Classification Accuracy using Forest Cover data	Time to build model using German Credit data	Time to build model using Forest Cover data
Decision Stump	70.00%	48.76%	0.02 sec	11.51 sec
Random Hoeffding Tree	65.70%	62.30%	0.02 sec	18.00 sec
Naive Bayes	75.10%	70.40%	0.03 sec	09.23 sec
Random Forest	75.30%	76.90%	0.52 sec	30.00 sec

Table 2: Performance on Streaming Data

Classifier	Classification Accuracy using German Credit data	Classification Accuracy using Forest Cover data	Time to build model using German Credit data	Time to build model using Forest Cover data
Decision Stump	70.00%	36.46%	0.08 sec	8.00 sec
Random Hoeffding Tree	70.00%	36.46%	0.03 sec	6.74 sec
Naive Bayes	70.00%	36.46%	0.02 sec	7.80 sec
Random Forest	70.00%	36.46%	0.02 sec	7.24 sec

5. Observation

Experiments were carried out to study the behavior of decision stump, random hoeffding tree, naive bayes and random forest classifiers under consideration on static and streaming data. Performance is measured using two parameters namely classification accuracy and total time required to build the model. While applying the classifiers over german credit data which contains 1000 instances and 20 attributes random forest outperforms other classifiers but at the cost of increased time. Similar results can be observed by using forest cover data which has considerably high number of

instances i.e., 581012 and instances count is 54. We can observe that accuracy percentage obtained by forest cover data has also decreased nearly to half as compare to the classification accuracy obtained by considering german credit data. The performances of classifiers are greatly affected by number of instances and number of attributes in the data.

Going further in streaming scenario, keeping the accuracy percentage fixed for all the classifiers the time taken to build the model is observed as shown in Table 2. Experiments were carried out 30 times and most sought value of classification accuracy is considered for further analysis. Here, we can observe time to build the model also depends upon the nature of data under consideration. Section 2 describes various techniques related to stream mining on the basis of five major factors namely decision tree classifier, concept drift technique, novel class detection, clustering and classification. Out of the above mentioned algorithms we have chosen ten algorithms representing each group based on their popularity and availability in literature. As per the literature we have identified benefits, limitations and methods used in these popular algorithms. The same has been summarized in Table 3.

Table 3: The Performance of Various Algorithms Based Upon Theoretical Study

Algorithm	Benefits	Limitations	Techniques used
SRMTDS [9]	Improved performance in time, space and accuracy as compared to very fast decision tree, suitable for application working in distributed environment	Need to focus on noise and concept drift classification	Random decision tree, hoeffding bound, naïve bayes
SCRIPT [10]	Suitable for both stable and instable data streams, concept drift detection	High system cost to rebuild the decision tree classifier	Decision tree, statistical X^2 test
Adapted One versus All Decision tree [11]	Faster training, faster updating and high classification accuracy	Need to focus more on concept change efficiency	Decision tree
E-Tree [13]	Shorter prediction time and sub linear complexity of algorithm	Need to extend it for spatial / temporal data stream	E-tree
Recurring class detection [16]	Distinguishes between novel class and recurring class	Time requirement is high	k-mean
Classification and Novel Class Detection [20]	Data stream classification and identifies the novel class	Recent classifiers should be included	Combination of outlier detection method, Naïve Bayes, nearest neighbor algorithm
CluStream [22]	Maintenance of Online and Offline phase	This model is available upon request	Uses micro clusters, pyramidal time frame
Option tree based mining [27]	Advantages over Hoeffding tree based method	Concept drift should be focused	Option trees
Classifier Ensemble [32]	Division of the process into online and off line components	Maintaining offline component and updating it more frequently would be costly	Pyramidal timeframe and micro clustering approach
Random Forest [33]	Collection of trees used for classification	Creation of tree, training and validation the results are time consuming	Decision trees

6. Conclusion

Data stream mining is getting huge importance nowadays as many applications require data to be analyzed on the fly. Because of transient nature it is not possible to store data completely. Also for making 'on-the-fly' decisions, algorithm should run fast without compromising on accuracy. There is a need of dynamic algorithm that can adapt to fast and evolving streams, identifies the concept drift without degrading the performance. We have carried out extensive experiments on various classifiers with static and streaming data. During our experiments we have considered two datasets which belongs to different domains, number of attribute and data size is also different. We have observed: 1) the accuracy percentage has declined substantially when the size of data is increased. This would be the big problem because in streaming, data is infinite in length. 2) Time to build model also depends upon the nature of data under consideration. Algorithms developed using single classifier does not achieve the desired accuracy in real time as expected from stream mining algorithms. Few hybrid approaches also exist in literature but their performances needs to be further improved to meet the real time need of streaming.

Based on our observations we propose to explore streaming algorithms to develop hybrid methodology to process data stream faster and better irrespective of nature of data stream.

References

[1] Babcock, Brian, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. "Models and issues in data stream systems." In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 1-16. ACM, 2002. https://doi.org/10.1145/543613.543615.

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

[3] Chaudhry N., Show K., and Abdelgurefi M., "Stream data Management", Advances in a Database system. Vol. 30: Springer, 2005.

[4] Han J. and Kamber M. Data Mining: Concepts and Techniques, Second ed. The Morgan Kaufmann Series in Data Management Systems: Elsevier, 2006.

[5] Ferreira Cordeiro, Robson Leonardo, et al. "Clustering very large multi-dimensional datasets with mapreduce." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[6] Lior Cohen, Gil Avrahami, Mark Last, Abraham Kandel, "Info-fuzzy algorithms for mining dynamic data streams", Applied Soft Computing, vol.8.4, pp 1283–1294, 2008. https://doi.org/10.1016/j.asoc.2007.11.003.

[7] Jurgen Beringer, Eyke Hullermeier, "Online clustering of parallel data streams", Data & Knowledge Engineering, vol. 58, pp 180–204, 2006. https://doi.org/10.1016/j.datak.2005.05.009.

[8] Charu C. Aggarwal, Philip S. Yu, "On Clustering massive text and categorical data streams", Knowledge Information System, vol. 24, pp 171–196, 2010. https://doi.org/10.1007/s10115-009-0241-z.

[9] Hu, Xue-Gang, Pei-Pei Li, Xin-Dong Wu, and Gong-Qing Wu. "A semi-random multiple decision-tree algorithm for mining data streams." Journal of Computer Science and Technology 22, no. 5 (2007): 711-724. https://doi.org/10.1007/s11390-007-9084-9.

[10] Tsai, Cheng-Jung, Chien-I. Lee, and Wei-Pang Yang. "An efficient and sensitive decision tree approach to mining concept-drifting data streams." Informatica 19, no. 1 (2008): 135-156.

[11] Hashemi, Sattar, Ying Yang, Zahra Mirzamomen, and Mohammadreza Kangavari. "Adapted one-versus-all decision trees for data stream classification." IEEE Transactions on Knowledge and Data Engineering 21, no. 5 (2009): 624-637. https://doi.org/10.1109/TKDE.2008.181.

[12] Mirzamomen, Zahra, and Mohammad Reza Kangavari. "Evolving Fuzzy Min–Max Neural Network Based Decision Trees for Data

Stream Classification." Neural Processing Letters 45, no. 1 (2017): 341-363. https://doi.org/10.1007/s11063-016-9528-8.

[13] Zhang, Peng, Chuan Zhou, Peng Wang, Byron J. Gao, Xingquan Zhu, and Li Guo. "E-tree: An efficient indexing structure for ensemble models on data streams." IEEE Transactions on Knowledge and Data Engineering 27, no. 2 (2015): 461-474. https://doi.org/10.1109/TKDE.2014.2298018.

[14] Gama, Joao, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. "Learning with drift detection." In Brazilian Symposium on Artificial Intelligence, pp. 286-295. Springer, Berlin, Heidelberg, 2004.

[15] Tatsuya Minegishi, Masayuki Ise, Ayahiko Niimi, Osamu Konishi, "Extension of Decision Tree Algorithm for Stream Data Mining Using Real Data", Fifth International Workshop on Computational Intelligence & Applications IEEE, pg 208-212, 2009.

[16] Mohammad M Masud, Tahseen M, Al-khateeb, Latifur Khan, Charu Aggrawal, Jing Gao, Jiawei Han and Bhawani Thuraisinghum,"Detecting Recurring and Novel classes in Concept Drift Data Streams", IEEE 11th International Conference On Data Mining, pp. 1176- 1181, 2011.

[17] Mohammad M. Masud, Qing Chen, Latifur Khan, Charu C. Aggarwal "Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams", TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE, pp. 1-14, 2011.

[18] Amit Biswas, Dewan Md. Farid and Chowdhary Mofizur Rahman, "A New Decision Tree Learning Approach For Novel Class Detection In Concept-Drifting Data Stream Classification", Journal of computer science and engineering, volume 14, issue 1, July 2012.

[19] Huan Liu, Hiroshi Motoda, Rudy Setiono, Zheng Zhao,"Feature Selection: An Ever Evolving Frontier in Data Mining", Fourth Workshop on Feature Selection in Data Mining, pp. 1-10, 2010.

[20] Divya, G., and M. R. D. BrightAnand. "An Effective Classification and Novel Class Detection of Data Streams." International Journal Of Engineering And Computer Science 3.4 (2014): 5314-5318.

[21] O' Callaghan, N Mishra, A. Meyerson, and S. Guha, "Streaming Data Mining for High-Quality Clustering", International Conference on Data Engineering, pp. 685, 2002. https://doi.org/10.1109/ICDE.2002.994785.

[22] C. C. Aggarwal, J Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams", International Conference on Very Large Database, pp. 81-92, 2003.

[23] C. C. Aggarwal, J Han, J. Wang, and P. S. Yu, "A Framework for Projected Clustering on high dimensional data streams", International Conference on Very Large Database, pp. 81-92, 2004.

[24] F. Cao, M. Ester, W. Qian, and A. Zhou "Density-based clustering over an evolving data streams with noise", SIAM International Conference on Data Mining, vol. 6, 2006.

[25] M. Kamber and J. had, "Data Mining: Concepts and Techniques", Second Edition, Elsevier, 2001.

[26] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani, "An online algorithm for segmenting time series", IEEE international conference on data mining, 2001.

[27] Yusuf, B. Reshma, and P. Chenna Reddy. "Mining data streams using option trees." International Journal of Computer Network and Information Security4.8 (2012): 49.

[28] Nan Jiang and Le Gruenwald, "Research Issues in Data Stream Association Rule Mining", SIGMOD Record, Vol. 35, No. 1, 2006. https://doi.org/10.1145/1121995.1121998.

[29] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining: Concepts and Techniques", Elsevier, 2011.

[30] Domingos, Pedro, and Geoff Hulten. "Mining high-speed data streams". In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 71-80. ACM, 2000. https://doi.org/10.1145/347090.347107.

[31] Florent Masseglia, Pascal Poncelet, Maguelonne Teisseire, "Successes and New Directions in Data Mining", Kluwer Academic Publishers Hingham, MA, USA, Volume 12 Issue 4, Pages 504 - 508, 2009.

[32] Aggarwal, Charu C., et al. "A framework for clustering evolving data streams." Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003.

[33] Abdulsalam, Hanady, David B. Skillicorn, and Patrick Martin., "Streaming random forests". Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International. IEEE, 2007.

[34] Qadeer, Mohammed A., Nadeem Akhtar, and Faraz Khan., "Comparison of Tools for Data Mining and Retrieval in High Volume Data Stream". Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on. IEEE, 2009.

[35] Wang, Aiping, et al. "An incremental extremely random forest classifier for online learning and tracking." Image Processing (ICIP), 2009 16th IEEE International Conference on. IEEE, 2009.

[36] Li, Peipei, Xuegang Hu, and Xindong Wu. "Mining concept-drifting data streams with multiple semi-random decision trees." Advanced data mining and applications (2008): 733-740.

[37] Gama, Joao, Ricardo Rocha, and Pedro Medas. "Accurate decision trees for mining high-speed data streams." In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 523-528. ACM, 2003. https://doi.org/10.1145/956750.956813.

[38] https://www.cs.waikato.ac.nz/ml/weka

[39] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[40] https://moa.cms.waikato.ac.nz/