International Journal of Engineering & Technology, 7 (4.39) (2018) 492-496

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Modern Very Fast Decision Tree Model for Mining High-Speed Time-Series Data Stream

A. Vanitha Katherine[1,*], T. Kamalavalli[2], S. Vinothini[3], M. Jagannath[4], V.E. Jayanthi[5]

[1]*Department of Master of Computer Applications, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India*
[2]*School of Electronics Engineering, Vellore Institute of Technology (VIT) Chennai, Tamil Nadu, India*
[3]*Department of Biomedical Engineering, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India*
**Corresponding author E-mail: avanitha@yahoo.com*

Abstract

Data mining is one of the drastically growing research fields in data analysis. Data is generated on a person, object, element, label in terms of time, days, months, years. Although ample algorithms currently exist for high-speed data streams, they fail to efficiently scale up the data when the data size is large. In this paper, an algorithm is proposed to perform clustering for high-speed data streams using Modern Very Fast Decision Tree (MVFDT) model. It replaces the old decision tree model by clustering to enhance its accuracy. MVFDT takes clusters based records in the database and compares with other cluster of records if any relationship among the records. MVFDT reads a model for clustering which is similar in accuracy of Very Fast Decision Tree (VFDT). In VFDT, new samples are arrived every time for a moving window. But the result of VFDT does not provide satisfactory in terms of data scalability, i.e., large in volume. MVFDT incorporates three different functionalities such as dynamic tree formation, windowing based clustering and classification for calculating the Frequent Pattern (FP) and query process. Experiments are carried out by using large set of time-series and time-changing data streams to compare the clustering and mining efficiency of MVFDT. Experiment results seem to prove that MVFDT model provides more mining efficiency than VFDT.

Keywords: *Data Mining; Data Streams; Very Fast Decision Tree; Tree Mechanism*.

1. Introduction

In the present global technological society, massive streams of data are being generated. These databases are very rich in handling information for decision-making. Most of the organizations hold gigantic databases process millions of data in a single day. An ordered sequence of transactions that arrives timely is called as time series data stream [1,2]. Mining these data streams involves unique opportunities and also brings forth new challenges. Efficient algorithms are available today to dissolve these issues. These algorithms peculiarly concentrates on database mining which does not fit in main memory but sequential scanning of the disk may be required.

Time, memory and sample sizes are the three resources that restrict the knowledge based systems. All these factors are optimized more by clustering the data. Data stream mining techniques are suitable for structured and simple data sets like relational, transferring databases, data warehouses, etc. Since the stream mining algorithms [3-5] need to cope up with availability and adaptability of data due to high fragile nature of data and it is considered as one of the challenges posed by data mining [6,7]. In general, the data mining process analyses the data using clustering as well as different prospectus and merging them into useful information [8]. The characteristics of the data are adapted to the updating mechanism in the form of data stream. With the broad applications in the development of the technology, the data stream bounded to a universal method. Data stream is widely applicable in the transactions in big supermarkets, satellites and on-line data of real time applications with internet basis. The traditional algorithms lead to heavy loss of information and they are also not able to mine from the on-line data environments. The storage of data also seems to be very difficult. Thus, efficient mining has to be introduced to overcome these issues. One of the techniques in the mining process is decision tree model. They have the ability to produce the readable descriptions in the dataset. Decision tree algorithm is used for both discrete and numerical attributes. This model has the anti-noise capability in an efficient manner. This helps in improving the accuracy of the classification of the decision tree and further benefits the reduction of decision tree in whole scale.

There are many decision tree algorithms prevail for the data mining process such as Very Fast Decision Tree (VFDT) with discrete attributes and VFDTc with numerical attributes for clustering and classification [9]. But they have certain disadvantage when they are applied to highly dimensioned attributes, such as difficult to process the data at large volume, enhances multiple passes and at sometimes it gets only one chance in processing of the data. Temporal data is considered for algorithm designed for data stream mining that follows instructions with one pass of data. Speed mining algorithms are designed in order to evaluate the underlying data carefully but they are not feasible to manage the order of the data items during arrival. Therefore the feasibility to save a stream in its entity is a difficult task.

In this paper, how the dynamic random decision tree algorithm helps to enhance the clustering performance of data in terms of virtue of time, accuracy and anti-noise capability are clearly explained. Random decision tree is a heuristic method to compute the information gain and also to obtain the threshold value of attributes with minimum number of split examples. Classifier esti-

mates tree leaves rate of error in the classification and it is applied at leaves of tree to test the data such that the created decision tree is dynamically adjust the height of the tree. Section 2 details on the present status. Section 3 discusses the existing VFDT algorithm. Section 4 illustrates the proposed algorithm. Section 5 depicts the results and discussion. The last section concludes with experimental results.

2. Present Status

To extract large number of interpretable features from time-series, a feature-based approach is introduced to classify time-series data which utilizes a broad database of algorithms [10]. These features are derived as far as their correlation structure, dissemination, stationary and scaling properties to upgrade the scope of time-series models from the scientific time-series analysis. For every time-series, most informative of the class structure is selected from thousands of features in a training set with the help of greedy forward feature selection in addition with linear classifier. Having the limited time-series properties, feature-based classifiers are used to study the variations between classes and deceive and also compute the distance between time-series. This product is set as self-determining training data set. The drawback for instance-based classification is that the process need not require training data to be fetched into memory. Without requiring any sphere of influence on the generated or measured data, the pertinent features are learned repeatedly from the labeled structure of the data set and allow the classifiers become accustomed to the data. The classification of time-series data stream is fast with better efficiency as compared with instance-based classification [10]. The same methodology is attempted directly to time-series data having different lengths and orders of magnitude. The limitation of this method is that the entire data set would be taken as input for the decision tree algorithm.

There are three limitations on traditional cluster ensemble approach. The traditional approach does not require the prior understanding of the data sets. When handling high dimensional data, most of the conventional cluster ensemble methods do not acquire reasonable results and all the ensemble members are considered. Yu et al. [11] proposed an incremental semi-supervised clustering ensemble framework (ISSCE) to tackle the drawbacks of conventional cluster ensemble approaches. To sensibly eliminate the redundant ensemble members, a local cost and a global cost functions, an incremental ensemble member is appropriately designed. The normalized cut algorithm is adopted to serve as the accord function to provide the solution with high stability, robustness and accuracy. Then the newly designed similarity function is adapted to measure of match between two sets of attributes in the sub spaces. Finally, non-parametric tests are used to compare multiple semi-supervised clustering ensemble methods over different data sets.

Cluster becomes a well-known task related with time-series. Due to the availability of large amount of time-series database and its time complexity, different options are used to combine for the measurement of distance, parameter designing and clustering algorithm [12]. For this purpose, a set of pertinent features are described in every time-series database and measure one distance over another. The proposed solution includes the pair of series with a high sufficient association and affords the mean of the complete log values as an estimate. The first technique is a moving average filter that measures the noise level of a time-series database. In this approach, a fixed-sized window is used and every point is substituted with the mean of window values. Shift features are introduced and its characteristics are defined as quite expensive to compute while working with bulk databases. To reduce the computational cost, the time-series database can be sampled since only the general statistics features are calculated. Meanwhile, the applicability of the proposal can be enhanced by minimizing the number of parameters related to performance of the characteristics. It is clear that the relationship between the databases characteristics and the parameters that describe each distance would be helpful for the choice of a distance measure.

Existing survey of feature selection algorithms for classification and clustering and its results are compared with parameters such as data mining tasks, framework based on search strategies and evaluation criteria [13]. The search starts with an unfilled set or filled set or both. Features are added (i.e., forward) consecutively. If it is empty set, features are removed (i.e., backward) successively or add and remove the features in the same time (i.e., bidirectional) at both the ends. Therefore, different strategies have been explored: complete (according to the used evaluation criterion, the optimal results are found), sequential (presence of a threat of losing optimal subsets as it swears out completeness), and random search (similar algorithms are clustered and their merits and demerits are investigated on the same platform.

Mining data streams at real-time signifies one of the finest Wireless Sensor Network (WSN) solutions than other machine learning techniques. To handle the missing data a stream mining algorithm is intended [14]. An experimental outcome of mock and real-life data shows that the mining algorithm is better than the conventional algorithms. The mining algorithm and an auxiliary control make the data mining model to ease the problems of data imperfection in WSNs. The data mining algorithms are applied unremittingly as the segmented data stream. The sliding window width is the length of each data segment. A complete segment enters the VFDT during each window period. The attribute values are considered to appear in synchronization with various sensors and sinks of clusters. Each record is appended with a unique time stamp and it increases in regular intervals. During the test-and-training process of VFDT the values of complete attributes of the identical record are used simultaneously. In WSNs, it is important to monitor the influence of noisy or corrupted data or irregular data stream patterns on data stream mining.

An enhanced very fast decision tree (EVFDT) is the improved version of VFDT [15]. It varies from the existing algorithms on the basis of classification accuracy, tree size, memory, and time. To find the attacks in sensor network, EVFDT is applicable. In EVFDT, a decision tree model algorithm is applied at the victim node. It is proficient to handle corrupted data and detect a Distributed Denial of Service (DDoS) attack effectively with greater accuracy. It permits the legitimate requester to use the resources with less false alarm rate. EVFDT achieved classification accuracy of about 96.5% with 0% noise and 81.5% with 20% noise in dataset. DDoS recognition techniques simulation experiments are carried out to generate attack traffic for evaluation. In high speed data mining, heuristics approaches provide better efficiency whereas, it fails providing efficiency on dynamic abundant high speed data streams. This pave a way to design and develop a new algorithm for high speed data streams mining.

3. Existing VFDT Algorithm

A sample problem is defined in VFDT [11] algorithm as follows: A set of input data stream is taken as N-sample, having a set of entities in the form of (x,y). Here, y is defined as a set of discrete Frequent Pattern (FP) label, where x symbolizes the vector of A attributes. The main objective is to generate a function from the above N-sample $y = f(x)$, which can predict the FP y for future N-samples of x with high accuracy. For example x could be of student's records, and y will be the decision to invite them for Award functions. Among all the classification methods, decision tree learning is the most effective and widely used in data mining research. Here, each node in the tree is assigned with an attribute and test function. The possible outcomes from the node are considered as the branches. An entire leaf node consists of a FP for prediction. While considering the FP, $y=DT(x)$ is obtained by comparing their attributes from the bottom (root) to the top (leaf) by investigating the suitable attributes in the branches. This is

followed by replacing each leaf by test node starting from root. The attributes of each node is tested and selecting the common best attribute among entire nodes in the tree. This best attribute helps to learn and mine the relevant FP in the high speed data streams using any heuristic approaches. Selection of the best attribute is applied on every training sample with each split.

VFDT Algorithm:

Input Parameters:

S	:	Input data stream
A	:	Set of all attributes
$G(\cdot)$	:	Divide function
δ	:	One minute to determine the correct attribute
τ	:	Threshold value
n_{min}	:	Number for verifying the growth

Output Parameters:

DT	:	Decision Tree

Function VFDT (S, A, G, δ, τ)

Assumptions:

Step-1: DT is a tree having a single leaf $l_{1[as\ the\ root]}$
Step-2: $A_1 = A \cup \{A_0\}$
Step-3: $\bar{G}_l(A_0)$ be the $\bar{G}$ fetched by predicting the most FP (Frequent Pattern) S.

Procedure:

For each FP in y_k, for each value x_{ij} of all the attributes $A_i \epsilon X$. Let $n_{ijk}(l_1) = 0$
for every sample (x,y) in S
sort(x,y) into a leaf l using DT
for each x_{ij} in A such that $A_i \epsilon A$
Increment $n_{ijy}(l)$ and label all major FP among samples observed so far at l
Let n_1 is the number of samples seen at l.
If the samples seen so far at l do not belong to the same FP and n_1 mod n_{min} is 0, then
Calculate $\bar{G}_l(A_i)$ for each attribute $A_i \epsilon A - \{A_0\}$ by counting $n_{ijy}(l)$.
A_a is the attribute with highest $\bar{G}_l$
A_b is the attribute with second highest $\bar{G}_l$
Compute $\varepsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n}}$
$\Delta\bar{G}_l = \bar{G}_l(A_a) - \bar{G}_l(A_b)$
If $((\Delta\bar{G}_l > \varepsilon)$ or $(\Delta\bar{G}_l \leq \varepsilon < \tau))$ and $A_a \neq A_b$, then replace l by an internal node the divides on A_a
for every split's branch, add a new leaf l_m, and let $A_m = A - \{A_a\}$
Let $\bar{G}_m(A_0)$ be the $\bar{G}$ achieved by forecasting the most FP at l_m
for each FP y_k and each value x_{ij} of all attribute $A_i \epsilon A_m - \{A_0\}$, let $n_{ijk}(l_m) = 0$
Return DT.

3.1. Description

To determine the size of the sample data required for each decision in experimental evaluation, VFDT utilizes a range (R) for all random values. Also, the computed mean should satisfy the equation (1).

$$\varepsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n}} \quad (1)$$

where, R is the Hoeffding bounds [8], n is the number of samples, $1 - \delta$ is the confidence factor. The generated observations are true regardless of the probability distribution. Let $G(A_i)$ be the heuristic measure used for selecting the test attributes. After fetching n samples at the leaf node, the A_a be the attribute with the heuristic measure and A_b be the attribute with second best. $\Delta\bar{G}_l = \bar{G}_l(A_a) - \bar{G}_l(A_b)$ is the random variable, the difference among the experimental heuristic values. Verifying the Hoeffding bound with $\Delta\bar{G} > \varepsilon$ and $\bar{G}_l(A_a)$ is greater than zero, then select A_a is the split attribute. n_{ijk} is the count for computing heuristic measures. The above VFDT algorithm given is only for symbolic attributes, we can also utilize it for the numerical attributes. The value of S may be infinitive, means the function never terminates and the sub procedure running parallel can predict the DT as the current tree to make FP predictions.

VFDT cannot provide accurate solution while the data becomes abundant and the computation cost, speed and split data become problematic. Also, the results remain unchanged in RAM, due to abundant data. In VFDT, the comparison is applied only for the first time data samples in the split attribute of a given node. Thus, VFDT is not efficient for dynamic data with dynamic attributes. In order to overcome the drawbacks mentioned above, here we proposed a Modified-VFDT algorithm which provides better performance in terms of scalability, time, dynamic attribute verification, execution time and cost.

4. Proposed MVFDT Algorithm

Modified Very Fast Decision Tree (MVFDT) is the modified form of VFDT algorithm which detects and responds to the dynamic nature of the data in terms of attributes in the tree. Clustering is applied on the attributes to improve the efficiency of the data mining. The old VFDT method is applied and utilized for the first time data processing in MVFDT. Then it is extended by adding a sliding window to learn the attributes of the new samples when a new data comes. Each time the new data is entered, MVFDT calls the clustering function dynamically and the new data is persisted in the appropriate cluster in terms of their attributes. If it finds a new attribute or new entity on the data, it updates the attribute list A, by increasing the counts and creates a new cluster. The new labels, attribute and FP are added each time a new data comes dynamically. Whenever the attribute of the newly arrived samples has high accuracy, then it replaces the old one. Dynamic clustering is one of the main added advantages in MVFDT. The pseudo code for MVFDT is given below and describes the entire functionality.

Assumptions:

Step-1: DT is a tree having a single leaf $l_{1[as\ the\ root]}$
Step-2: $A_1 = A \cup \{A_0\}$
Step-3: $\bar{G}_l(A_0)$ be the $\bar{G}$ fetched by predicting the most FP (Frequent Pattern) S.
Step-4: $New(l)$ be the new attribute for l and it is empty in the beginning.
Step-5: W is the sliding window and it is empty initially.
Step-6: AA be the data differently [new] entered into MVFDT.

Procedure:

For each FP in y_k, for each value x_{ij} of all the attributes $A_i \epsilon X$. Let $n_{ijk}(l_1) = 0$
for every sample (x,y) in S
sort(x,y) into a leaf l using DT
for each x_{ij} in A such that $A_i \epsilon A$
If $x_{ij} = AA$ then AA is updated in A
Increment $n_{ijy}(l)$ and label all major FP among samples observed so far at l
Let n_1 is the number of samples seen at l.
for $p = 1$ to W step w
If the samples seen so far at l do not belong to the same FP and n_1 mod n_{min} is 0, then
Calculate $\bar{G}_l(A_i)$ for each attribute $A_i \epsilon A - \{A_0\}$ by counting $n_{ijy}(l)$.
A_a is the attribute with highest $\bar{G}_l$
A_b is the attribute with second highest $\bar{G}_l$
Compute $\varepsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n}}$
$\Delta\bar{G}_l = \bar{G}_l(A_a) - \bar{G}_l(A_b)$

If $((\Delta\bar{G}_l > \varepsilon)$ or $(\Delta\bar{G}_l \leq \varepsilon < \tau))$ and $A_a \neq A_b$, then replace l by an internal node the divides on A_a
for each branch of the split, add a new leaf l_m, and let $A_m = A - \{A_a\}$
Let $\bar{G}_m(A_0)$ be the $\bar{G}$ achieved by estimating the most FP at l_m
for each FP y_k and each value x_{ij} of all attribute $A_i \epsilon A_m - \{A_0\}$, let $n_{ijk}(l_m) = 0$
End p
Return DT.

5. Results and Discussion

Experiments are carried out to compare the clustering and mining efficiency of MVFDT against VFDT. Objective of the work is to estimate the ability to scale up by evaluating MVFDT model with different drift levels and also to illustrate the circumstances where MVFDT performs well as compared to the other system model. The mock data is taken from UCI repository [16] for validating VFDT and MVFDT algorithm. The mock data is used in the experiments as a varying concept based on a hyper plane. A hyper plane in d-dimensional space is the set of points **X** that satisfy the equation (2)

$$\sum_{i=1}^{d} w_i x_i = w_0 \qquad (2)$$

where, x_i is the i^{th} coordination of x. Examples, for which $\sum_{i=1}^{d} w_i x_i \geq w_0$ are labeled positive, and examples for which $\sum_{i=1}^{d} w_i x_i < w_0$ are tagged negative. To simulate time-changing concepts, hyper planes are used as the orientation and position of the hyper plane that can be altered using the relative size of the weights. Weight is utilized as an index for the magnitude of each data which can maintain the dimension consists most of the data information. To process, only the non-zero weight of the data is considered. In a decision tree representing the hyper plane, the attributes relative information can be controlled by changing the most favorable order of tests. The experiment is done on a 4GB RAM, Pentium- Core2-Duo machine with 500GB hard disk, running Windows.
Initially, the experiment is carried out on the mock data and compares the ability of MVFDT with the VFDT to decide its efficiency for large set of data having drifts. In terms of weight, the drifts are created in the data to verify the errors and they are relabeled with the updated concept. The drift point is modified according to the data magnitude and the size of the data. Precision of the algorithm as a function of d, the dimension of the space is illustrated in Figure 1. By testing the precision of the well-read models on every 10,000 sample data the described values are attained. The minor axis describes the drift levels as the average percentage of the test set. The concept varies by altering the label at each point.

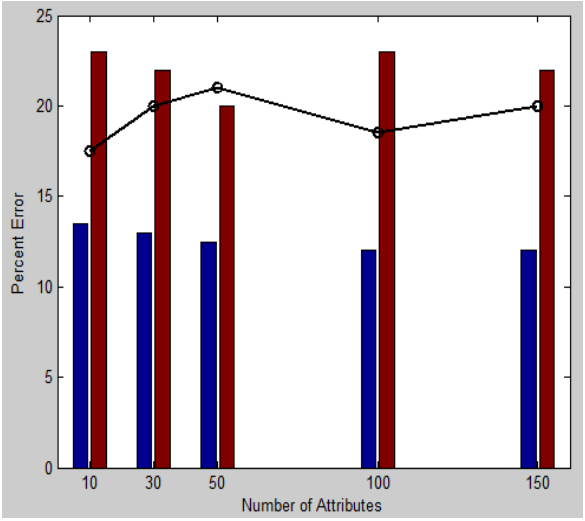

Figure 1: Number of attribute versus error detection.

MVFDT is considerably more precise than VFDT by approximating 10% on the mean value and performance of MVFDT is enhanced a little by rising d-dimensional space. Figure 2 shows the average model size that depicts the number of nodes. The advantage of MVFDT over VFDT is that MVFDT is consistent across every single value of d. The accuracy and the size are derived from the reality that MVFDT model is constructed on the 100000 most appropriate examples, while VFDT model is developed on millions of out-of-date examples.

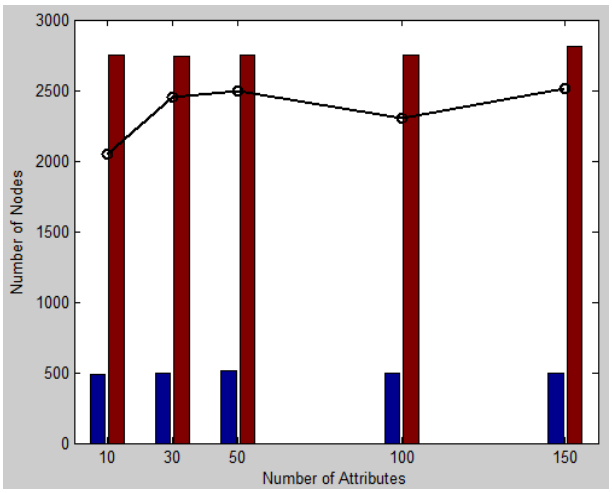

Figure 2: Number of nodes versus number of attributes.

Finally, how the concept of changing the levels of MVFDT drift are examined on all the five data sets having d = 50. The drift is indicated with a parameter D. For every 75,000 examples, D of the concept hyper plane's weights are chosen at random and modified as before, $w_i = w_i + 0.01d\sigma_i$. The comparison on these data sets is shown in Figure 3. The MVFDT substantially outperforms VFDT at each drift level. The larger variance in VFDT's data points can be observed if VFDT's error rate comes within reach of 50% for D > 2. MVFDT's error rate appears to develop smoothly with raise in the concept level change. The result reveals the fact that the drift adjustments are effective and robust.

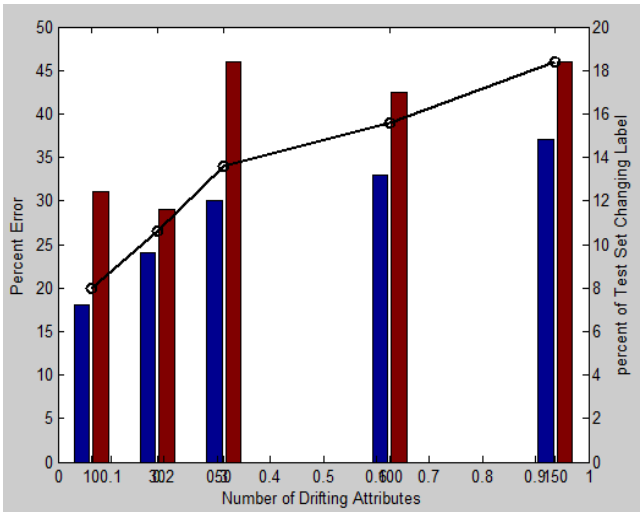

Figure 3: Number of attributes drift versus error detection.

The time taken for analyzing and clustering the data in terms of scalability is shown in Figure 4. According to the size of the data the time taken to process the data is proportionally increased. Since providing a solution for high dimensional data in terms of scalability will be considered in our future work.
The entire experiment shows that MVFDT has the capability to react to the drift are the gain over VFDT. To estimate the nature of the drift, the MVFDT may be run with diverse dynamic data sizes.

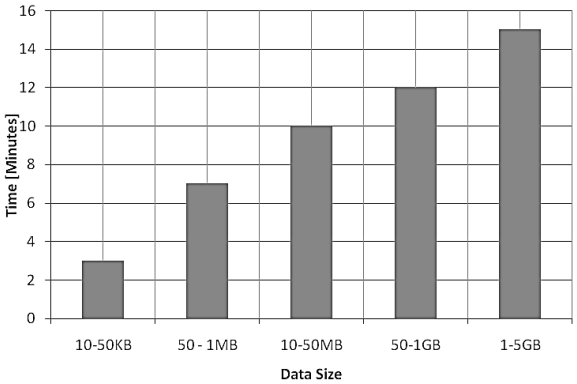

Figure 4: Time taken for analyzing and clustering the data in terms of scalability.

6. Conclusion

The MVFDT model is introduced for dynamic random decision tree induction system. The proposed model seems to be an accurate model for high-speed and concept drifting data streams. This model preserves a decision tree latest by holding a small constant time for every arrival of new example. Every time, a new example would arrive by applying a conventional learner that decides the output accuracy. The model is up-to-date with a enormous data streams which is shown to be the empirical studies of MVFDT.

Acknowledgement

Authors would like to thank all researchers for their contribution which eventually enhance the quality of our paper.

References

[1] A. Bifet, R. Kirkby, Data Stream Mining a Practical Approach, Technical Report, University of WAIKATO, 2009.
[2] D. Brzezinski, Mining Data Streams with Concept Drift, Master's Thesis, Poznan University of Technology, Poznan, Poland, 2010. Available at: http://www.cs.put.poznan.pl/dbrzezinski/publications/ConceptDrift.pdf.
[3] K. Patel, Review on data stream classification, In Proceedings of the International Conference on Computing and Information Technology, Tirupati, India, 2012, pp. 13–35.
[4] P.L. Barlett, S. Ben-David, S.R. Kulkarni, Learning changing concepts by exploiting the structuer of change, Machine Learning, Vol. 41, No. 2, 2000, pp. 153–174.
[5] V. Ganti, J. Gehrke, R. Ramakrishnan, DEMON: Mining and monitoring evolving data, In Proceedings of the 16th International Conference on Data Engineering, San Diego, CA, USA, 2000, pp. 439–448.
[6] G. Hulten, L. Spencer, P. Domingos, Mining time-changing data streams, In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, 2001, pp. 97–106.
[7] C.C. Aggarwal, J. Han, J. Wang, P.S. Yu, A framework for clustering evolving data streams. In Proceedings of the 29th International Conference on Very Large Data Bases, 2003, pp. 81–92.
[8] C.C. Aggarwal, J. Han, J. Wang, P.S. Yu, A framework for projected clustering of high dimensional data streams, In Proceedings of the 13th International Conference on Very Large Data Bases, Toronto, Canada, 2004, pp. 852–863.
[9] P. Domingos, G. Hulten, Mining high speed data streams. In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, USA, 2000, pp. 71–80.
[10] B.D. Fulcher, N.S. Jones, Highly comparative feature-based time-series classification, IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 12, 2014, pp. 3026–3037.
[11] Z. Yu, P. Luo, J. You, H. Wong, H. Leung, S. Wu, J. Zhang, G. Han, Incremental semi-supervised clustering ensemble for high dimensional data clustering, IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 3, 2016, pp. 701–714.
[12] U. Mori, A. Mendiburu, J.A. Lozano, Similarity measure selection for clustering time series databases, IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 1, 2016, pp. 181–195.
[13] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 4, 2005, pp. 491–502.
[14] H. Yang, S. Fong, G. Sun, R. Wong, A very fast decision tree algorithm for real-time data mining of imperfect data streams in a distributed wireless sensor network, International Journal of Distributed Sensor Networks, Vol. 8, No. 2, 2012, pp. 863545.
[15] R. Latif, H. Abbas, S. Latif, EVFDT: An enhanced very fast decision tree algorithm for detecting distributed denial of service attack in cloud-assisted wireless body area network, Mobile Information Systems, Vol. 2015, Article ID 260594, 2015, pp. 1–13.
[16] http://www.USI/repoistory/syntheticdataset.html [Accessed on July 18, 2018].