Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 5, 330-334 (2021)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Enhancing Decision Trees for Data Stream Mining

Mostafa Yacoub[1,*], Amira Rezk[1], Mohamed Senousy[2]

[1]Faculty of Computers and Information, Information System Department, Mansoura University, Mansoura, 35511, Egypt

[2]Faculty of Management Sciences, Computer and Information system Departments, Sadat Academy for Management Sciences, Cairo, 00202, Egypt

ARTICLE INFO

Article history:
Received: 29 July, 2021
Accepted: 15 October, 2021
Online: 23 October, 2021

Keywords:
Data Stream Mining
Classification
Decision Trees
VFDT

ABSTRACT

Data stream gained obvious attention by research for years. Mining this type of data generates special challenges because of their unusual nature. Data streams flows are continuous, infinite and with unbounded size. Because of its accuracy, decision tree is one of the most common methods in classifying data streams. The aim of classification is to find a set of models that can be used to differentiate and label different classes of objects. The discovered models are used to predict the class membership of objects in a data set. Although many efforts were done to classify the stream data using decision trees, it still needs a special attention to enhance its performance, especially regarding time which is an important factor for data streams. This fast type of data requires the shortest possible processing time. This paper presents VFDT-S1.0 as an extension of VFDT (Very Fast Decision Trees). Bagging and sampling techniques are used for enhancing the algorithm time and maintaining accuracy. The experimental result proves that the proposed modification reduces time of the classification by more than 20% in more than one dataset. Effect on accuracy was less than 1% in some datasets. Time results proved the suitability of the algorithm for handling fast stream mining.

1. Introduction

Recently, information played a major role in our world. Subsequently, the process of extracting knowledge is becoming very important. New applications that depend on data streams became more popular with time. Stream data are clear in sensors, telephone call records, click streams, social media, and stock market.

Contrary to traditional data mining, which analyses a stored data set, the stream mining analyses a data stream which cannot be saved as it's infinite and needs expensive storage capabilities. Data streams arrive continuously and with fast pace, this prevents multiple passes of the data. So, processing time is more constrained in data streams.

Classification is a mining technique used to build a classification model based on the training data set which used to predict the class label of a new undefined data. Decision trees, neural networks, Bayesian networks, and Support Vector machines (SVM) are considered the most effective methods of classification. Decision trees are data structures organized hierarchically by splitting input space into local zones to predict the dependent variable.

Decision trees are hierarchical data structures for supervised learning by which the input space is split into local regions to predict the dependent variable [1]. It is classified as greedy algorithms which try to find a decision at each step of small steps. Decision trees consist of nodes and edges (branches). Root node has no incoming edge. Leaves or terminal nodes have no outgoing edges. All other nodes – besides root – have exactly one ingoing edge. Internal or test nodes are the nodes with outgoing edges. Each internal node splits the instance space into two or more instance sub-space. These splits are done according to a specific splitting discrete function of attribute values (inputs). Classes are assigned to leaf nodes.

Decision trees are characterized by simplicity, understandability, flexibility, adaptability and higher accuracy [2], [3]. The ability to handle both categorical and continuous data is an important advantage of decision trees. So, there is no need to normalize the data before running the decision tree model, that means fewer preprocessing processes. Being easier to construct and understand is another important factor for preferring decision

*Corresponding Author: Mostafa Yacoub, Email: mostafayacoub3@gmail.com

https://dx.doi.org/10.25046/aj060537

trees over other data mining techniques. In addition, decision trees are interpretable as it can be expressed as a logical expression. Missing values in data are considered issues need to be handled before running data mining techniques in order not to affect the results. Decision trees can handle data with missing values successfully.

Traditional decision tree learners like ID3 (Iterative Dichotomiser 3) and C4.5 (Classification 4.5) have problems in handling data streams. It presumes that the whole training examples can be stored concurrently in main memory, which is not valid in data streams [4].

Very Fast Decision Trees (VFDT) was introduced by Domingos and Hulten in 2000[5]. VFDT uses the Hoeffding bound for node splitting and creating Hoeffding trees. The basis of Hoeffding trees is "a small sample can often be enough to choose an optimal splitting attribute". Hoeffding bound gives a mathematical support to that basis quantifying the number of examples needed to estimate some statistics within a prescribed accuracy [6].

According to Hoeffding bounds, with probability 1 – δ, the true mean of r is at least $\bar{r}$ – ε, where

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

In equation (1), *r* represents continuous random variables whose range is R. $\bar{r}$ is the observed mean of the samples after *n* independent observations. [7]. The VFDT defines the two attributes t_1, t_2 with highest information gain G_{t1} and G_{t2}. If $\triangle G = G_{t1} - G_{t2}$ is higher than Ɛ (equation 1), then G_{t1} is the best split attribute with probability of $1 - \delta$ and the split is done. (Algorithm 1: VFDT)

In VFDT, leaves are replaced with decision nodes recursively. Statistics about attributes values are saved in each leaf. Based on these statistics, a heuristic function calculates the value of split tests. Each new instance passes from root to a leaf. At each leaf, attribute evaluation is done and follow the branch according to evaluation result. An important step must be done, which is updating the enough statistics of the leaf [8].

VFDT can address the research issues of data streams such as ties of attributes, bounded memory, efficiency and accuracy[9]. VFDT is known for having decent memory management. It can save memory by deactivating less promising leaves when memory reaches a limit then it turns back to normal when memory is free[10]. Also, it monitors the available memory and prunes leaves (where sufficient statistics are stored) depending on recent accuracy [11], [12].

The rest of this paper will discuss the related work in section two, the proposed modification on VFDT in section three, the evaluation of the proposed modification in section four and finally the conclusion and future work in section five.

Algorithm 1: VFDT

Result: very fast decision tree
begin
Let T be a tree with a one leaf (the root)
for all training examples **do**
  Update sufficient statistics in *l*
  Increment n_1, the number of examples seen at *l*
  if n_1 mod $n_{min} = 0$ **and** all examples seen at *l* not all same class **then**
    Compute $\bar{G}_l(Xi)$ for each attribute
    Let X_a be the attribute with highest $\bar{G}_l$
    Let X_b be the attribute with the second highest $\bar{G}_l$
    Compute Hoeffding bound $\mathcal{E}=\sqrt{\frac{R^2\ln(\frac{1}{\delta})}{2nl}}$
    if $X_a \neq X_\Phi$ **and** $(\bar{G}_l(X_a) - \bar{G}_l(X_b)) > \mathcal{E}$ **or** $\mathcal{E} < \tau$ **then**
      Replace *l* with an internal node that splits on X_a
      for all branches of the split **do**
        Add new leaf with initialized sufficient statistics
      end
    end
  end
end

2. Related Work

Although decision trees have more than accepted results in data stream mining, there have been many trials of modification to enhance results. For being one of the noticeable algorithms in decision trees, VFDT has share in these studies. Following studies present VFDT modifications to achieve higher accuracy, less time, or both. Next section summarizes these studies followed by a table to show impact on time and accuracy.

2.1. Bagging

In [13], the author proposed VFDTc and VFDTcNB, which can include and classify new data online with a one scan of the data for medium and large size datasets. VFDTc can deal with numerical attributes heterogeneous data, while VFDTcNB can apply naive Bayes classifiers in tree leaves and reinforces the anytime characteristic. In [14], the authors presented GVFDT, an employment of the VFDT used for creating random forests that use VFDTs for GPUs data streams. This technique takes advantage of the huge parallel architecture of GPUs. Furthermore, GVFDT algorithm reduces the communication between CPU and GPU by constructing the trees inside the GPU.

2.2. Adaptability

In [15], the authors proposed Strict VFDT in two versions; SVFDT-I and SVFDT-II. Both are seeking reducing tree growth and decreasing memory usage. Both algorithms produce trees much smaller than those produced by the original VFDT algorithm. Testing them on eleven datasets, SVFDT-II produced better accuracy than the SVFDT-I, together with significantly reducing tree size.

In [16], the authors presented ODR-ioVFDT (Outlier Detection incremental optimized VFDT) as an extension of VFDT to handle outliers in continuous data learning. The new algorithm was applied onto bioinformatics data streams–loaded by sliding windows – to diagnose and treat disease more efficiently. The ODR model chooses the outlier, which is stored into misclassified database. Clean data will be passed through ioVFDT classifier for decision tree building. The lower performance will send response to outlier and classifier model, the model update will be needed. In

[17], the authors proposed an optimization of VFDT algorithm to decrease the effect of concept drift by utilizing sliding windows and fuzzy technology. Results showed improvements in accuracy results.

Table 1: Summary of related work

Title	Year	Algorithm Name	Algorithm Idea	Time Results	Accuracy Results
Speeding up Very Fast Decision Tree with Low Computational Cost	2020	IMAC (Incremental Measure Algorithm Based on Candidate Attributes)	The algorithm calculates the heuristic measure of an attribute with lower computational cost. Possible split timing is found by selecting subset of attributes precisely.	Decreased in most datasets except two with minor increase	No loss in some datasets and minor loss of accuracy in few datasets
A VFDT algorithm optimization and application thereof in data stream classification	2020	Optimized VFDT	an optimization of VFDT algorithm to decrease the effect of concept drift by utilizing sliding windows and fuzzy technology	Lower Time	Higher Accuracy
Enhancing Very Fast Decision Trees with Local Split-Time Predictions	2018	OSM (One-sided minimum)	replaced the global splitting scheme with local statistics to predict the split time which leads to lower computational cost by avoiding excessive split tries.	Decreased run-time	Same accuracy
Strict Very Fast Decision Tree: a memory conservative algorithm for data stream mining Victor	2018	Strict VFDT: SVFDT-I & SVFDT-II	Both are seeking reducing tree growth and decreasing memory usage. Both algorithms produce trees much smaller than those produced by the original VFDT algorithm.	Decreased in 3 datasets, and higher in the other 8 datasets	Decreased in 5 datasets, same accuracy in 3 datasets, and higher accuracy in 3 more
Robust High-dimensional Bioinformatics Data Streams Mining by ODR-ioVFDT	2017	ODR-ioVFDT	The ODR model chooses the outlier, which is stored into misclassified database. Clean data will be passed through ioVFDT classifier for decision tree building. The lower performance will send response to outlier and classifier model, the model update will be needed.	Higher in all datasets	Higher in all datasets with small percentage
Random Forests of Very Fast Decision Trees on GPU for Mining Evolving Big Data Streams	2014	GVFDT: Very Fast Decision Trees for GPU	This technique takes advantage of the huge parallel architecture of GPUs. Furthermore, GVFDT algorithm reduces the communication between CPU and GPU by constructing the trees inside the GPU.	Lower time in the three datasets	Lower Accuracy in two datasets and same accuracy in one.
Accurate Decision Trees for Mining High-speed Data Streams	2003	VFDTc & VFDTcNB	VFDTc: can deal with numerical attributes. VFDTcNB: apply naive Bayes classifiers in tree leaves	Decrease with more than 50%	Increase by 2% (average)

2.3. Split Function

In [18], the authors replaced the global splitting scheme with local statistics to predict the split time which leads to lower computational cost by avoiding excessive split tries. Results showed decreased run-time with no loss in accuracy. In [19], the authors introduced IMAC (Incremental Measure Algorithm Based on Candidate Attributes) an online incremental algorithm with a much lower computational cost. The algorithm calculates the heuristic measure of an attribute with lower computational cost. Possible split timing is found by selecting subset of attributes precisely. The algorithm showed faster and more accurate results by decreasing split attempts with much lower split delay.

Table 1 summarizes efforts in this area, but the time still a challenge that face the algorithms that applied to the stream data. All mentioned studies achieved better time results except on research. From accuracy side, only three studies achieved higher accuracy and another two achieved less accuracy. So, this paper will try to propose a modification to reduce the time of the decision tree in stream data.

3. The proposed VFDT-S1.0

The proposed VFDT-S1.0 aims to modify the original VFDT algorithm to reduce the time of classification. The idea of the modification is based on two main factors. First is bagging more than one algorithm to improve performance and second factor is using random sampling with fixed percentage from the whole data.

Algorithm 2: VFDT-S1.0

Result: M: Model with the highest accuracy
begin
Load Data Stream S
For every record in S:
Delete record if contains null value
Let $S_{train} = S * 0.8$
$S_{test} = S - S_{train}$
S_{train} = SimpleRandomSample(S_{train})
HT=HoeffdingTree(S_{train})
HT_{Pred}=Predict(HT,S_{test})
HT_{Acc}=mean(HT_{Pred} , S_{test}Class)*100
HOT=HoeffdingOptionTree(S_{train})
HOT_{Pred}=Predict(HOT,S_{test})
HOT_{Acc}=mean(HOT_{Pred} , S_{test}Class)*100
HAT=HoeffdingAdaptiveTree(S_{train})
if (HTAcc > HOTAcc **and** HTAcc > HATAcc) **then**
  M = HT
else
  if ($HOT_{Acc} > HT_{Acc}$ & $HOT_{Acc} > HAT_{Acc}$) **then**
    M = HOT
  else
    if ($HAT_{Acc} > HT_{Acc}$ & $HAT_{Acc} > HOT_{Acc}$) **then**
    M = HAT
    end
  end
end
end

The three algorithms are run sequentially to find the one with more accurate results. Accuracy is measured for the three models generated by the three algorithms. The algorithm with highest accuracy is used on the rest of data.

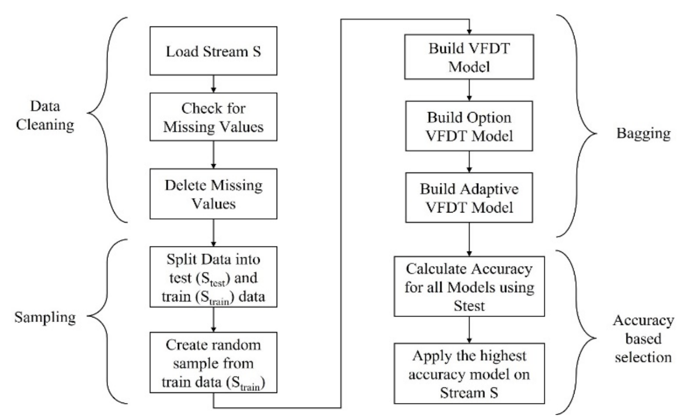

Figure 1: VFDT-S1.0 Framework

Sampling is used to compensate using three different algorithms sequentially. Using sampling in data streams has been discussed in many studies. Three sampling techniques related to data streams are reservoir sampling, AMS-sampling, and Sliding window sampling. In [20], random sampling was used to challenge time constraint. As shown in figure 1, the three algorithms were trained using the same sample. As we choose the best accuracy of the three to use and compare with original VFDT algorithm. Figure 1 displays VFDT-S01 framework, explaining the four basic stages of it.

4. Implementation and Evaluation

To examine the proposed algorithm, it is tested and compared to the original VFDT algorithm. Coding and evaluation were done using Java and R languages working on Microsoft Windows 10 environment on core i5-5200U processor machine. Source code of algorithms is written in Java in Massive Online Analytics (MOA) tool, employing MOA codes in R is done by using RMOA package. RMOA is connecting R with MOA to build classification and regression models on streaming data.

The test is done using 7 different real classification datasets; covType[21], Airlines[22], KDD99[23], Elecnorm[24], MplsStops[25], Chess[26], and Income[27]. Table 2 summarizes the seven datasets and comparing them according to number of instances, attributes, and classes.

Table 2: Sample Table

Dataset	Number of Instances	Number of attributes	Number of Classes
covType	581,012	55	7
Airlines	539,383	8	2
KDD99	494,020	42	23
Electricity	45,312	9	2
MplsStops	51,920	15	2
Chess	28,056	7	18
Income	48,842	15	2

Each dataset was divided into training and test set. Training set is 80% the whole data and the reminder was the test set for prediction. Both algorithms were tested using the same test set to get more accurate comparison results. Accuracy was calculated as number of true predictions divided by test set size.

Time was calculated by using built-in time function in R at the start and end of code. Both accuracy and time were calculated as an average of three runs of both algorithms on every dataset.

Table 3 compares the proposed VFDT-S1.0 and VFDT based on the accuracy and time. Also shows that the original algorithm achieves higher accuracy in all seven datasets.

Table 3: Algorithms Comparison

	VFDT		VFDT-S1.0		Difference Percentage	
Data set	Accuracy%	Time (sec)	Accuracy %	Time (sec)	Accuracy %	Time %
CovType	72.86 %	816.00	69.95 %	620.74	-4.00 %	-23.93 %
Airline	65.06 %	635.10	60.93 %	539.86	-6.34 %	-15.00 %
KDD99	99.79 %	638.39	99.55 %	492.65	-0.24 %	-22.83 %
Elec.	77.11 %	52.70	76.38 %	45.11	-0.94 %	-14.40 %
MplsStop	79.53 %	20.12	77.91 %	18.60	-2.04 %	-7.54 %
Chess	33.70 %	29.71	32.29 %	27.06	-4.18 %	-8.92 %
Income	83.94 %	53.18	81.92 %	46.51	-2.40 %	-12.53 %

Differences between VFDT accuracy and VFDT-S1.0 accuracy varies from 0.24% at KDD99 dataset to 4.13% at Airline dataset. Figure 2 displays the accuracy between the two algorithms.

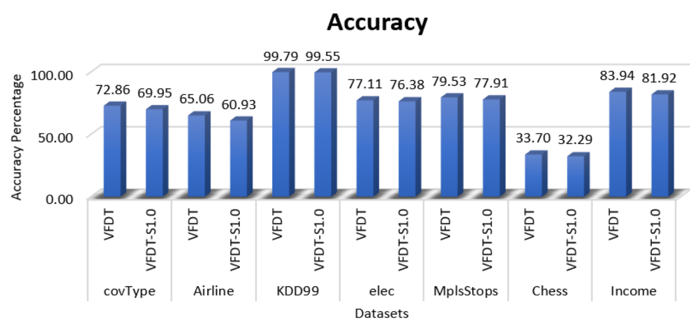

Figure 2: Accuracy Comparison on all datasets

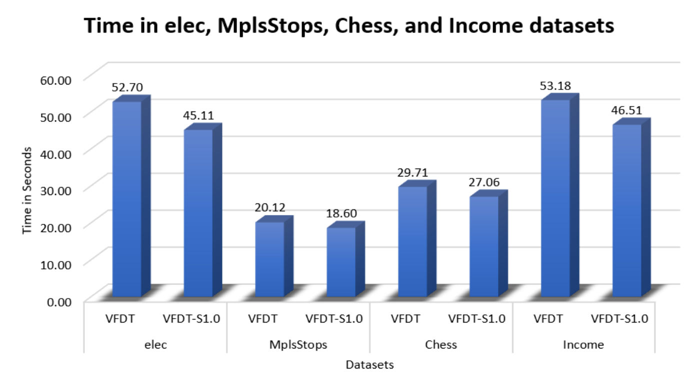

Figure 3: Time Comparison on datasets (covType, Airline and KDD99)

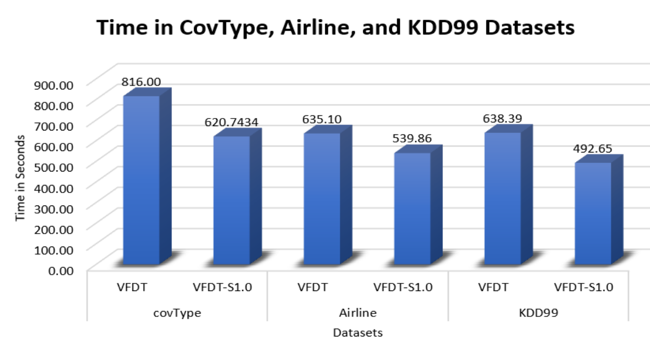

Figure 4: Time Comparison on datasets (elec, MplsStops, Chess, and Income)

Figure 3 represents processing time of both algorithms on largest three datasets and figure 4 displays time on the reminder datasets. Time was always better with VFDT-S1.0 at all datasets. 1.52 seconds was the minimum difference between two algorithms on MplsStops dataset. CovType dataset had the major difference with 195.26 seconds. At KDD99 dataset, which had the highest accuracy difference, the time was less by 145.74 seconds.

5. Conclusion

This paper proposed the VFDT-S1.0; a modified VFDT algorithm that uses bagging techniques to achieve most possible accuracy. In time factor, we used random sampling to achieve better processing time. We tested the new algorithm using seven real classification datasets and compared results with VFDT algorithm. Improvements have been noticed in time as VFDT-S1.0 took much less time with all datasets. Biggest time difference was 24% in CovType dataset. In KDD dataset the time dropped by 23% with -0.2% in accuracy. This time difference shows potential for scaling VFDT. As it can be processed by much lower processing resources. Also, the ability to handle very fast data streams with dependable accuracy.

6. Future Work

In future work, tree size, Kappa, sensitivity, and specificity will be measured for both algorithms. Accuracy can be enhanced with bagging more models and choosing a sample with the same class representation in dataset. Also, parallel processing is considered for much time improvement. Change detection techniques are going to be added to deal with concept drifts.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] E. Alpaydın, "Introduction to machine learning," Methods in Molecular Biology, **1107**, 105–128, 2014, doi:10.1007/978-1-62703-748-8-7.

[2] Z. Çetinkaya, F. Horasan, "Decision Trees in Large Data Sets," International Journal of Engineering Research and Development, **13**(1), 140–151, 2021, doi:10.29137.

[3] S. Moral-garcía, J.G. Castellano, C.J. Mantas, A. Montella, J. Abellán, "Decision Tree Ensemble Method for Analyzing Traffic Accidents of Novice Drivers in Urban Areas," 1–15, 2019, doi:10.3390/e21040360.

[4] F.M.J.M. Shamrat, R. Ranjan, A. Yadav, A.H. Siddique, S. Engineering, C. Neusoft, C.C. Officer, "Performance Evaluation among ID3 , C4 . 5 , and CART Decision Tree Algorithms," International Conference on Pervasive Computing and Social Networking, 2021.

[5] P. Domingos, G. Hulten, "Mining high-speed data streams," Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '00, 71–80, 2000, doi:10.1145/347090.347107.

[6] M. Yacoub, A. Rezk, M. Senousy, "Adaptive classification in data stream mining," Journal of Theoretical and Applied Information Technology, **98**(13), 2637–2645, 2020.

[7] W. Zang, P. Zhang, C. Zhou, L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," Journal of Big Data, **1**(1), 1–16, 2014, doi:10.1186/2196-1115-1-5.

[8] J. Gama, P.P. Rodrigues, An Overview on Mining Data Streams, Springer-Verlag Berlin Heidelberg: 38–54, 2009, doi:10.1007/978-3-642-01091-0.

[9] C.C. Aggarwal, Data streams: Models and Algorithms, 1st ed., Springer-Verlag US, 2010, doi:10.1007/978-0-387-47534-9.

[10] E. Ikonomovska, J. Gama, S. Džeroski, "Learning model trees from evolving data streams," Data Mining and Knowledge Discovery, **23**(1), 128–168, 2011, doi:10.1007/s10618-010-0201-y.

[11] A. Muallem, S. Shetty, J.W. Pan, J. Zhao, B. Biswal, "Hoeffding Tree Algorithms for Anomaly Detection in Streaming Datasets : A Survey," Journal of Information Security, **8**(4), 339–361, 2017, doi:10.4236/jis.2017.84022.

[12] D.H. Han, X. Zhang, G.R. Wang, "Classifying Uncertain and Evolving Data Streams with Distributed Extreme Learning Machine," Journal of Computer Science and Technology, **30**(4), 874–887, 2015, doi:10.1007/s11390-015-1566-6.

[13] J. Gama, R. Rocha, P. Medas, "Accurate Decision Trees for Mining High-speed Data Streams," Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 523–528, 2003, doi:10.1145/956750.956813.

[14] D. Marron, A. Bifet, G. De Francisci Morales, "Random forests of very fast decision trees on GPU for mining evolving big data streams," Frontiers in Artificial Intelligence and Applications, **263**, 615–620, 2014, doi:10.3233/978-1-61499-419-0-615.

[15] V. Guilherme, A. Carvalho, S. Barbon, "Strict Very Fast Decision Tree : a memory conservative algorithm for data stream mining," Pattern Recognition Letters, 1–7, 2018.

[16] D. Wang, S. Fong, R.K. Wong, S. Mohammed, J. Fiaidhi, K.K.L. Wong, "Robust high-dimensional bioinformatics data streams mining by ODR-ioVFDT," Scientific Reports, **7**, 1–12, 2017, doi:10.1038/srep43167.

[17] S. Jia, "A VFDT algorithm optimization and application thereof in data stream classification A VFDT algorithm optimization and application thereof in data stream classification," Journal of Physics: Conference Series, 1–7, 2020, doi:10.1088/1742-6596/1629/1/012027.

[18] V. Losing, H. Wersing, B. Hammer, "Enhancing Very Fast Decision Trees with Local Split-Time Predictions," IEEE International Conference on Data Mining (ICDM), 287–296, 2018, doi:10.1109/ICDM.2018.00044.

[19] J. Sun, H. Jia, B. Hu, X. Huang, H. Zhang, H. Wan, X. Zhao, "Speeding up Very Fast Decision Tree with Low Computational Cost," International Joint Conferences on Artificial Intelligence, 1272–1278, 2020.

[20] E. Ikonomovska, M. Zelke, Algorithmic Techniques for Processing Data Streams, 2013.

[21] J A Blackard D J Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," Computers and Electronics in Agriculture, **24**, 131--151, 1999.

[22] E. Ikonomovska, Airline, 2009.

[23] S.D. Hettich, S. and Bay, The UCI KDD Archive, 1999.

[24] M. Harries, Electricity, Aug. 2019.

[25] M. GIS, Police Stop Data, 2017.

[26] M. J, Chess Game Dataset, 2017.

[27] W. Liu, Adult income dataset, 2016.