Neural Ensemble Concept Drift Aware Data Stream Classification

Radin Hamidi, Maryam Amir Haeri

Neural Ensemble Concept Drift Aware Data Stream Classification

Radin Hamidi Rad[a*], Maryam Amir Haeri[b]

[a] *Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto, Canada*

[b] *Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran*

Keywords

Big Data,
Neural Network,
Data Stream,
Classification,
Ensemble

Abstract

Data stream mining becomes more and more critical nowadays. These algorithms can be used in various scenarios in the industry and daily use. Dynamic networks, user-generated data, and so many other causes are the main sources of data streams. Hoeffding Trees as a data stream classification approach have shown a good performance in dealing with the data stream related challenges. This method makes the any-time prediction possible. Concept drift is a well-known problem coexist with data streams and there are plenty of approaches available today that tried to handle this problem. In this article, we introduced a new algorithm that uses VFDT and neural network to handle concept drift phenomenon in an acceptable way. This algorithm also benefits from fast startup ability which helps systems to be able to predict faster than other algorithms at the beginning of data stream arrival or concept drift event. We also have shown that our approach will overperform other controversial approaches for the classification task.

1. Introduction

There are many machine learning methods applied to regression and classification problems in the literature. But there are issues that the existing methods are not equipped to deal with. Hoeffding Trees as a data stream classification approach have shown a good performance in dealing with the data stream related challenges. This method makes the any-time prediction possible. VFDTs are one of the most famous approaches in dealing with high-speed data streams with concept drift feature. VFDTs overcome other methods problems such as easy to be trapped in a local minimum (for ANN) and hard to decide proper kernel parameters and penalty (for SVR) [1], in contrast, they suffer from high latency in learning progress due to the existence of equally discriminative attributes. Load forecasting in power electricity market, network monitoring, and fault detection in industrial equipment are most common use cases of concept-drift detection algorithms [2-8]. In this article, we proposed a new algorithm that tries to use VFDT as judgment system and leverage neural network system to determine concept drift occurrence and alarm the main judgment system. The main contribution of this paper is the following:

- We proposed a novel algorithm for data stream classification which is adaptable with concept drifts and can be used for both concept drift detection and data stream forecasting with the chance of having concept drifts.
- We discussed mathematical aspects of our proposed algorithm by details and proved that under certain conditions our approach works efficiently and guarantee to yields desirable output.
- We show that our new method has low computational complexity and can improve the prediction accuracy and over-perform some of the latest data stream mining systems.

First, we discuss the related works in Section 2, then in Section 3 we introduce proposed method. Experimental results and discussion of implemented algorithms are available in Section 4. At the end, Section 5 utters the conclusion and future work.

2. Related Works

Decision and Regression Trees [9] are one of the most used topics among methods in machine learning algorithms [10] due to the learning speeds, accuracy, and overall performance [11]. Some critical features like easy knowledge visualization, robustness and low chance of over-fit made this method preferable respect to other learning algorithms. There are lots of implementations of this algorithm for the different case of uses such as classification on batch data, classification on stream data, regression on batch data and finally regression on a stream of data. In this paper, we propose a new method for regression trees which operate on stream data. Regression on data streams faces challenges including concept drift and inability to access to

* Corresponding Author
E-mail addresses: radin@ryerson.ca, haeri@aut.ac.ir

Received: 29 April 2020; Accepted: 26 May 2020

all samples at any time [12]. Aforementioned problems do not allow the learner to obtain overall vision on the whole set of data that cause lack of certainty for choosing the best feature to split data respect to. There are a couple of approaches for addressing the above issues which we will study briefly in the following paragraphs.

Very Fast Decision Tree: VFDT has been announced by Domingos et al in 2000. This algorithm has been specially designed for data streams and uses Hoeffding inequality to guarantee the right moment for turning a leaf to a splitting node. Consider a real-valued random variable r whose range is R (e.g., for a probability the range is one, and for an information gain the range is $R=\log c$, where c is the number of classes). Suppose we have made n independent observations of this variable and computed their mean $\bar{r}$. The Hoeffding bound states that, with the probability of $1-\delta$, the true mean of the variable is at least $\bar{r}-\varepsilon$, Eq. (1), where:

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

The Hoeffding bound has a very attractive property that it is independent of the probability distribution generating the observations [13]. The price of this generality is that the bound is more conservative than distribution-dependent ones (i.e., it will take more observations to reach the same δ and ε) in return we will be able to determine the suitable amount of samples needed to start splitting the tree and making new decisions.

Random Forest of Very Fast Decision Trees: Random forests can solve data variety and fit better on samples. This attitude let to implementing RFVDFTs which are a subclass of the Decision Trees [14]. This fact that ensemble methods are always outperformed single algorithms is proved in Hensen & Salamon 1990 [15]. They have uttered two condition for weak learners which guarantees this notion. A weak learner must be:

1. *Accurate: An accurate classifier is one that has an error rate of better than random guessing on new data values.*
2. *Diverse: Two classifiers are diverse if they make a different error on new data points*

Support Vector Machine Approaches: Using a SVM as a solution for data stream mining is discussed in various articles including [16-18]. Despite different proposed solutions to achieving an acceptable result, it can be seen that all of the approaches need kernel parameters to be set and other settings get done before starting. These pre-processing configurations make SVM/SVR approaches vulnerable to concept drifts.

Neural Network Approaches: Neural Networks is one of the favorite solutions for sequential data mining. ANN has various applications such as noise reduction [19] and control chart pattern recognition [20, 21] which indicate it performs well in real engineering problems. Recurrent neural networks and LSTMs are common approaches for sequential data mining. However, data streams are slightly different in their definition from sequential data, but these neural structures sometimes will be used as a solution [22, 23] The most important problem with these solutions is their famous problem with their local minima. That behavior makes them not suitable for cases with concept drifts or situations that we know our problem is non-convex.

In the rest of the paper, we discuss about our proposed approach and report the results of implementing random forest of VFDT and then introduce a brand new method for combining trees which not only improves the overall accuracy but also has significant effect on converge speed of single VFDT that can be an alternative solution beside Hoeffding trees with options that has its own difficulties and weak points.

3. Ensemble Neural Data Stream Classification Algorithm

In this section, we will introduce our proposed algorithm by detail. As we discussed in last section, ensemble methods can over perform ordinary algorithms under certain conditions. Thus, if we try to use accurate weak learners in a way that guarantee they cover the diversity, we can achieve better performance theoretically. However, these conditions need specific calculation to determine number of weak learners and size of feature vector. The mathematical proof will be discussed in the following. First, we represent procedure of algorithm in subsection 3.1 and then we discuss about the mathematical aspect of our algorithm in subsection 3.2.

3.1. Procedure

This section provides a step by step introduction through the algorithm. First, we generate a set of weak learners. All these weak learners are Hoeffding trees, that have access to the square root number of original feature vector size. Then we create a single complete VFDT and call it the main tree. Now that we have all the required learners, we put them together as a forest. Therefore, this forest contains a single complete tree and a bunch of weak learners. Experiments have shown that the main tree performs better than weak learners in normal conditions due to the fact they populate more complete samples and can decide better. However, weak learners can perform better when a concept drift phenomenon occurred. That is because weak learners with less feature contained are less vulnerable to the concept drift effects. As a result, weak learners will show a robust performance when it comes to the concept drift problem. A symbolic representation of the proposed algorithm can be seen in Figure 1.

Now that we have both the concept drift aware algorithm and main VFDT tree, we must propose an approach to combine results from these two sources. We have proposed a neural network solution for the mentioned problem. There are several architectures available for our problem. We could use multi-layer perceptron, deep neural networks, and recurrent neural networks. By implementing and testing all the possible approaches, we finally decide to use LSTM neural networks as the accepted architecture for our algorithm. You can see the neural network's architecture in Figure 2.

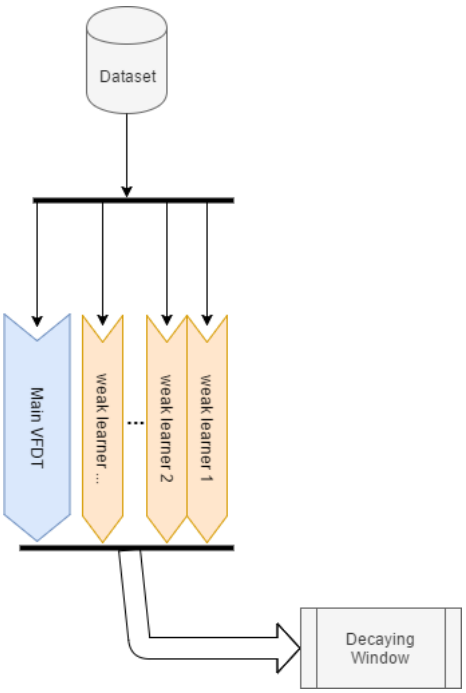

Figure 1. Symbolic presentation of proposed algorithm. Note the created ensemble consisting of a main tree and a bunch of weak learners.

As it is shown, the result of each weak learner tree and the result of the main tree is gathered and entered as an input vector to the network. The predicted class is the neural network's output. Also, we are using decaying windows that only saves last *n* samples and LSTM network will be trained for a specific number of epochs on that frame to adapt properly.

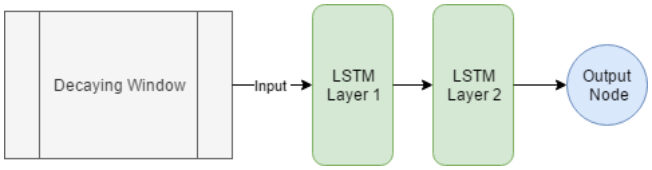

Figure 2. Neural network architecture for proposed algorithm.

When a concept drift has occurred, the accuracy of the LSTM system will decrease dramatically and can be used as an alarm for concept drift detection. Also, weak learners in the algorithm help system to get back on the track soon after a concept drift arrived.

3.2. Efficient Number of Weak Learners

Replacing Single VFDT with an ensemble version of it can cause more use of both memory and computation resource of the machine which we name it computation cost here. To minimize the computation cost, it's needed to determine the efficient state for the proposed algorithm. We will show that under certain constraints for our random forest creation we can guarantee that our random forest members contain all the features at least once and therefore, our desired phenomenon of concept drift detection by weak learners will show up consequently. Here are defined constraints of our problem:

1. For each attribute, there must be at least on weak learner among the group of weak learners that has it inside its bag of features.
2. For convenience, we assumed that the number of features for each weak learner derive from thumb rule in Eq. (2).

$$f_{wl} = \sqrt{f_{total}} \tag{2}$$

Where f_{wl} is the size of feature vectors for each weak learner and f_{total} is the size of a complete feature vector of our system.

Therefore, if we shoe probability of not being chosen by none of the weak learners by *P*, then we can calculate *P* as stated in Eq. (3):

$$P = \binom{\sqrt{n}}{0} \left(\frac{1}{n}\right)^0 \left(\frac{n-1}{n}\right)^{\sqrt{n}} \tag{3}$$

Where *n* is size of feature vector for data stream.

Above equation can be simplified and be written in Eq. (4).

$$p = \left(\frac{n-1}{n}\right)^{\sqrt{n}} \tag{4}$$

Now that we calculated the probability of not being chosen at all, we can calculate the probability of being chosen at least once by simply subtracting it from 1. So the probability of a feature to be chosen at least once, Eq. (5) will be:

$$q = 1 - p \tag{5}$$

Where q is representation of probability of a feature to be chosen at least once. Since procedure of random forest creation consist of repetitive selection of features for each weak learner, we must consider number of total selection times in our calculation. Let take *m* as number of total weak learners and *n* size of the original feature vector, then the probability of not being shown up ever during whole process of feature selection Eq. (6) would be:

$$\binom{\sqrt{m}}{0} (q)^0 (p)^m = p^m \tag{6}$$

$$p_{not\text{-}observed} = \left(\frac{n-1}{n}\right)^{m\sqrt{n}} \tag{7}$$

Therefore, as the Eq. (7) utters, the probability of not being chosen even once in *m* times random with replacement selection of features for weak learners is equal to $\left(\frac{n-1}{n}\right)^{m\sqrt{n}}$. Now we can select the desired size of our random forest according to *n* which is the size of the original feature vector and calculated probability of presentation of each feature at least once. For better intuition of the current equation, it's better to subtract the probability from one to obtain the probability of being seen at least once since it's our final desired value. The final form of the desired variable for our work is shown in Eq. (8).

$$P_{confidence}=1-(\frac{n-1}{n})^{m\sqrt{n}} \quad (8)$$

It is clear that by increasing *m* which is the number of weak learners, the confidence will approach to its maximum value that is one and by decreasing the size of the feature vector our confidence will grow faster that means needs less number of weak learners so dimensionality reduction and removing redundant features may cause better performance significantly.

4. Experimental Results

The experimental result that is achieved from running the proposed algorithm is brought in this section.

4.1. Dataset

Due to the fact that best illustration will be achieved when we check any-time accuracy and converge speed on the same datasets that are used on previous methods, we have tested our algorithm on a famous waveform dataset from the UCI Machine Learning Repository (Frank & Asuncion, 2010) [24]. This dataset is a synthetical dataset which means is generated by data generator. Thus, the dataset characteristics are defined by the user (us). For testing the algorithm, we used a waveform dataset with 21 features and 3 classes. Also, all the results are reported after 30 runs to avoiding any unstable result.

4.2. Classification Results and Discussion

In this section, we will discuss the results of the proposed algorithm and will analyze its performance. The result of repeated runs on two different configurations of waveform dataset for three different methods including the proposed model can be found in Table 1. These results are reported after running algorithms for 30 times repeatedly.

Table 1. Accuracy of different methods

Data Set	Method	Accuracy	StDev
Waveform 1	VFDT	70.2%	0.0
	Random Forest	71.3%	1.77
	*Proposed Algorithm	**74.68%**	1.41
Waveform 2	VFDT	72.6%	0.0
	Random Forest	73.5%	1.82
	*Proposed Algorithm	**75.76%**	1.50

Time complexity is one of the most prominent issues that must be considered in data stream mining. In Table 2, consumed time for three algorithms are shown. As can be seen, our proposed algorithm does not add much time load on the system in comparison to normal VDFT. Thus, we can safely use this new method with no concern about input overload. We used a processing system with Intel Core i7 @2.3 GHz CPU, 16Gb DDR4 RAM and Nvidia GeForce 1060 GPU for recording mentioned measurements.

Table 2. Consumed time for each method

Data Set	Method	Time (μs)
Waveform 1	VFDT	235
	Random Forest	332
	*Proposed Algorithm	392
Waveform 2	VFDT	241
	Random Forest	394
	*Proposed Algorithm	398

As can be seen in Figure 3, proposed algorithm over-performs both random forest and VFDT. The result is brought from a different configuration for better inference.

4.3. Concept Drift

This section represents experimental results of concept drift detection solution that were introduced in subsection 3.1. For a better comparison over all methods, we compared the result of concept drift detection and recovery after that on three different approaches: CVFDT, EDDM-VFDT, and proposed algorithm. As it is shown in Figure 4, we can see that the proposed algorithm can better handle the concept drift and lose less accuracy. That is because, when the main tree falls in wrong track of stream and cannot follow that correctly, the weak learners will start alarming and switch output values from a single tree to the ensemble mode.

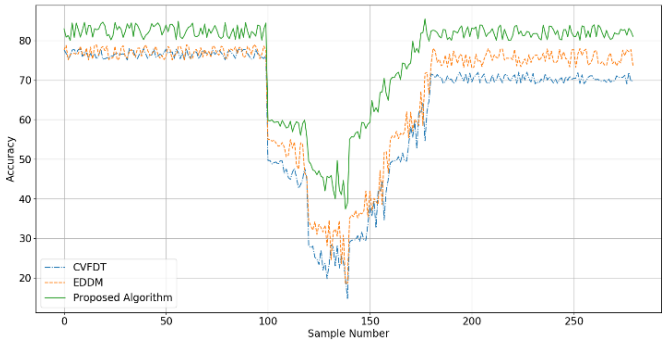

Figure 4. Concept drift detection and recovery comparison

5. Conclusion

In this paper, we tried to solve the concept drift problem by using a neural network as a detection system and an ensemble approach for the structure of the main algorithm. Our experimental results show that adding weak learners to the algorithm will cause the algorithm to be aware of concept drift. This effect is because of increased sensitivity of the algorithm to the incoming data stream by adding weak learners which do not have vision on the whole variable of a dataset and grow their own tree easier respect to the complete tree. The proposed neural network will use weak-learner's judgment as input and tries to learn the importance and real influence of each weak learner by modifying the weights. The main contribution of this method lies in proper concept drift detection and high accuracy classification. The

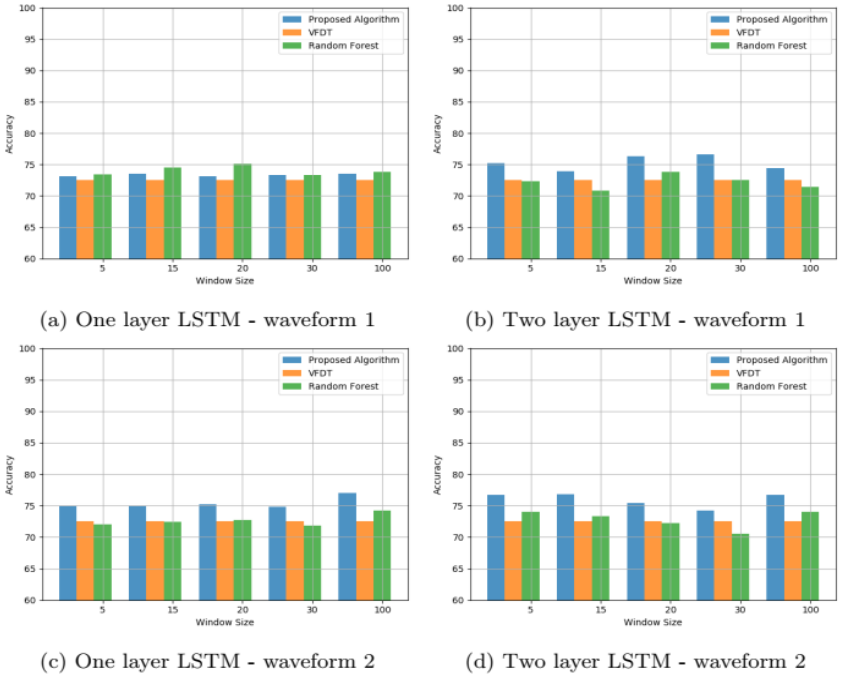

Figure 3. Comparison of Accuracy between single Hoeffding tree, proposed model, and random forest for different configurations.

experimental results showed that our algorithm generally achieved good performance and can outperform other controversial approaches. For future work, we can upgrade the current neural network for achieving higher accuracy.

References

[1] X. Wu, J. He, T. Yip, and N. Lu, "A two-stage random forest method for short-term load forecasting," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1-5.

[2] H. Panamtash, Q. Zhou, T. Hong, Z. Qu, and K. O. Davis, "A copula-based Bayesian method for probabilistic solar power forecasting," *Solar Energy*, vol. 196, pp. 336-345, 2020.

[3] R. F. Widjaja, H. Panamtash, Q. Zhou, and D. Li, "Solar Power Forecasting with Model Selection Analysis," in *2018 Clemson University Power Systems Conference (PSC)*, 2018, pp. 1-6.

[4] A. Alzghoul, M. Löfstrand, and B. Backe, "Data stream forecasting for system fault prediction," *Computers & industrial engineering*, vol. 62, pp. 972-978, 2012.

[5] F. Rezaeimehr, P. Moradi, S. Ahmadian, N. N. Qader, and M. Jalili, "TCARS: time-and community-aware recommendation system," *Future Generation Computer Systems*, vol. 78, pp. 419-429, 2018.

[6] K. Grolinger, A. L'Heureux, M. A. Capretz, and L. Seewald, "Energy forecasting for event venues: Big data and prediction accuracy," *Energy and Buildings*, vol. 112, pp. 222-233, 2016.

[7] H. Panamtash and Q. Zhou, "Coherent probabilistic solar power forecasting," in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2018, pp. 1-6.

[8] N. S. Gilani, M. T. Bina, F. Rahmani, and M. H. Imani, "Data-mining for Fault-Zone Detection of Distance Relay in FACTS-Based Transmission," in *2020 IEEE Texas Power and Energy Conference (TPEC)*, 2020, pp. 1-6.

[9] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*: CRC press, 1984.

[10] L. Ali, S. U. Khan, N. A. Golilarz, I. Yakubu, I. Qasim, A. Noor, *et al.*, "A Feature-Driven Decision Support System for Heart Failure Prediction Based on Statistical Model and Gaussian Naive Bayes," *Computational and Mathematical Methods in Medicine*, vol. 2019, 2019.

[11] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, pp. 660-674, 1991.

[12] M. Elyasi, M. Meybodi, A. Rezvanian, and M. A. Haeri, "A fast algorithm for overlapping community detection," in *2016 Eighth International conference on information and knowledge technology (IKT)*, 2016, pp. 221-226.

[13] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 71-80.

[14] D. Marron, A. Bifet, and G. D. F. Morales, "Random Forests of Very Fast Decision Trees on GPU for Mining Evolving Big Data Streams," in *ECAI*, 2014, pp. 615-620.

[15] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, pp. 993-1001, 1990.

[16] J. Liu and E. Zio, "A SVR-based ensemble approach for drifting data streams with recurring patterns," *Applied Soft Computing*, vol. 47, pp. 553-564, 2016.

[17] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE Transactions on Neural Networks*, vol. 22, pp. 1901-1914, 2011.

[18] L. Ali, A. Niamat, J. A. Khan, N. A. Golilarz, X. Xingzhong, A. Noor, *et al.*, "An optimized stacked support vector machines based expert system for the effective prediction of heart failure," *IEEE Access*, vol. 7, pp. 54007-54014, 2019.

[19] N. A. Golilarz and H. Demirel, "Thresholding neural network (TNN) based noise reduction with a new improved thresholding function," *Computational Research Progress in Applied Science and Engineering*, vol. 3, pp. 81-84, 2017.

[20] A. Addeh, A. Khormali, and N. A. Golilarz, "Control chart pattern recognition using RBF neural network with new training algorithm and practical features," *ISA transactions*, vol. 79, pp. 202-216, 2018.

[21] N. A. Golilarz, A. Addeh, H. Gao, L. Ali, A. M. Roshandeh, H. M. Munir, *et al.*, "A new automatic method for control chart patterns recognition based on ConvNet and Harris Hawks meta heuristic optimization algorithm," *IEEE Access*, vol. 7, pp. 149398-149405, 2019.

[22] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural network for semi-supervised data stream classification," in *The 2010 international joint conference on neural networks (IJCNN)*, 2010, pp. 1-8.

[23] Y. Cui, S. Ahmad, and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," *Neural computation*, vol. 28, pp. 2474-2504, 2016.

[24] D. Dua and E. Karra Taniskidou, "UCI Machine Learning Repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California," *School of Information and Computer Science*, 2017.