Online Data Steam Classification

Emre Doğan
Bilkent University
Ankara, Turkey
emre.dogan@bilkent.edu.tr

1 INTRODUCTION

This document is the technical report of CS533 Assignment #3. In Section 2, the related work on the data streams, data stream classification and the challenges faced in this area are discussed. In Section 3, the general concepts of the methodology are defined. In Section 4, the experimental setups and the corresponding results are shared. Finally, Section 5 gives a brief conclusion about this study.

2 RELATED WORK

Related work section is organized as follows: a brief background on data streams and its classifications are given in Section 2.1. Then, prevalent challenges faced in data stream classification task are discussed in Section 2.2. Finally, a short survey of the literature is completed in Section 2.3.

2.1 Data Streams

Data streams play an important role in many real life applications such as telecommunications, social networks, algorithmic trading, supply chain optimizations,network monitoring, predictive maintenance. Since stream processing tasks have been suffering from the challenges due to the resource-constrained factors *(memory, power consumption, time)*, advances in hardware technology over the last decade have led to the increasing popularity of data streams and its applications [1].

One of the most popular applications regarding the data streams is the classification task. The data instances, consisting of a stream of data, are labelled to a data class. The main purpose is to predict the label of the new-coming instances successfully.

2.2 Challenges in Data Stream Classification

Concept Drift:

Concept drift can be defined as the deviation of the relationship between input and output over a period of time. Zliobaite et al. [13] defines it in a formal manner in the following way:

"In most challenging data analysis applications, data evolve over time and must be analyzed in near real time. Patterns and relations in such data often evolve over time, thus, models built for analyzing such data quickly become obsolete over time. In machine learning and data mining this phenomenon is referred to as concept drift."

In general, predictive models are represented as a static mapping function. For input as X and output as Y, this relation is represented as :

$$Y = f(X)$$

In this scenario, the relation between X and Y is assumed to be static and never changes, even when new data instances arrive. For some classification tasks, this representation might be suitable. But for the data stream classification task, the relation between input and output does not have to remain stable. The properties of data records that the predictive model is based on may change over the passing time. This phenomenon is called as *concept drift*.

Predictive models dealing with data streams should take some actions in order to avoid resulting with a static mapping function such that the mapping does not change even if the future data instances show different characteristics than the historical training data.

Other Challenges Faced in Data Stream Classification:

- Due to the massive amount of data, it might be impossible to process the entire stream in a single processor. In such cases, distributed data stream applications should be used.
- It is a challenging task to deal with imbalanced stream datasets. The stream instances with a few number of occurrences become hard to predict. This fact cannot be ignored since predicting such instances are often more important than the frequent ones*(e.g. Credit card fraud detection)*.

2.3 Data Stream Classification Studies

In the literature, there are many studies proposing distinct methods to perform data stream classification.

Classical Approaches:

Domingos and Hulten [3] proposed a modified decision tree approach designed for data stream environments. Instead of scanning the whole dataset many times in the training phase, the Hoeffding bound is used to find the ideal splitting attribute. The model works in only one single-pass and suitable to use in a real life setup. The only problem with this approach is that it cannot handle the concept drift problem.

To solve the concept drift problem in Hoeffding Tree approach, Hulten et al.[5] proposed a new model, *CVFDT*. The tree stays stable and creates some alternative subtrees at the same time. When an old subtree becomes questionable and the new one beats it in terms of performance, then the new subtree is replaced with the old one. CVDFT scans the nodes regularly to detect the concept drift and overcomes this issue with the help of new subtrees.

Seidl et al. [11] came up with a hierarchical clustering algorithm based on Gaussian mixture model. In multi-labeled classification, this model creates a separate Bayesian tree for each class which is expensive. Kranen et al. [6] solves this problem by combining all classes into a single Gaussian Tree.

Leite et al. [7] proposed a granular neural network (eGNN) approach to classify data streams. The major problem of this approach is the long training time due to computational complexity of neural networks. For this reason, it is not feasible to apply this model on massive datasets.

Algorithm	Single-Pass	Robustness to Noise	Concept Drift Handling	Real-Time Response	Ensemble
VDFT [3]	✓			✓	
CVFDT [5]	✓		✓	✓	
Bayes Tree [11]	✓	✓	✓	✓	
MC-tree [6]	✓	✓	✓	✓	
e-GNN [7]	✓		✓		
Online Bagging & Boosting [10]	✓			✓	✓
Aggregated Ensemble [12]	✓		✓	✓	✓
HEFT [9]	✓		✓	✓	✓
OVA [4]	✓		✓	✓	✓

Table 1: Comparison of Different Data Stream Classification Studies

Ensemble Approaches:

In a general manner, ensemble models increase the classification success. Therefore, several ensemble methods have been applied in the data stream classification.

Oza et al. [10] proposed one of the earliest ensemble methods in stream classification. They adapted classical bagging and boosting approaches into the domain. Zhang et al.[12] suggested a robust classifier method, *Aggregated Ensemble*, in order to train the several learning models with noisy data. Nguyen et al.[9] illustrated that a small number of data attributes are significant in the learning phase and the rest might be ignored to speed up the process. Their approach is important in terms of learning from high-dimensional data streams. Hashemi et al. [4] proposed the model, *OVA, Adapted One-vs-All Decision Trees* in which a set of CVFDT classifiers are trained. Each CVFDT model is used to classify between one specific class and all the rest. When a new data instance comes, all these classifiers are run and the class whose classifier gives the highest confidence is returned.

A comparison of the related work is given in Table 1. While creating this table, the survey of Nguyen et al. [8] was used.

3 METHODOLOGY

In order to implement Hoeffding Tree and Naive Bayes classifiers for online data stream classification task, the framework *MOA (Massive Online Analysis)* is used. The following steps are followed during this study:

- First, a dataset is generated by using RandomRBFGenerator method and written to the file *RBFdataset.arff*.
- Then, Hoeffding Tree and Naive Bayes Classifers are created by using the implementations available in MOA. When a new data instance arrives, the model is tested with this instance and then trained with it.
- After the base case scenario is implemented (10 features, 2 classes), some controlled experiments are made to observe the effect of number of features and classes.

3.1 Massive Online Analysis (MOA)

Massive Online Analysis (MOA) is a software environment, *developed at the University of Waikato, New Zealand*, in order to implement algorithms and running experiments for online learning from evolving data streams [2]. It includes a set of classification and clustering algorithms and also some evaluation tools for these algorithms.

MOA supports bi-directional interaction with WEKA Tool (Waikato Environment for Knowledge Analysis), a machine learning framework with Java API and GUI editions. As the data stream environments have some differences compared with classical batch learning ones, MOA provides support for data stream environments differing from WEKA.

3.2 Synthetic Dataset Generation

In order to generate a dataset with 10,000 instances consisting of 10 features and 2 class labes, RandomRBFGenerator method from MOA is used. Then, these instances are written to a file called *RBFdataset.arff*.

3.3 Classifiers

3.3.1 Hoeffding Tree: Hoeffding Tree is a method proposed by Hulten et al.[5] in which an incremental version of decision tree is used for data streams by assuming that the data distribution does not change over the passing time.

3.3.2 Naïve Bayes: Despite of its being an old-fashioned algorithm, Naive Bayes is one of the most reliable and simple classifiers available in the literature. It has a large number of applications especially in natural language proessing and information retrieval tasks. Naive Bayes classifiers simply calculate the probability of a data instance to belong to a certain category, based on prior knowledge coming from calculations.

3.4 Training and Test Methodology

In data stream mining tasks, the most popular evaluation method is the *prequential* or *interleaved-test-then-train* evolution. This method is based on using each data instance first to test the model, and then to train the model. It has two types of evaluation: *classical approach* and *window based*. Classical approach measures the accuracy since the start of the evaluation where the window based approach measures it only within the current sliding window of recent data instances.

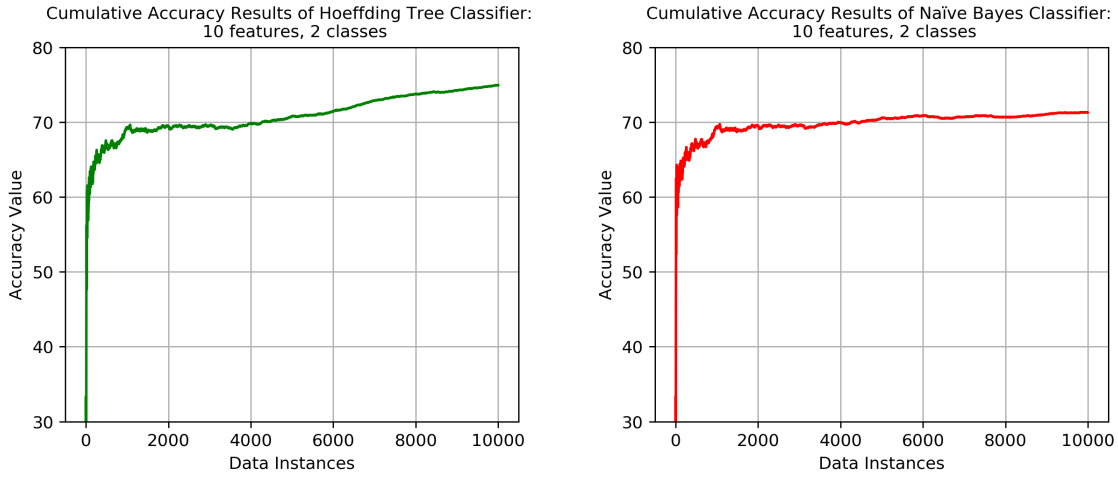

Figure 1: Cumulative Accuracy Results of HT and NB classifiers with 10 features and 2 classes

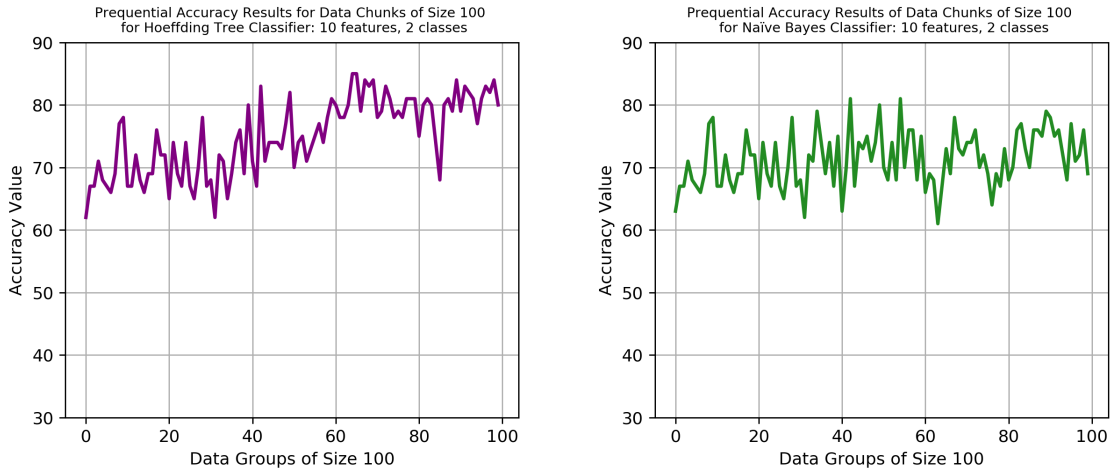

Figure 2: Prequential Accuracy Results (100 chunks consisting of 100 data instances each) of HT and NB classifiers with 10 features and 2 classes

4 RESULTS

4.1 Base Case Scenario

As the base case scenario, the number of features is 10 and the number of classes is 2. By using these parameters, the accuracy results for both models (Hoeffding Tree and Naïve Bayes Classifiers) are calculated and illustrated in Table 2.

In order to visualize the change in the accuracy value as more data instances are used in the training phase, 2 different methods are used: plotting *the cumulative accuracy* and *the prequential accuracy* by dividing the dataset into 100 chunks with 100 instances.

In order to plot the cumulative accuracy, the number of correctly predicted instances is divided to the total number of instances examined until that point. In the earlier steps, the accuracy value is low, as expected due to the small number of training instances. However, the model starts to make more accurate predictions as it is trained by more and more data instances. The plottings of cumulative accuracy for Hoeffding Tree and Naïve Bayes Classifiers are available in Figure 1.

Although the cumulative accuracy plot gives an idea for the evaluation of the model prediction success, it does not show the correct accuracy values of temporal regions. For example, consider the accuracy value calculated in the 1000^{th} data instance.

Model	Accuracy Value
Hoeffding Tree	0.749
Naïve Bayes	0.713

Table 2: Cumulative Accuracy Values of HT and NB Models for the Last Data Instance

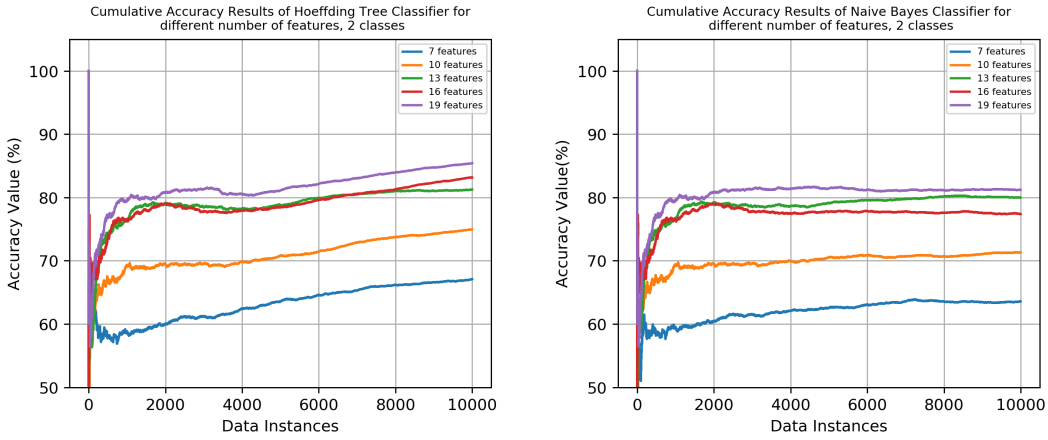

Figure 3: Cumulative Accuracy Results of HT and NB classifiers with different number of features and 2 classes

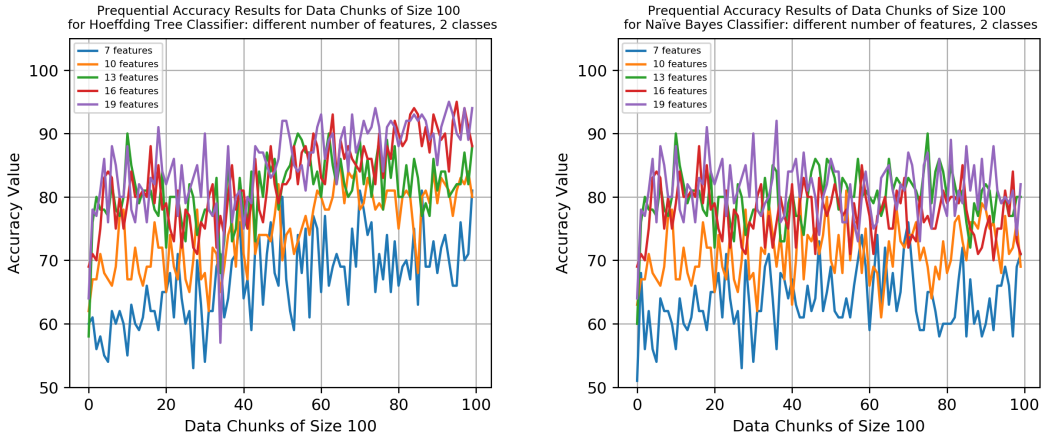

Figure 4: Prequential Accuracy Results (100 chunks consisting of 100 data instances each) of HT and NB classifiers with different number of features and 2 classes

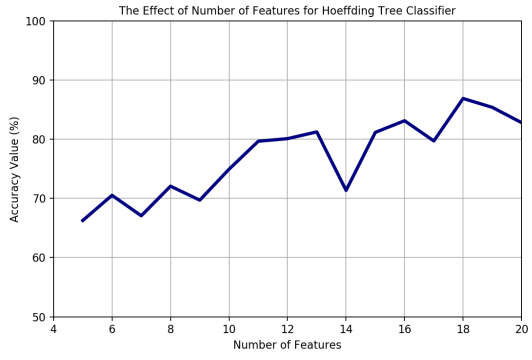

Figure 5: The Effect of Number of Features on the Final Accuracy Values for Hoeffding Tree Classifier

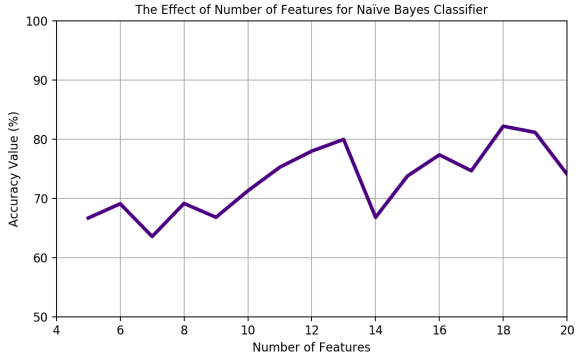

Figure 6: The Effect of Number of Features on the Final Accuracy Values for Naive Bayes Classifier

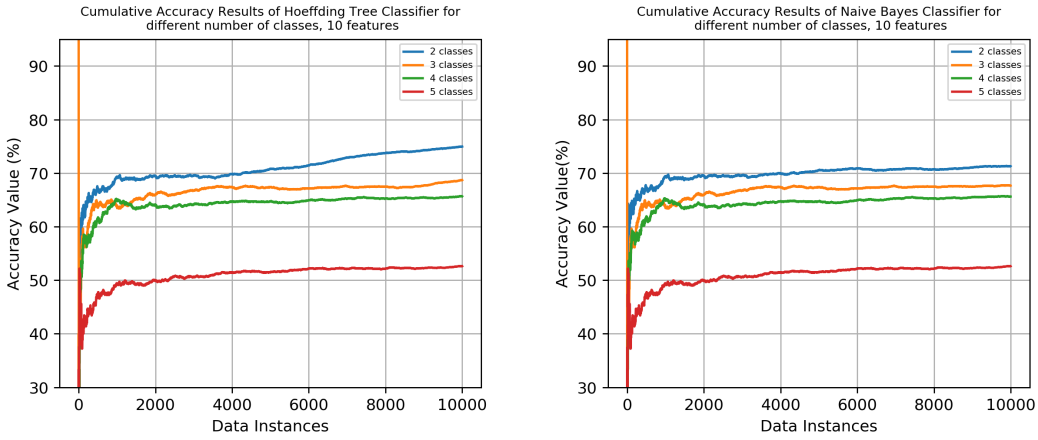

Figure 7: Cumulative Accuracy Results of HT and NB classifiers with different number of classes and 10 features

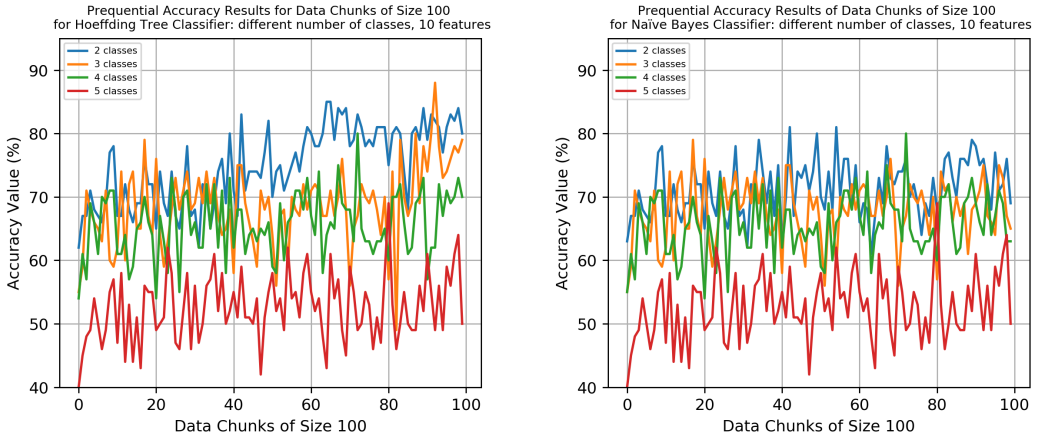

Figure 8: Prequential Accuracy Results (100 chunks consisting of 100 data instances each) of HT and NB classifiers with different number of classes and 10 features

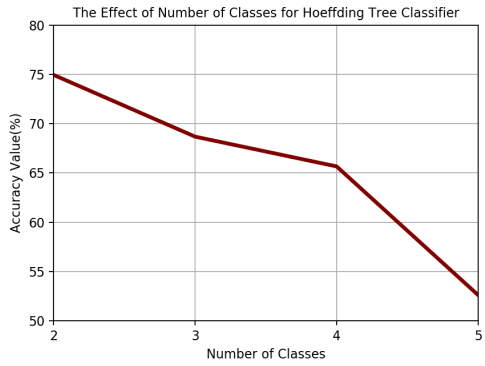

Figure 9: The Effect of Number of Classes on the Accuracy Values for Hoeffding Tree Classifier

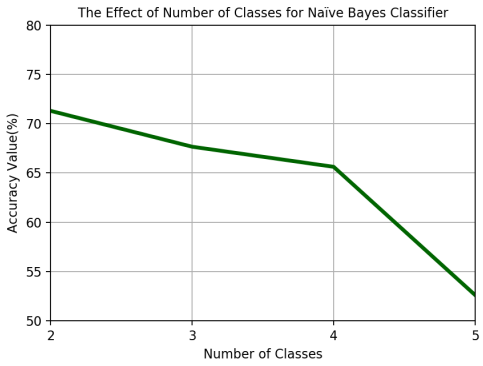

Figure 10: The Effect of Number of Classes on the Accuracy Values for Naive Bayes Classifier

This value would be seriously affected by the first accuracy values (e.g. first 100 instances) which are too low due to the lack of training. To mitigate this problem, the whole dataset of 10000 samples is split into 100 chunks consisting of 100 data instances each. Then the average accuracy value for each chunk is calculated so that each chunk can be evaluated independently. The graphs of prequential accuracy values of 100 data chunks for both classifiers are given in Figure 2.

When both Figure 1 and 2 are investigated, it is seen that prequential accuracy values of data chunks are more meaningful in terms of evaluating the model. For instance, after the 60^{th} chunk, the accuracy value is calculated more around the value , 0.80, although it never passes beyond the value of 0.75 in the cumulative accuracy.

In general, the accuracy value gets better as the model is trained by more data instances. HT classifier seems to have an increasing pattern in terms of accuracy. If there were more data instances to train the model, we could result with a higher accuracy value. On the contrary, the cumulative accuracy of Naïve Bayes Classifier seems to saturate around 0.71 and not increase in a significant way.

4.2 Effect of the Number of Features

In order to observe the effect of the number of features in the data stream instances, a controlled experiment setup is used. The number of classes (2) and the number of data instances (10000) are kept constant so that the effect of number of attributes can be observed easily. The accuracy results are calculated for different numbers of features within the range [5,20].

The cumulative and prequential accuracy plots of HT and NB Classifiers for different number of features are given in Figure 3 and Figure 4, respectively.

Also, the final accuracy values are given in Figure 5 for HT classifier and in Figure 6 for NB classifier.

4.3 Effect of the Number of Classes

In order to observe the effect of the number of classes in the data stream classification task, a controlled experiment setup similar to the one mentioned in Section 4.2 is preferred. Different number of classes (2, 3, 4, 5) are used in order to achieve accuracy results while other parameters are kept constant (number of features: 10, number of data instances: 10000).

The cumulative and prequential accuracy plots of HT and NB Classifiers for different number of classes are given in Figure 7 and Figure 8, respectively.

Also, the final accuracy values are given in Figure 9 for Hoeffding Tree Classifier and in Figure 10 for Naive Bayes Classifier.

When both plots are investigated, it is observed that accuracy values drop significantly when the number of classes is increased. In order to prevent the drop in accuracy, the number of features and the number of data instances could be increased so that the model can be trained in a better way.

5 CONCLUSION

In this assignment, Hoeffding Tree and Naive Bayes Classifiers are implemented to classify data streams. First, the base results with 10 features and 2 classes are achieved. Then, some experiments are made to find the effect of number of features and classes on the model success.

According to my findings, the following conclusions are drawn:

- Hoeffding Tree can achieve better results than the Naive Bayes approach when we have a large number of data instances.
- When the prequential accuracy plots are investigated, it is observed that the accuracy value increases significantly after a few data chunks of 100 instances. It shows that we need at least a few hundred data instances in order to make reasonable predictions.
- Increasing the number of features increases the performance of the model to a certain extent. After some point, it saturates and the model begins to overfit.
- When the number of class increases, the accuracy value drops. For a dataset with a larger number of classes, either a larger dataset or data instances with more features should be considered.

REFERENCES

[1] Charu C. Aggarwal. 2014. A Survey of Stream Classification Algorithms. In *Data Classification: Algorithms and Applications*.

[2] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. 2010. Moa: Massive online analysis. *Journal of Machine Learning Research* 11, May (2010), 1601–1604.

[3] Pedro Domingos and Geoff Hulten. 2000. Mining high-speed data streams. In *Kdd*, Vol. 2. 4.

[4] Sattar Hashemi, Ying Yang, Zahra Mirzamomen, and Mohammadreza Kangavari. 2008. Adapted one-versus-all decision trees for data stream classification. *IEEE Transactions on Knowledge and Data Engineering* 21, 5 (2008), 624–637.

[5] Geoff Hulten, Laurie Spencer, and Pedro Domingos. 2001. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 97–106.

[6] Philipp Kranen, Stephan Günnemann, Sergej Fries, and Thomas Seidl. 2010. MC-tree: Improving bayesian anytime classification. In *International Conference on Scientific and Statistical Database Management*. Springer, 252–269.

[7] Daniel F Leite, Pyramo Costa, and Fernando Gomide. 2009. Evolving granular classification neural networks. In *2009 International Joint Conference on Neural Networks*. IEEE, 1736–1743.

[8] Hai-Long Nguyen, Yew-Kwong Woon, and Wee-Keong Ng. 2015. A survey on data stream clustering and classification. *Knowledge and information systems* 45, 3 (2015), 535–569.

[9] Hai-Long Nguyen, Yew-Kwong Woon, Wee-Keong Ng, and Li Wan. 2012. Heterogeneous ensemble for feature drifts in data streams. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 1–12.

[10] Nikunj C Oza. 2005. Online bagging and boosting. In *2005 IEEE international conference on systems, man and cybernetics*, Vol. 3. Ieee, 2340–2345.

[11] Thomas Seidl, Ira Assent, Philipp Kranen, Ralph Krieger, and Jennifer Herrmann. 2009. Indexing density models for incremental learning and anytime classification on data streams. In *Proceedings of the 12th international conference on extending database technology: advances in database technology*. ACM, 311–322.

[12] Peng Zhang, Xingquan Zhu, Yong Shi, Li Guo, and Xindong Wu. 2011. Robust ensemble learning for mining noisy data streams. *Decision Support Systems* 50, 2 (2011), 469–479.

[13] Indrė Žliobaitė, Mykola Pechenizkiy, and Joao Gama. 2016. An overview of concept drift applications. In *Big data analysis: new algorithms for a new society*. Springer, 91–114.