Data Stream Drift Adaptation and Adversarial Attack Detection using Ensemble Learning Algorithms

Efe Tarhan
Electrical and Electronics Engineering
22002840

I. Introduction

Online machine learning is a technique that handles the data that does not exist as a whole or in batches but arrives one by one to the processing pipeline. The methods of this field are used frequently for e-commerce, healthcare, autonomous driving systems and content streaming. This work delves into the topic of data stream mining with a specific focus on the concepts of concept drift and adversarial attacks. In Section II, theoretical concepts of the data stream mining are explained by referring to the work related to the field. Following that performance of an ensemble classifier of the scikit-multiflow library and a custom ensemble model are tested and compared with each other for their performance of handling valid concept drifts in Section III. This part covers the part D.3.b of the assignment. Following that the performance of these models are tested against adversarial attacks and a valid method for detecting the attack is proposed and tested in Section IV. Final remarks on the assignment and the data stream mining are given in Section V.

II. Related Work

Data stream mining is a critical technique in modern data analysis, focusing on processing continuous streams of data to extract meaningful patterns and insights in real-time. This is particularly useful in applications where immediate decision-making is crucial, such as network monitoring, financial transactions, and personalized recommendations. The fundamental challenge in data stream mining is to efficiently handle the high volume and velocity of data while maintaining accurate and timely insights [1].

Concept drift is a phenomenon that complicates data stream mining, referring to the change in the statistical properties of the target variable over time. This drift can be gradual or abrupt and can significantly impact the performance of predictive models if not properly managed. For example, in a spam detection system, the characteristics of spam emails might change over time, necessitating continuous model updates [2]. Effective detection and adaptation to concept drift are essential to maintain model accuracy and relevance, with various techniques being proposed to address this issue, including incremental learning and ensemble methods [3].

Adversarial attacks present another significant challenge in data stream mining. These attacks involve deliberately manipulating input data to deceive the model, causing it to make incorrect predictions. Such attacks are particularly concerning in security-sensitive applications like network intrusion detection and fraud detection. Robust models must be designed to detect and adapt to these adversarial activities, ensuring the integrity and reliability of the mining process [4]. Techniques such as adversarial training and robust optimization are commonly employed to enhance model resilience against these attacks [5].

Prequential evaluation, also known as interleaved test-then-train, is a widely used method in data stream mining for continuously assessing model performance. This evaluation method involves using each incoming data point first as a test instance and then as a training instance, providing an ongoing assessment of model accuracy and adaptability as new data arrives [6]. This approach is particularly useful for timely identification of concept drift and for maintaining the accuracy of adaptive models in dynamic environments [7]. Prequential evaluation is preferred over traditional static evaluation methods as it reflects the real-time performance of the model more accurately [8].

III. Concept Drift

This section of the report covers the initial experiments considering data streams with valid concept drifts. Initially the data preparation was done by creating a data stream with 100.000 instances using SEA and AGRAWAL generator functions. Concept drifts were generated by switching between different density functions embedded in the generators that belong to the scikit-multiflow library. There are two mentioned datasets in the assignment which are called as ”Spam” and ”Electricity” datasets. These datasets are downloaded from the link provided from GitHub and all datasets are prepared to create a valid data stream to evaluate the performance of the data stream classifiers.

The Hoeffding Tree Classifier was used as the base comparison model against the tested drift-aware methods. The Adaptive Random Forest model was selected as the model requested in part 3.b.1 of the assignment guideline which

is a built-in ensemble drift-aware method inside the scikit-multiflow library. Results of utilizing the Adaptive Random Forest Classifier as the ensemble model for different values of ensemble predictors are shown in Figures 1-4.

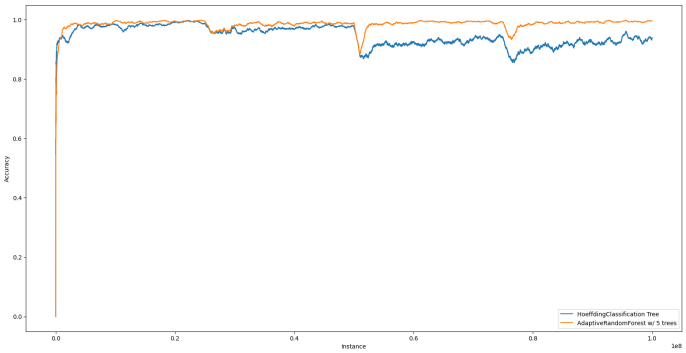

Fig. 1: Performance comparison of Adaptive Random Forest Classifier against a single Hoeffding Tree for different number of neighbours on the SEA data stream generator.

In the Figure it can be seen that while the performance of the basic Hoeffding Tree Classifier is severely hurt from the concept drifts, the Adaptive Random Forest classifier can handle it using drift detection mechanism. It can be seen that while the accuracy of a single decision tree increases gradually after a concept drift, the random forest mechanism with the drift detection algorithm can reset the trees and be able to adapt the changes better. For the SEA dataset the overall accuracy of the Hoeffding Tree is 0.946 while the overall accuracy of Adaptive Random Forest is 0.988 which is a significant increase combined with the observation of the phenomena in Figure 1.

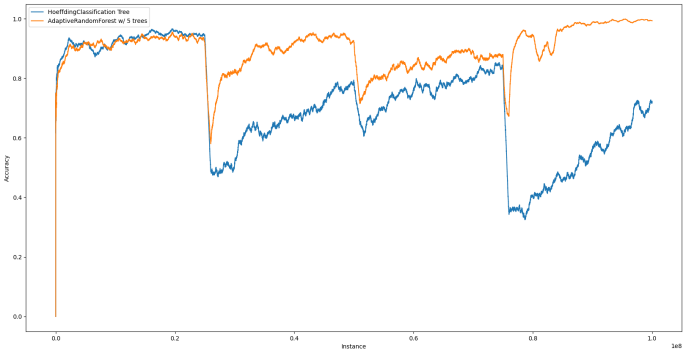

Fig. 2: Performance comparison of Adaptive Random Forest Classifier against a single Hoeffding Tree for different number of neighbours on the AGRAWAL data stream generator.

Similar with the SEA generator the drift-aware Adaptive Random Forest method performed better than a single tree. The overall accuracy of the Hoeffding Tree Classifier was 0.714 while the overall accuracy of the Adaptive Random Forest was 0.914 which is a significant increase compared to the models' performance on SEA dataset.

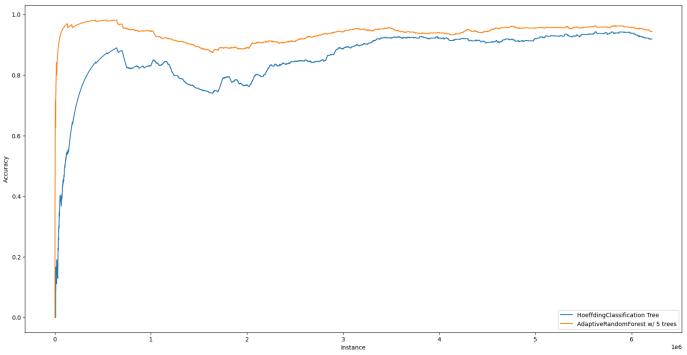

Fig. 3: Performance comparison of Adaptive Random Forest Classifier against a single Hoeffding Tree for different number of neighbours on the spam dataset.

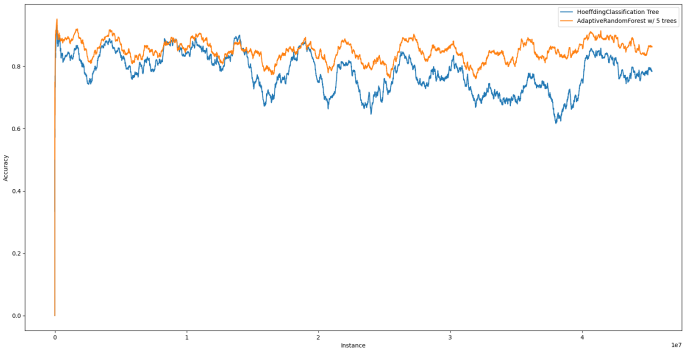

Fig. 4: Performance comparison of Adaptive Random Forest Classifier against a single Hoeffding Tree for different number of neighbours on the electricity dataset.

For both of the real datasets the Adaptive Random Forest Classifier did not show performance significantly higher than the base model Hoeffding Tree Classifier. Compared to synthetic datasets real datasets contain irregularities with higher frequency which might make it harder to detect drifts. For the "Spam" dataset the accuracy of the Hoeffding Tree Classifier is 0.949 whereas the accuracy of the Adaptive Random Forest was 0.877. For the "Electricity" dataset the accuracy of the Hoeffding Tree Classifier is 0.859 whereas the accuracy of the Adaptive Random Forest was 0.780.

After implementing the Adaptive Random Forest Classifier following the part 3.b.1 of the report, a custom drift-aware classifier has been built using the Hoeffding Tree Classifier as the base function of the ensemble model. For the model, the predicted classes are determined using a weighted majority voting algorithm. Weight of each tree is updated throughout the training process using an individual model's performance on predicting a data instance. If the prediction is correct, its weight was multiplied with 1.05 and 0.95 if wrong. Before initializing each tree classifier a feature selection has been done similar to the one in classical random forests. By

assigning only a subset of features to each tree, the learners are weakened and became elements of an ensemble model. For the drift detection, the Adaptive Windowing (ADWIN) function is imported from scikit-multiflow and integrated to the model. The algorithm resets the classifiers if the drift detector detects a number of drifts in a row which is a new hyperparameter of the new ensemble model. The Results of using this model with 5 trees can be seen from Figures 5-8 with the corresponding hyperparameters. The sensitivity parameter of the ADWIN function will be mentioned as delta and the number of detected changes to call it a drift is named as the warning threshold in this report.

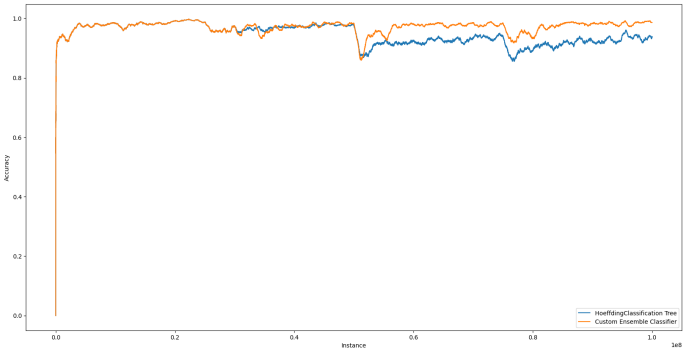

Fig. 5: Performance comparison of Custom Ensemble Classifier against a single Hoeffding Tree for delta = 0.01 and warning threshold equal to 6 for the SEA data stream generator.

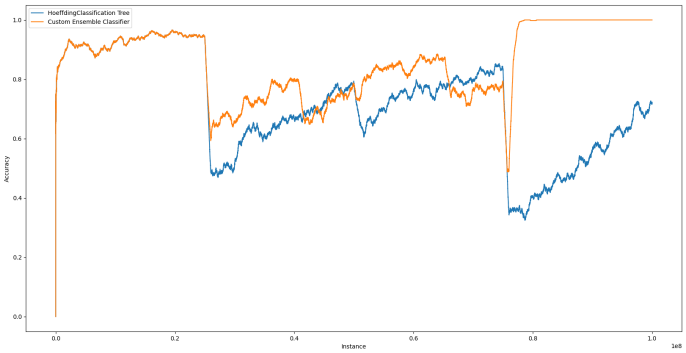

Fig. 6: Performance comparison of Custom Ensemble Classifier against a single Hoeffding Tree for delta = 0.01 and warning threshold equal to 7 for the AGRAWAL data stream generator.

As it can be seen from results, for the synthetic datasets the proposed model can detect drifts and be able to adapt to the new data.

Compared to the Adaptive Random Forest, the custom designed model has an overall accuracy of 0.961 and 0.834 for the datasets SEA and AGRAWAL respectively. By changing the hyperparameters of the model sensitivity of the drift detection algorithm can be improved and better results can be obtained. Compared to Adaptive Random Forest algorithm

the custom ensemble model adapts significantly worse on the real data. Compared with the Hoeffding Tree the ensemble model barely performs better than the base element. For the spam dataset the overall accuracy of the custom model is 0.874 whereas the overall accuracy of the Hoeffding Tree is 0.877. For the electricity dataset, where the model performs comparably better than the spam dataset the accuracy of the custom model is 0.799 and the overall accuracy of the Hoeffding Tree is 0.779. The results can be further examined from Figures 7 and 8.

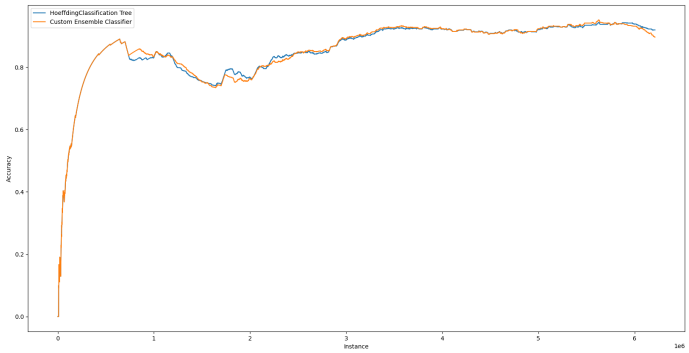

Fig. 7: Performance comparison of Custom Ensemble Classifier against a single Hoeffding Tree for delta = 0.01 and warning threshold equal to 4 for the Spam dataset.

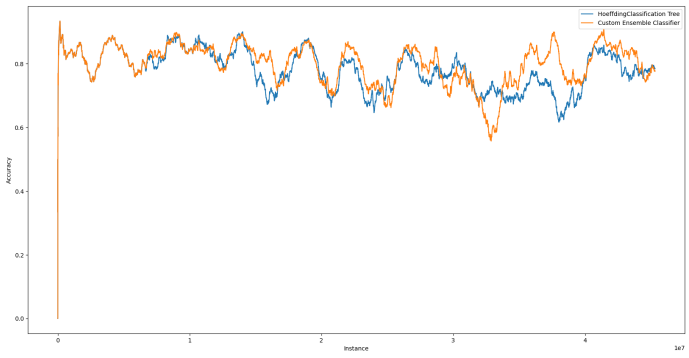

Fig. 8: Performance comparison of Custom Ensemble Classifier against a single Hoeffding Tree for delta = 0.002 and warning threshold equal to 4 for the Electricity dataset.

Both the Adaptive Random Classifier and the Custom Ensemble Model employs the ADWIN detector for drift detection. In Adaptive Random Forest the algorithm employs a nested drift detection scheme where the changes of the detected changes affect if model catches a drift. The custom made model has a more naive approach on detecting the drifts. There are as many drift detectors as the number of predictors in the ensemble. If a drift detector detects drifts that can pass a threshold in a row it resets the corresponding predictor. Whereas the prediction method of the custom model relies on a simple weighting between the trees whereas the Adaptive Random Forest uses a more sophisticated bagging technique. The overall accuracy of the tested 3 models on 4 different datasets are given in Table I.

TABLE I: Accuracy Results of Models on Datasets

Dataset	Hoeffding Acc	ARF Acc	Custom Model Acc
SEA	0.946	0.986	0.971
AGRAWAL	0.714	0.899	0.858
Spam	0.877	0.937	0.874
Electricity	0.780	0.850	0.799

In the following section the affect of adversarial attacks on the dataset will be inspected and the results of the technique designed for the custom model to protect itself from adversarial attacks will be explained.

IV. Adversarial Attacks

In this part adversarial attacks will be inspected and a method to detect them will be proposed. Adversarial attacks deceive models as they are concept drifts where the effects can be seen from Figures 9 and 10.

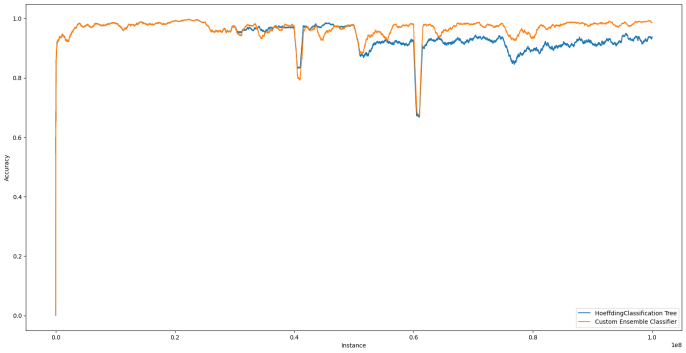

Fig. 9: Performance of the Custom model and Hoeffding Tree against adversarial attacks in the SEA data stream.

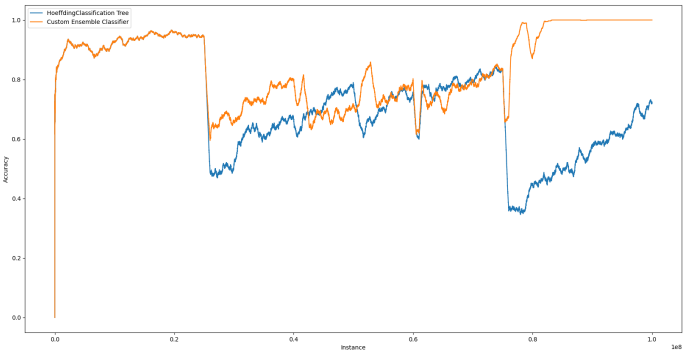

Fig. 10: Performance of the Custom model and Hoeffding Tree against adversarial attacks in the AGRAWAL data stream.

By observing the results in Figures 9 and 10, it can be argued that the models are resettled at the points where the adversarial attacks are injected. This has caused unnecessary information loss and loss of the learning process of the model. To overcome this problem an attack detection method was developed so that the models will stop training if the detector outputs that the currently arriving data contains a poisonous attack for the development of the model.

The task of detecting adversarial attacks can be done by implementing a method of the scikit-learn library called Isolation Forest. This method creates a random forest mechanism that creates trees with feature selection. The individual trees select split points randomly between the minimum and maximum values. Since the points belonging to a concept will densely located in a region outliers will be easily classified by quickly splitting them to nodes where they will be left behind.

The technique employs a metric called anomaly score which decides if a data point is an anomaly which is highly correlated with the inverse of the length of the data point to the top node of the tree. For this implementation, the Isolation Forest algorithm is imported from the sklearn library and implemented to the custom ensemble model. Results of this implementation on the SEA and AGRAWAL dataset can be seen from Figures 11 and 12 respectively.

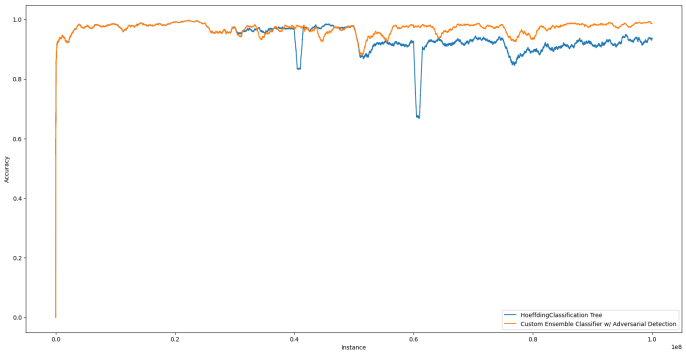

Fig. 11: Performance of the Custom model and Hoeffding Tree against adversarial attacks in the SEA data stream.

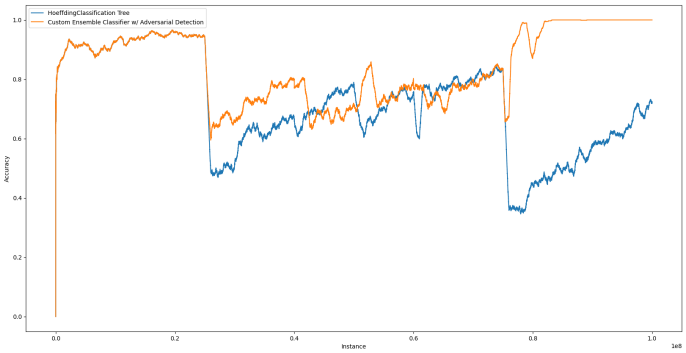

Fig. 12: Performance of the Custom model and Hoeffding Tree against adversarial attacks in the AGRAWAL data stream.

From the figures it can be seen that the adversarial attacks were succesfully detected using the Isolation Forest algorithm. For the spesific set-up Isolation Forest with 10 number of elements were employed. The sensitivity values and detection margin values were also selected considering the performance of the model on various data. As a result the overall accuracy of the custom model on the SEA dataset appeared to be 0.962 and the performance of the model on the AGRAWAL dataset appeared to be 0.823. From these results it can be argued that the used algorithm has worked well. Because of the computation time of that the Isolation Forest requires only several trials can be made and therefore the parameters couldn't be optimized with the given settings. Therefore there were not enough time for experimenting the results with higher rate of adversarial attacks.

V. Conclusion

.

In this assignment, concepts of data stream mining were investigated with a focus on concept drift and adversarial attacks. The experiments compared the performance of the Adaptive Random Forest and a custom ensemble classifier. The Adaptive Random Forest outperformed the single Hoeffding Tree classifier on synthetic datasets, but both models struggled with real-world data, indicating a need for better drift adaptation techniques. Impact of adversarial attacks were also explored, which mimic concept drifts and degrade model performance. To counter this, Isolation Forest algorithm was implemented for attack detection, which improved model stability and accuracy.In summary, I learned that while ensemble methods and drift detectors like ADWIN are effective in controlled environments, they require refinement for real-world applications. The integration of anomaly detection techniques like the Isolation Forest enhances model resilience against adversarial attacks, emphasizing the need for robust, adaptive approaches in data stream mining.

References

[1] A. Alazab and T. Al-Quraishi, "Roadmap of concept drift adaptation in data stream mining, years later," *La Trobe University*, 2024. [Online]. Available: https://opal.latrobe.edu.au/ndownloader/files/44734168

[2] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122002854

[3] M. M. Ferdaus, T. Dam, and S. Alam, "X-fuzz: An evolving and interpretable neurofuzzy learner for data streams," *IEEE Transactions on Fuzzy Systems*, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10432928/

[4] M. A. Elsayed and N. Zincir-Heywood, "Boostsec: Adaptive attack detection for vehicular networks," *Journal of Network and Systems Management*, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10922-023-09781-w

[5] B. Zhang, "Handling concept drift using the correlation between multiple data streams," Ph.D. dissertation, ProQuest, 2022. [Online]. Available: https://search.proquest.com/openview/f93e1650d4e6df7ab348e2dc28e32791/1?pq-origsite=gscholar&cbl=2026366&diss=y

[6] S. Zhang, P. Tino, and X. Yao, "Hierarchical reduced-space drift detection framework for multivariate supervised data streams," *IEEE Transactions on Knowledge and Data Engineering*, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9540313/

[7] B. Ida Seraphim and E. Poovammal, "Adversarial attack by inducing drift in streaming data," *Wireless Personal Communications*, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11277-021-08479-z

[8] P. Casas, P. Mulinka, and J. Vanerio, "Should i (re) learn or should i go (on)? stream machine learning for adaptive defense against network attacks," in *Proceedings of the 6th ACM Workshop on Artificial Intelligence and Security*. ACM, 2019, pp. 201–209. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3338468.3356829