GE 461: Introduction to Data Science

Spring 2024

Project: Data Stream Mining

Adaptive Algorithms in Data Stream Mining: An Analysis of Concept Drift and Adversarial Attack Detection

Introduction

Data stream mining has become an essential area of research in data science due to the continuous and rapid generation of data in various fields such as finance, telecommunications, and social media. This project focuses on exploring and evaluating different methods for handling data streams. The primary goals are to handle concept drift and detect adversarial attacks in streaming data, using both synthetic and real datasets.

To achieve these objectives, a variety of methods are used, including adaptive random forest (ARF), streaming agnostic model with k-nearest neighbors (SAMKNN), and dynamic weighted majority (DWM) classifiers. Additionally, a custom ensemble classifier built from Hoeffding tree classifiers is implemented.

The datasets used include two synthetic datasets generated from the Agrawal and SEA generators, and two real datasets comprising spam and electricity data. Each dataset presents unique challenges for the classifiers.

Concept drift is the changes in the statistical properties of the target variable over time. This project assesses various strategies for detecting and adapting to concept drift. Additionally, the project addresses the issue of instance based adversarial attacks.

1. Dataset Preparation

Two synthetic and two real datasets are used in this project. The first synthetic dataset is constructed by sampling the Agrawal generator of the scikit-multiflow library 100,000 times. The second synthetic dataset is constructed by sampling the SEA generator of the same library 100,000 times. For the synthetic libraries, to create abrupt concept drift points, classification functions 0, 1, 2, and 3 are used for 25,000 sample long partitions. The real datasets are the spam and electricity datasets. All datasets except spam have numerical features and all datasets have binary categorical targets.

Dataset Name	Number of Samples	Number of Features	Percentage of Samples With Target = 0	Percentage of Samples With Target = 1
Agrawal	100,000	9	49.1%	50.9%
SEA	100,000	3	35.7%	64.3%
Spam	6,213	499	66.7%	33.3%
Electricity	45,312	6	57.5%	42.5%

Table 1: Overview of datasets.

Some datasets are balanced while others can be considered imbalanced. Another thing to note is that the "Spam" dataset has many more features compared to the other datasets.

2. Handling Concept Drift

2.1. Scikit-Multiflow Inbuilt Algorithms

Adaptive random forest (ARF), streaming agnostic model with k-nearest neighbors (SAMKNN), and dynamic weighted majority (DWM) classifiers are implemented and evaluated on all datasets. All classifiers use the default parameters.

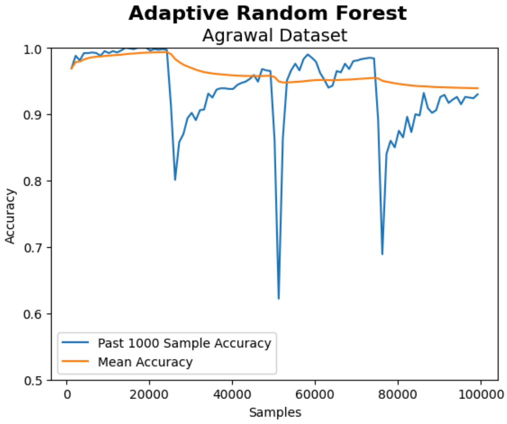

Figure 1: Windowed and mean test accuracy of **ARF** classifier on the **Agrawal** dataset.

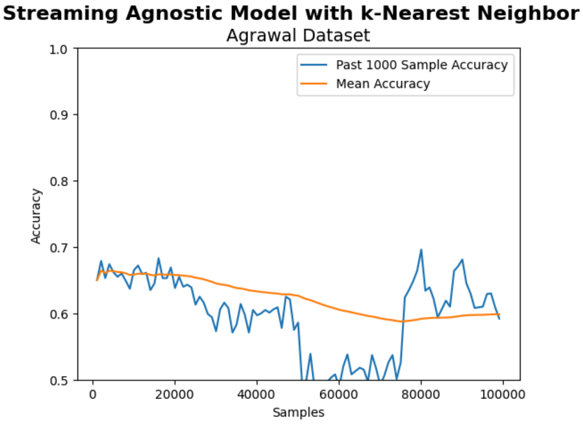

Figure 2: Windowed and mean test accuracy of **SAMKNN** classifier on the **Agrawal** dataset.

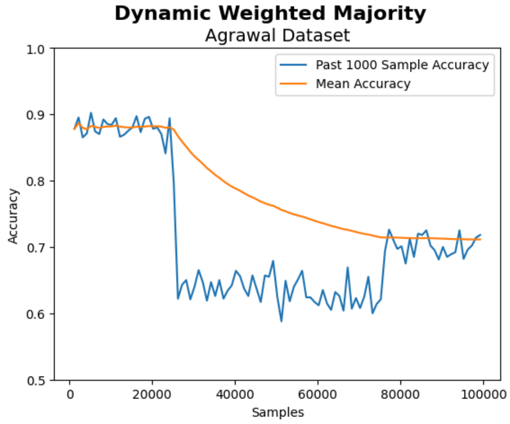

Figure 3: Windowed and mean test accuracy of **DWM** classifier on the **Agrawal** dataset.

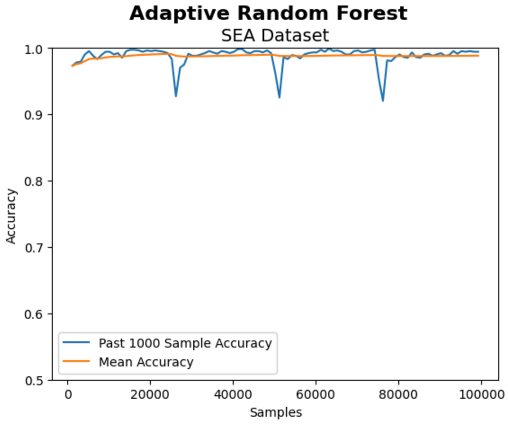

Figure 4: Windowed and mean test accuracy of **ARF** classifier on the **SEA** dataset.

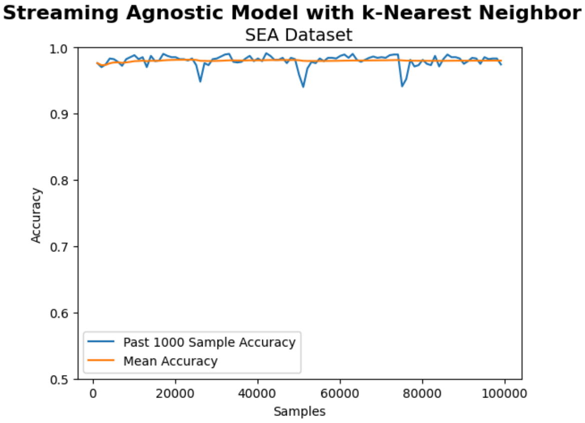

Figure 5: Windowed and mean test accuracy of **SAMKNN** classifier on the **SEA** dataset.

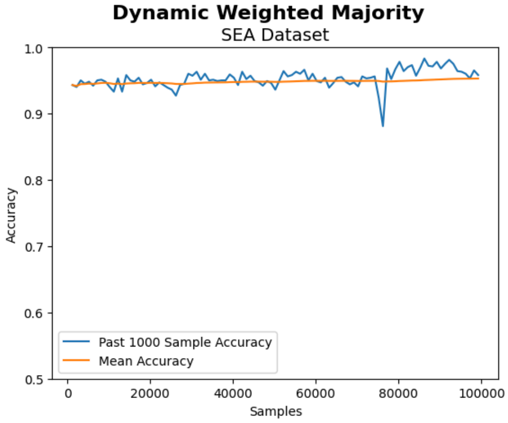

Figure 6: Windowed and mean test accuracy of **DWM** classifier on the **SEA** dataset.

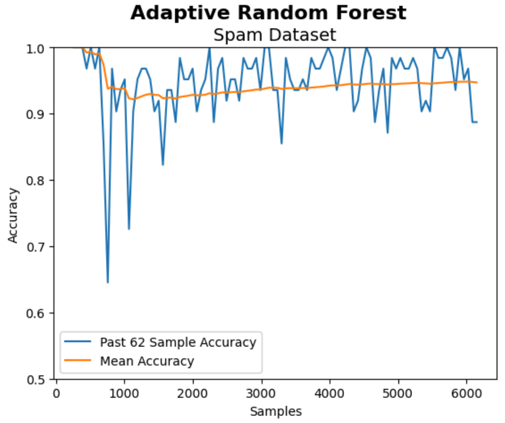

Figure 7: Windowed and mean test accuracy of **ARF** classifier on the **Spam** dataset.

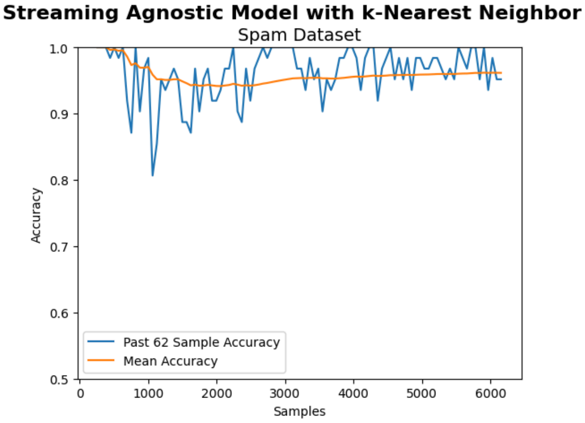

Figure 8: Windowed and mean test accuracy of **SAMKNN** classifier on the **Spam** dataset.

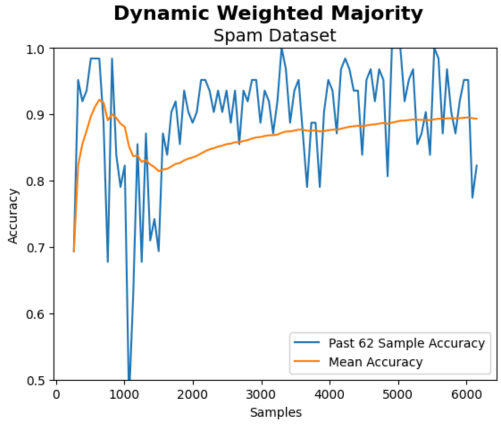

Figure 9: Windowed and mean test accuracy of **DWM** classifier on the **Spam** dataset.

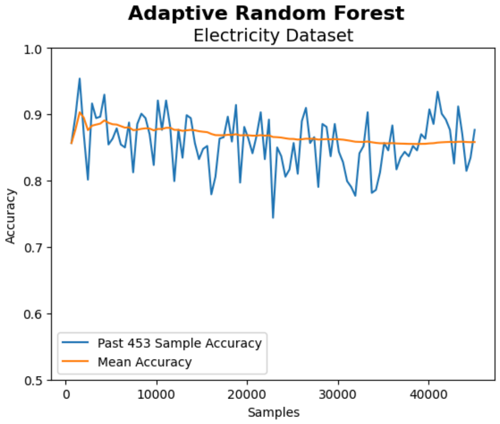

Figure 10: Windowed and mean test accuracy of **ARF** classifier on the **Electricity** dataset.

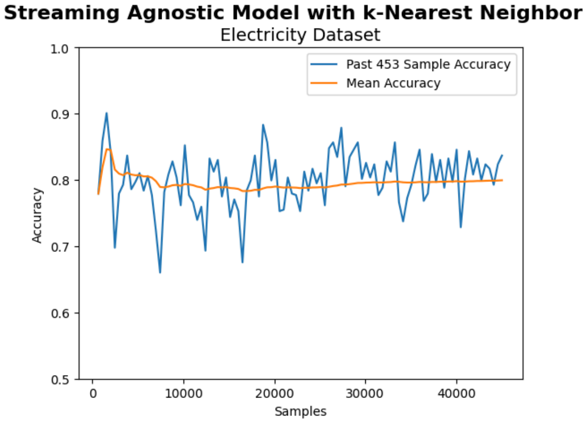

Figure 11: Windowed and mean test accuracy of **SAMKNN** classifier on the **Electricity** dataset.

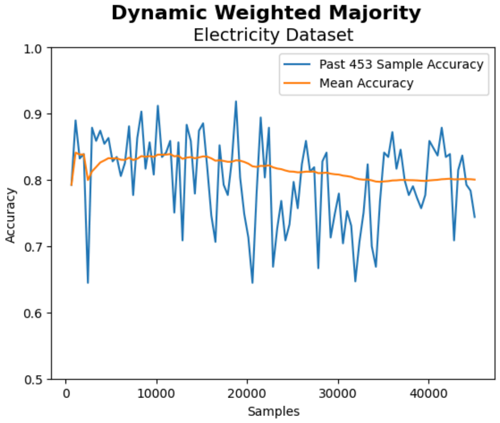

Figure 12: Windowed and mean test accuracy of **DWM** classifier on the **Electricity** dataset

From figures 1, 2, and 3, it can be seen that the ARF classifier adapts to concept drifts much better compared to others on the Agrawal dataset. Figures 4, 5, and 6 show that the Sea dataset is relatively easy for all classifiers. About the Spam dataset, figures 7, 8, and 9 depict more consistent behavior for SAMKNN followed by ARF and DWM. While the shape of the lines look similar, figures 10, 11, and 12 show that ARF performs the best on the electricity dataset.

2.2. Ensemble Classifier From Scratch

The custom ensemble classifier is implemented and evaluated on all datasets. It is built from **10** Hoeffding tree classifiers from the scikit-multiflow library. The classifier uses **online bagging** for training and **majority voting** for prediction. For each sample, each classifier is trained k times where $k = Poisson(1)$. The idea is taken from [1]. For concept drift detection, the **adaptive windowing method** from the scikit-multiflow library was used (**ADWIN**). When concept drift is detected, all classifiers in the ensemble are reset.

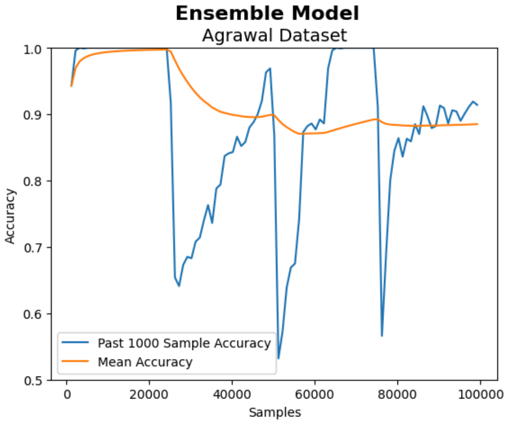

Figure 13: Windowed and mean test accuracy of **Custom** classifier on the **Agrawal** dataset.

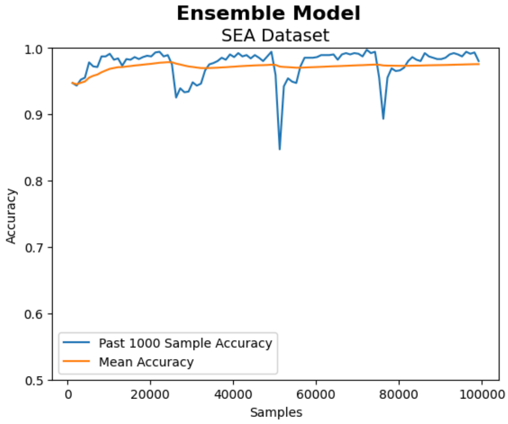

Figure 14: Windowed and mean test accuracy of **Custom** classifier on the **SEA** dataset.

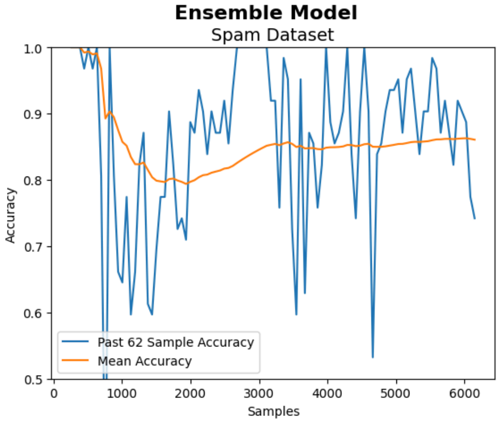

Figure 15: Windowed and mean test accuracy of **Custom** classifier on the **Spam** dataset.

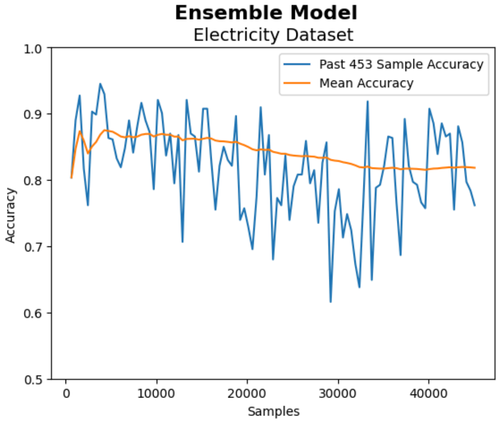

Figure 16: Windowed and mean test accuracy of **Custom** classifier on the **Electricity** dataset.

While the artificially inserted abrupt concept drifts in the synthetic datasets are nicely detected by the ADWIN algorithm (figures 13 and 14), frequent detections by the same algorithm cause the learners to reset numerous times in the real datasets. This is the reason for the spiky accuracy plots in figures 15 and 16.

2.3. Summary

Dataset Name	Classifier Name	Mean Test Accuracy
Agrawal	ARF	91.3%
...	SAMKNN	59.8%
...	DWM	71.1%
...	Custom	88.5%
SEA	ARF	98.9%
...	SAMKNN	97.9%
...	DWM	95.3%
...	Custom	97.5%
Spam	ARF	94.7%
...	SAMKNN	96.1%
...	DWM	89.3%
...	Custom	86.1%
Electricity	ARF	85.8%
...	SAMKNN	79.9%
...	DWM	80.0%
...	Custom	81.8%

Table 2: Summary of classifier performance by dataset.

Considering all datasets, ARF performs the best on Agrawal, SEA, and Electricity, and SAMKNN performs the best on Spam. The Custom algorithm performs comparably with the others and comes last in only on the Spam dataset.

3. Handling Adversarial Attacks

3.1. Dataset Preparation

To simulate adversarial attacks, in both of the synthetic datasets, the target values of randomly selected 50 samples from the interval 40,000 - 40,500, and 100 samples from the interval 60,000 - 60,500 are flipped.

3.2. Modified Custom Ensemble Classifier

The z-score statistic is used for the detection of flipped samples. A large z-score means a lower probability of a sample being from the same distribution. Samples with z-scores over a threshold are excluded from training. The z-score of a sample is calculated with relation to a moving window of previous values in the data stream. There are two windows for samples with target values of 0 and 1. The z-score of the current sample is calculated using the same window as the alleged target value of the sample. The maximum value of z-scores of all features is compared with the threshold. The threshold must be tuned to avoid false positives and false negatives. Note that this method assumes Gaussian distribution of the samples, which is a key limitation of this method.

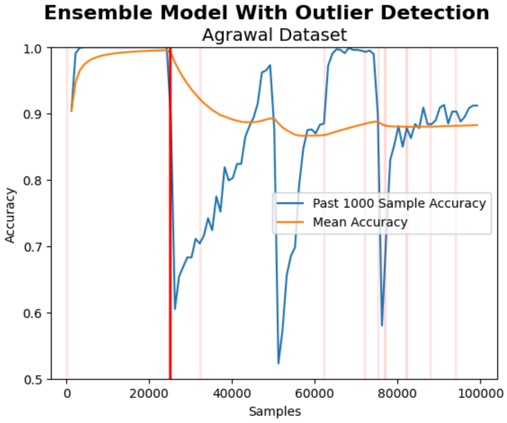

Figure 17: Windowed and mean test accuracy of **Custom** classifier on the **corrupted Agrawal** dataset.

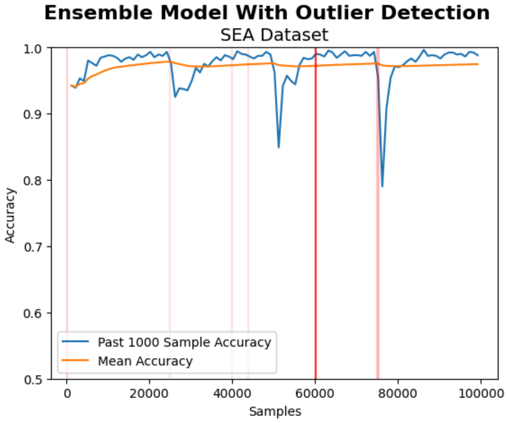

Figure 18: Windowed and mean test accuracy of **Custom** classifier on the **corrupted SEA** dataset

In figures 17 and 18, the red vertical lines represent samples detected as outliers. In the Agrawal dataset, the abrupt concept drift at sample number 25,000 triggered many false positives. Also, none of the adversarial samples at 40,000 and 60,000 are detected. In the SEA dataset, the abrupt concept drift at sample number 75,000 triggered some false positives. Also, some of the adversarial samples at 40,000 and 60,000 are detected. In both datasets, some false positives are present throughout.

The detection of adversarial samples is heavily dependent on the underlying distribution of the dataset.

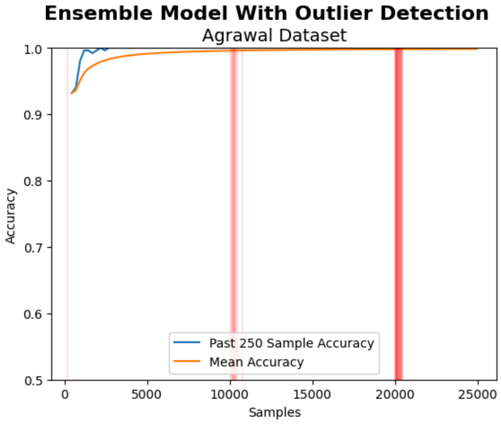

Figure 19: Windowed and mean test accuracy of **Custom** classifier on the **corrupted Agrawal** dataset (0, 25,000).

Taking the first part of the Agrawal dataset (0, 25,000) and flipping samples at 10,000 and 20,000 samples, from figure 19, we see that the adversarial samples are detected. The reason for the failed detection of said samples in figure 17, and the opposite in figure 19 is the classification function used in data generation. As explained in section 1, for samples in the 40,000 and 60,000 ranges, the classification functions 1 and 2 are used respectively. As understood from the figures above, the z-score method is not effective for these classification functions. However, when the adversarial samples are inserted into the samples generated with the classification function 0, many of them are detected successfully.

The versions of the Custom classifier in sections 2.2. and 3.3. perform very similarly on the corrupted dataset. This might be because of the false positives and false negatives of the modified Custom classifier.

Results and Discussion

While different classifiers perform differently on different datasets, on average, the **ARF** classifier performed the best on the datasets examined in this project compared to the other classifiers. This is mainly because of the concept drift detection capabilities of the classifier. While other classifiers experienced deteriorated performance long after a concept drift had occurred, the ARF classifier adapted quickly and thus did not experience a large decrease in accuracy.

For the custom ensemble algorithm, while the approach used for detection of concept drift was the ADWIN algorithm, many other more complex algorithms exist and can be used. The situation is the same for the actions taken in response to a concept drift detection. Rather than resetting all the classifiers, a more sophisticated method can be utilized. However, both of these approaches are effective in countering the effects of concept drift, especially in the synthetic datasets.

The method for detection of adversarial samples was the thresholded z-score. The most apparent limitation of this approach is the assumption of Gaussian distribution of the data. As a result, the z-score is not a reliable method for the detection of adversarial samples. Usage of more sophisticated methods such as isolation forests are highly recommended for this purpose.

Conclusion

In this project, I have explored the challenges associated with data stream mining, focusing on handling concept drift and detecting adversarial attacks in streaming data. By evaluating various classifiers such as adaptive random forest (ARF), streaming agnostic model with k-nearest neighbors (SAMKNN), dynamic weighted majority (DWM), and a custom ensemble classifier built from Hoeffding tree classifiers, I have gained experience about their performance across different datasets.

The ARF classifier consistently demonstrated superior performance, particularly in adapting to concept drift, making it a robust choice for real-time data stream mining. While the custom ensemble classifier showed acceptable performance, its usage of simpler methods for handling concept drift and adversarial sample detection made the need for more sophisticated approaches obvious. The thresholded z-score method for adversarial detection was limited by its assumption of Gaussian distribution, making the importance of exploring alternative techniques such as isolation forests clear.

References

[1] N. C. Oza, “Online Bagging and Boosting,” Systems, Man and Cybernetics, Jan. 2005, doi: https://doi.org/10.1109/icsmc.2005.1571498.