University of Ottawa

Kafka Streams

Assignment 2

Mohamed Ahmed Elnamoury
11-22-2021

Table of Contents

List of Figures

1. Network Intrusion Detection

1.1. Algorithm Description

I Decided to use Random Forest as it's great algorithm that produces great results without even without hyperparameter tuning, Random Forest combine simplicity of decision trees with flexibility resulting in vast improvement in accuracy, I fitted Random Forest algorithm with number of estimators equal to ten estimator one of the main reasons for using Random Forest algorithm is that it's great at handling imbalanced data. It also adapts well with changes in distribution over time.

1.2. Algorithm Evaluation

Metrics:

- Accuracy is initial metric but as the distribution of the data is imbalanced domain
- F1_score is used but I used it as weighted as I took in consideration the imbalanced dataset to make sure that my evaluation is right as the incoming streams didn't change the shape of distribution that much
- Recall and Precision Weighted were used as weighted too because of imbalanced data
- ROC_AUC_SCORE the most Important metric for evaluating imbalanced dataset

Hyperparameters:

At early stage of exploring the dataset and investigation I tried to use Random Forest with the default hyperparameters, and I found that the model scored 100% accuracy and f1_score as well so I tried to reduce the number of estimators to ten estimators to take in consideration the time for fitting the dynamic model over the incoming streams the model still scoring the same accuracy so I decided to go on with ten estimators over the incoming streams.

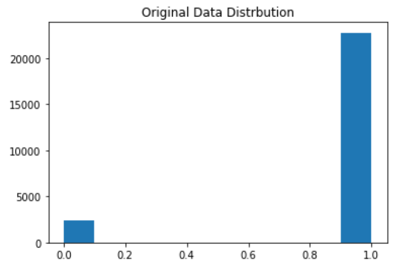
- n_estimator = 10 to reduce the fitting time during updating the data.
- Criterion = for the Gini impurity
- max_depth = none, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- min_samples_split = 2, considering the minimum split is the integer number
- max_leaf_nodes = none, then unlimited number of leaf nodes.
- Bootstrap = True
- Random_state = 0, setting a fixed number
- Max_samples = None as it will draw the shape of rows provided which will be good in the case continues learning

Performance Estimation:

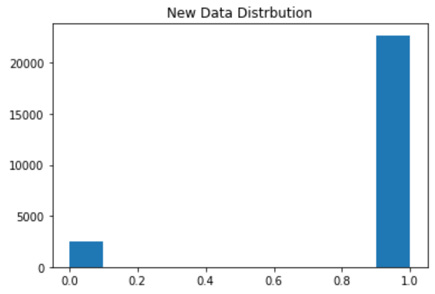
Overall, the performance of the two models were so close they oscillated in the same region one goes higher the other follows and this all will be described in detail in the upcoming section for plots and graphs but overall, the performance was so close, and this was different from my initial estimation.

1.3. Plots and Tables

Beginning



Middle



End

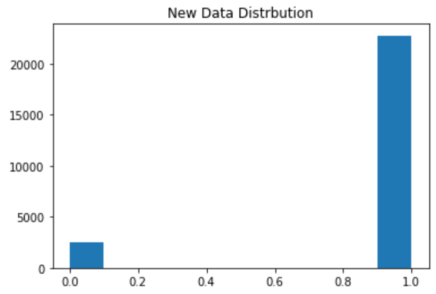
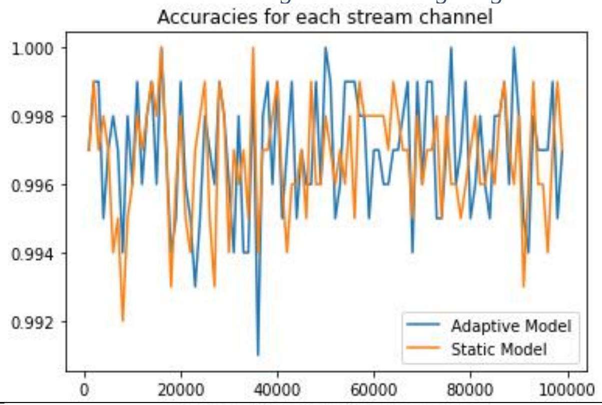
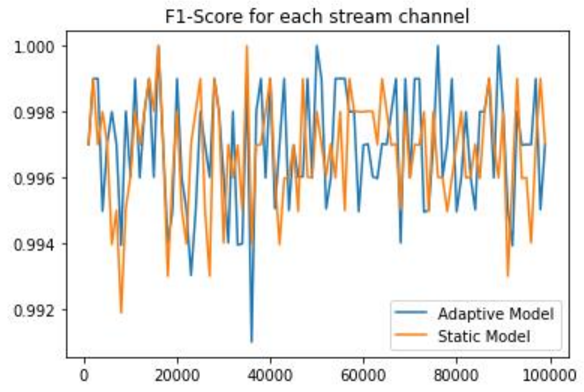
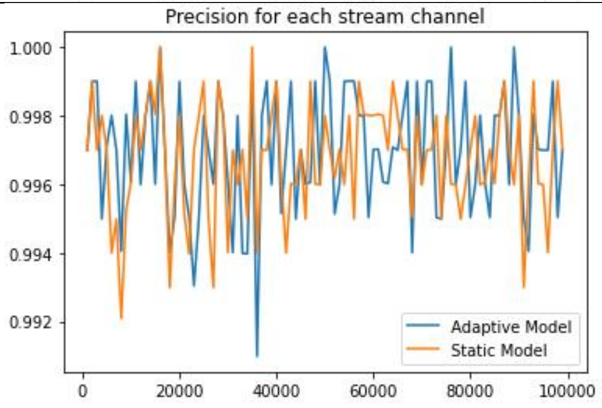
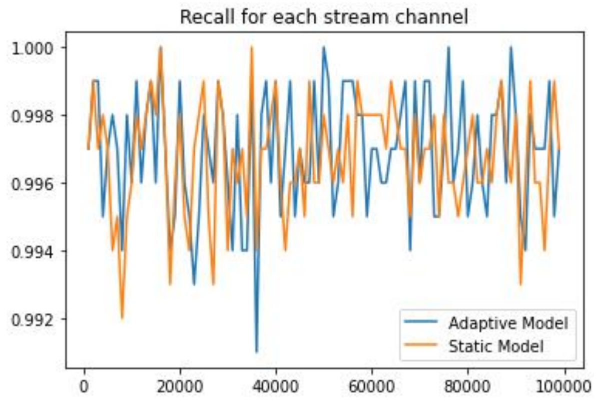


Figure 1: Investigating data distribution over time through stream









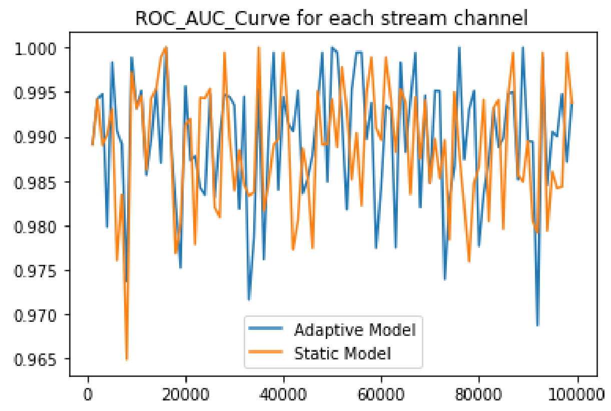


Figure 2: Graphs Comparing the two models over time using the above-mentioned metrics

1.4. Discussion

In terms of evaluation metrics, I choose it's obvious that even if there's a slight change between the two models but over time the adaptive model started to score better scores but not that huge changes and coming to the ROC_AUC_score graph you will see that the models are swapping the peaks with each other over time that you can't really tell that one model is obviously winning here or I should stick to the dynamic and exclude the static model but this all because of the distribution shape of the data as the incoming data didn't change the distribution shape of the target value from the initial dataset distribution this is the main reason that the two models scoring close score over the four evaluation metrics.

Limitations and knowledge gained: continuous Learning process is time and computational resources consuming process but it's necessary one in real world problems for sure so we can get over this problem by checking the distribution of the data over the time and big shifting in data distribution would be the trigger point for the dynamic model to start fitting onto the updated dataset, Tree based models is great in fitting to new data and score great results even the static model that was fitted only once scored great scores and stayed head to head against the dynamic model, data distribution is the main reason for model performance and how the model will deal with the distribution of the data.

1.5. Conclusion

After investigating 100000 Streams I would stick to the static model as the differences wasn't huge and it would save time and computational resources.

2.IOT Botnet Attack Detection problem

2.1. Algorithm Description

In this problem I used CatboostClassifier as boosting techniques can do well with imbalanced dataset, CatboostClassifier is built on d on gradient boosted decision trees, during training, a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to the previous trees. The number of trees is controlled by the starting parameters. So it fits well in the distribution well every time.

2.2. Algorithm Evaluation

Metrics:

- Accuracy is initial metric but as the distribution of the data is imbalanced domain
- F1_score is used but I used it as weighted as I took in consideration the imbalanced dataset to make sure that my evaluation is right as the incoming streams didn’t change the shape of distribution that much
- Recall and Precision Weighted were used as weighted too because of imbalanced data
- ROC_AUC_SCORE couldn’t be used in this problem as the number of incoming classes in each stream

Hyperparameters*:*

At early stage of exploring the dataset and investigation I tried to use Random Forest with the default hyperparameters, and I found that the model scored 98% accuracy and f1_score as well so I tried to reduce the number of estimators to ten estimators to take in consideration the time for fitting the dynamic model over the incoming streams the model still scoring the same accuracy so I decided to go on with ten estimators over the incoming streams.

- random_state = 0 two for both models
- learning_rate = 0.1 for learning process.
- depth=None,
- l2_leaf_reg=None,
- model_size_reg=None,
- rsm=None,
- loss_function=None,
- border_count=None,
- feature_border_type=None,
- per_float_feature_quantization=None,
- input_borders=None

Performance Estimation:

Overall, the performance of the two models were so close they oscillated in the same region one goes higher the other follows and this all will be described in detail in the upcoming section for plots and graphs but overall, the performance was so close, and this was different from my initial estimation.

2.3. Plots and Tables

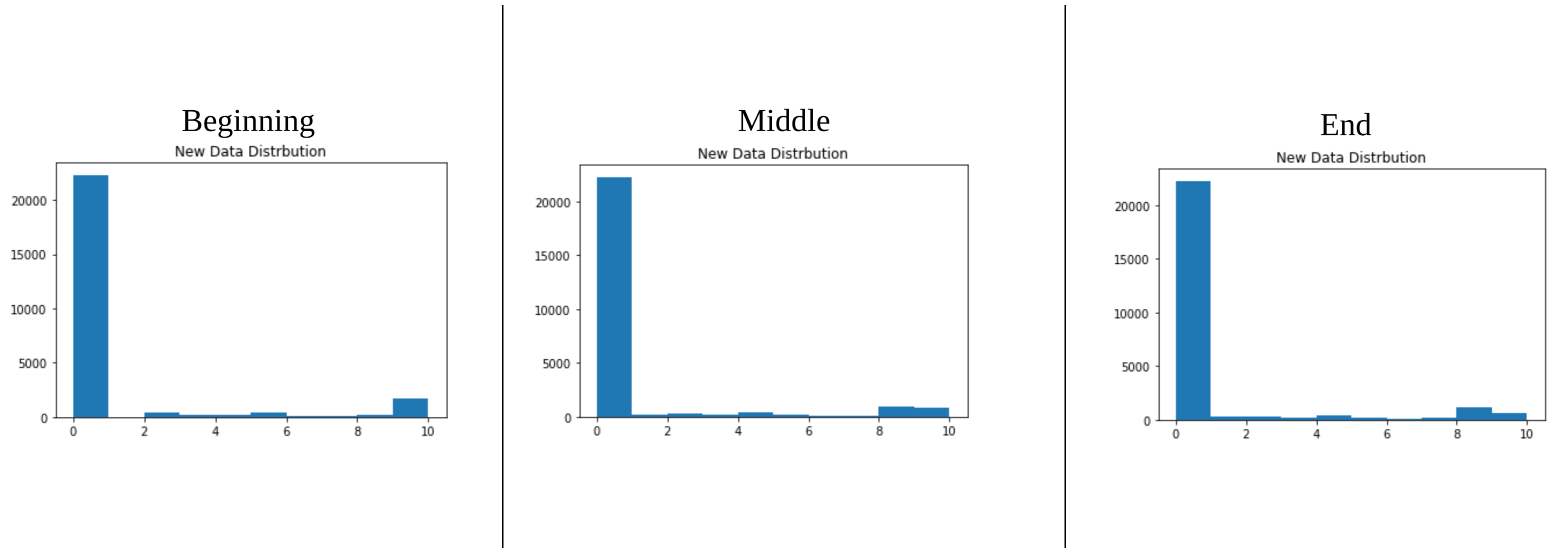


Figure 3: Investigating data distribution over time through stream

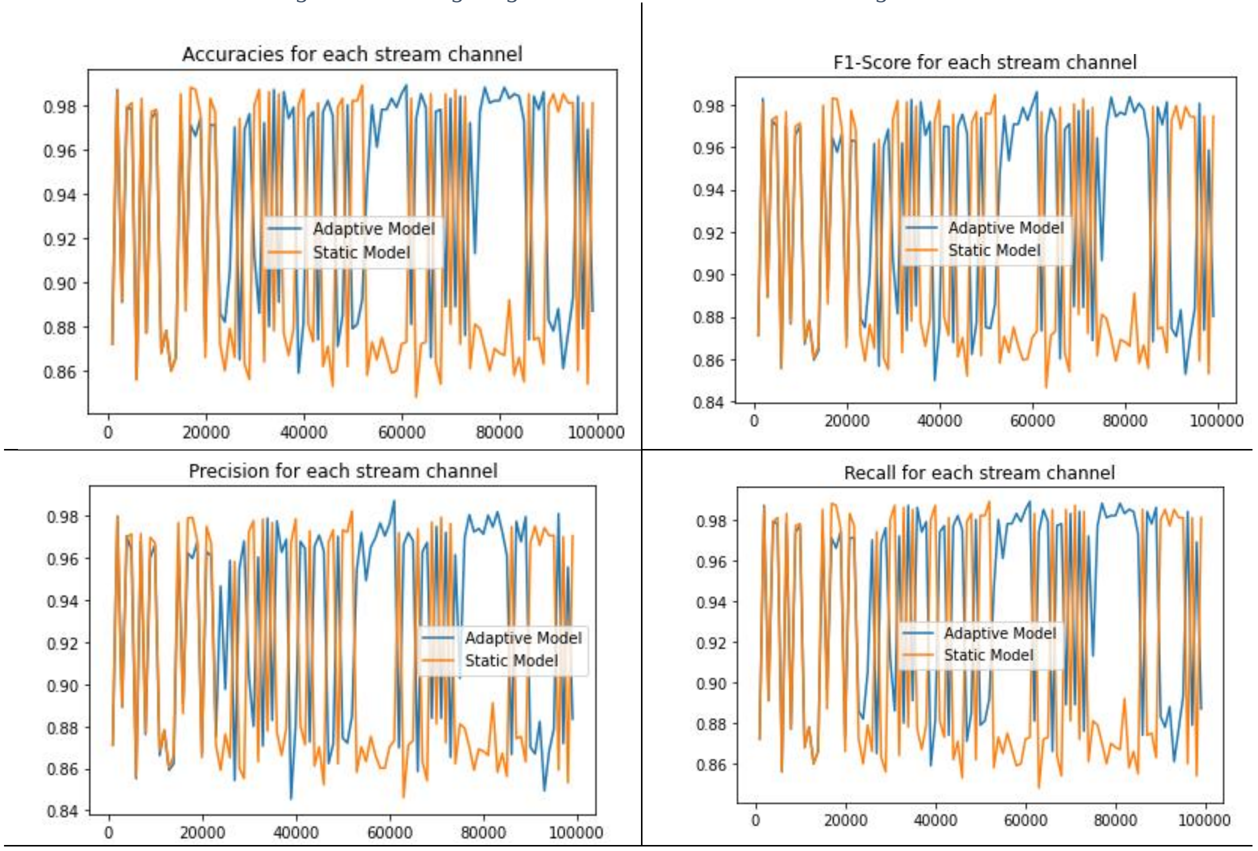


Figure 4: Graphs Comparing the two models over time using the above-mentioned metrics

2.4. Discussion

In terms of evaluation metrics, I choose it’s obvious the adaptive model started to score better scores in the region of 50000 and 6000 and over time you will see that the two models starts to intersect with each other till reaching 65000 the adaptive model started to take over the static model

till it reaches 85000 then the two models started two intersect with each other again, tell that one model is obviously winning here, the incoming data distribution changed from the beginning over the time of data streams as described in the plots describing the distribution of the data.
Limitations and knowledge gained: continuous Learning process is time and computational resources consuming process but it's necessary one in real world problems for sure so we can get over this problem by checking the distribution of the data over the time and big shifting in data distribution would be the trigger point for the dynamic model to start fitting onto the updated dataset, Tree based models built on boosting techniques is great in fitting to new data and score great results even the static model that was fitted only once scored great scores and stayed head to head against the dynamic model, data distribution is the main reason for model performance and how the model will deal with the distribution of the data.

2.5. Conclusion

After investigating 100000 Streams if the incoming streams will shift the data distribution over the time it would be good to start continuous learning process as it will help the model in adapt to the new distribution of the new dataset and perform better on new data distribution.

References:

1 – "RandomForestClassifier" sklearn.ensemble.RandomForestClassifier — scikit-learn 1.0.1 documentation

2- "CatboostClassifier' Overview - CatBoostClassifier | CatBoost

3- "Evaluation Metrics" More Performance Evaluation Metrics for Classification Problems You Should Know - KDnuggets