KAFKA STREAMS – INTRUSION DETECTION

Kamel AlaaEldin Elsehly

UNIVERSITY OF OTTAWA

“Binary Problem”

“Algorithms”:

The used algorithm is **random forest** with just 2 trees (for simplicity and fast computations):

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

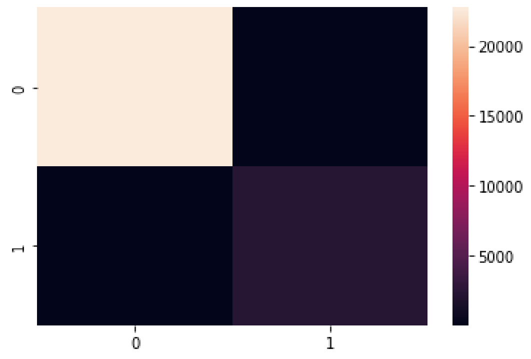
“Experiments”:

The metrics of Evaluation we used are:

Accuracy: TP+TN/TP+FP+FN+TN	Precision: TP/TP+FP
Recall: TP/TP+FN	F1-Score: 2 * (precision * recall) / (precision + recall)

the results obtained (tables, plots, etc.):

1) Static Model Results:
   - Here is the heatmap for the binary classes, as we can see there is a huge imbalance between the two classes (only 2443 records for class 1 and the rest are 0).

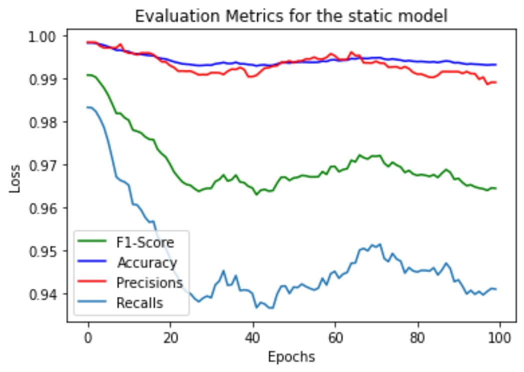
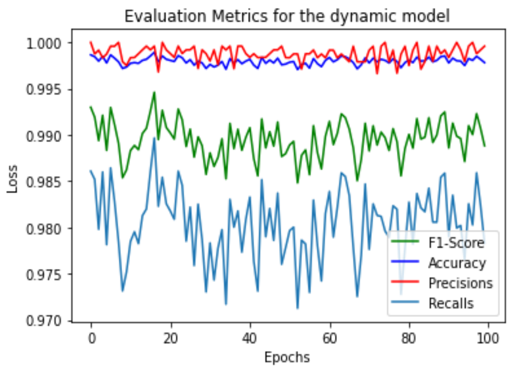


   - And here are the metrics results:

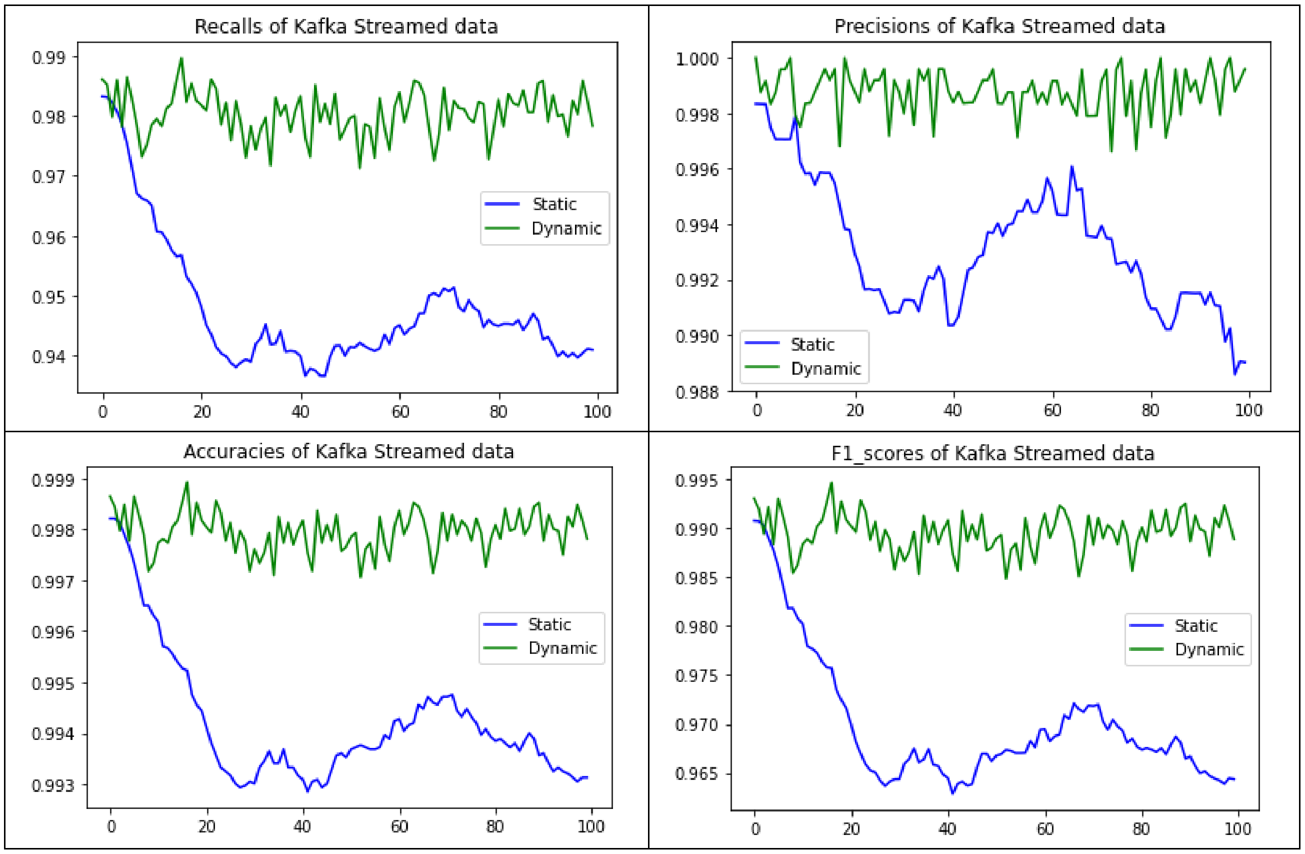
f1 score: 0.9892561983471074	Precision score: 0.9987484355444305
Accuracy score: 0.9979347869256127	recall score: 0.9799426934097422

2) Results of Dynamic Model with Kafka Streams:

   - The below plots show the metrics of each model.

This plot shows the metrics results of the static model along all baches.	And this plot shows the metrics results of the dynamic model along all batches.
	

- The below plots show the comparison between the static and dynamic model with every evaluation metric one at a time.



-

Discussion:

- From All of the above plots and tables, it's clear that adaptive or dynamic learning is very important especially with varying data like Kafka streams.
- Based on the results, the Static model performed very well with the original data in spite of the imbalance between the two classes.
- Based on the results and plots, the static model performed will with the initial data and started to downgrade the more new data is appended from the Kafka Streams until it's exposed to totally new data that differs from the one it trained on.
- Here comes the strength and the advantage the of the dynamic model which stands still despite the varying data, it may take much more computations but eventually it will keep the good performance till the end.
- So, the only advantage of the static model is the less computations against the adaptive one.

“Multiclass Problem”

“Algorithms”

The used algorithm is **random forest** with just 2 trees (for simplicity and fast computations):

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

“Experiments”:

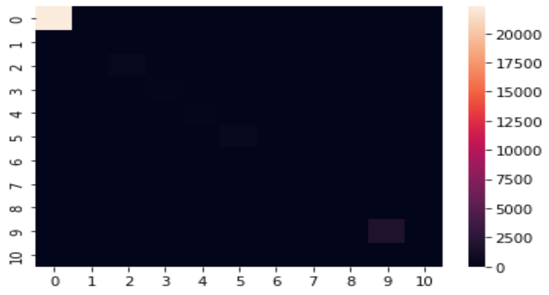
The metrics of Evaluation we used are:

Accuracy: TP+TN/TP+FP+FN+TN	Precision: TP/TP+FP (**Weighted** Average)
Recall: TP/TP+FN (**Weighted** Average)	F1-Score: 2 * (precision * recall) / (precision + recall) (**Weighted** Average)

We used the weighted average method as it calculates metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters ‘macro’ to account for label imbalance; it can result in an F-score that is not between precision and recall.

The results obtained (tables, plots, etc.):

1) Static Model Results:
   - Here is the heatmap for the multi classes, as we can see there is a huge imbalance between the 0 class and all other classes which represents types of attacks (only 3322 records for attacks classes and the rest are 0).

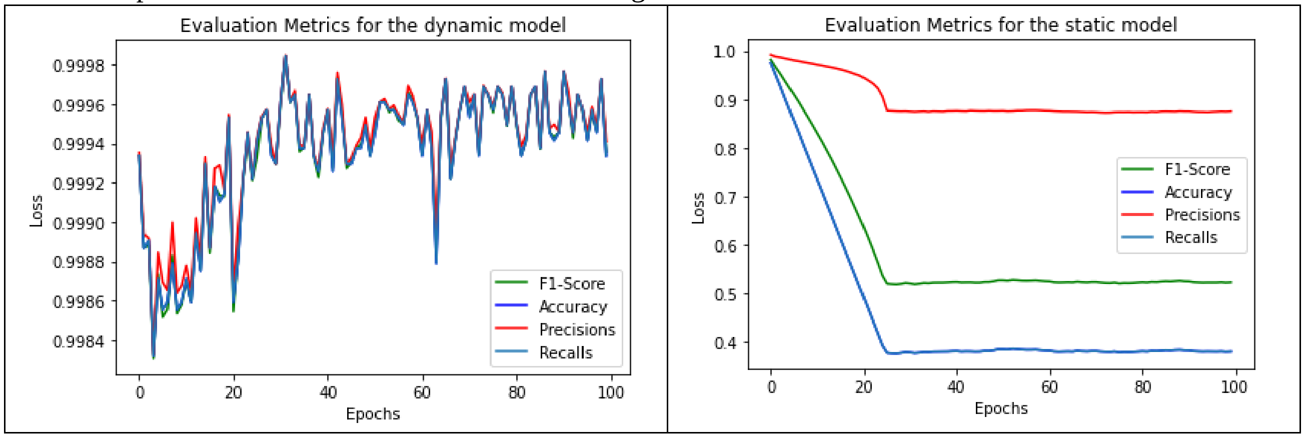


   - And here are the metrics results:

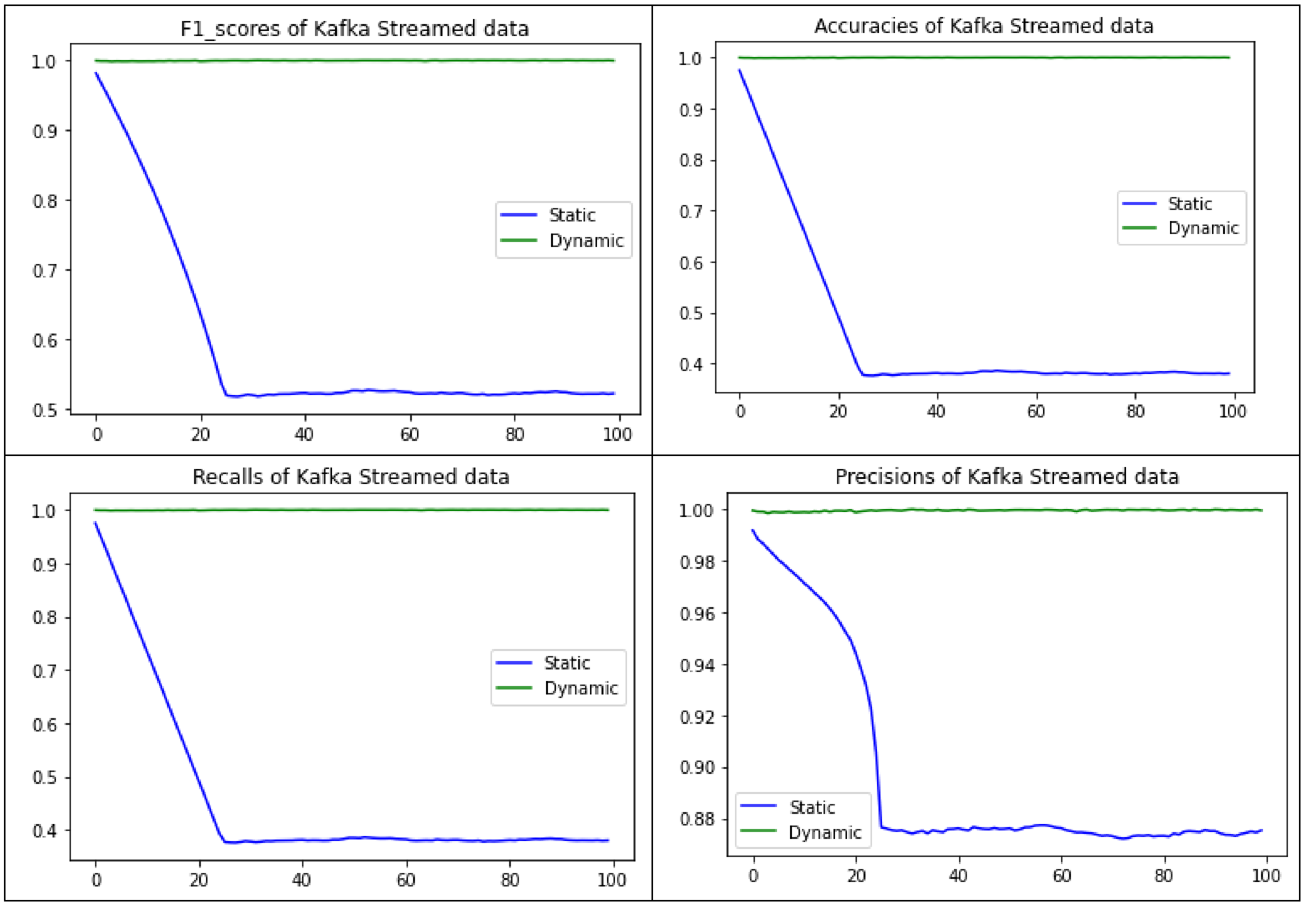
f1 score: 0.9992590737608429	Precision score: 0.99931486308258
Accuracy score: 0.9992580733335936	recall score: 0.9992580733335936

2) Results of Dynamic Model with Kafka Streams:

   - The below plots show the metrics of each model along all batches.



- The below plots show the comparison between the static and dynamic model with every evaluation metric one at a time.



Discussion:

- From All of the above plots and tables, it’s clear that adaptive or dynamic learning is very important especially with varying data like Kafka streams.
- Based on the results, the Static model performed very well with the original data in spite of the imbalance between the multiple classes.
- Based on the results and plots, the static model performed will with the initial data and started to downgrade the more new data is appended from the Kafka Streams until it’s exposed to totally new data that differs from the one it trained on.
- Here comes the strength and the advantage the of the dynamic model which stands still despite the varying data, it may take much more computations but eventually it will keep the good performance till the end.
- So, the only advantage of the static model is the less computations against the adaptive one.
- In multi class classification we may use different averaging method based on the nature of our data, so as for our case there is a remarkable imbalance so it’s preferable to use the weighted average method.

References:

https://builtin.com/data-science/random-forest-algorithm

https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

https://stackoverflow.com/questions/55740220/macro-vs-micro-vs-weighted-vs-samples-f1-score#:~:text=average%3Dmacro%20says%20the%20function,each%20label%20in%20the%20dataset.