uOttawa

Faculté de génie
Faculty of Engineering

Report Assignment 2

Ai For Cyber Security - ELG7186[EG]

Name: Eslam Ahmed Abdelrahman Mahmoud

Email: emahm025@uottawa.ca

ID: 300267020

1. Binary Problem

1.1. Algorithm’s description

after load dataset I applied simple preprocessing to drop NAN, Inf values and repair initial spaces in the column name.

1.1.1. Static Part

In Binary Problem in static part, I splitted given dataset to 80 %training and 20% testing part, I trayed 3 different algorithms with All default parameters and fixed “random state”:

A. **decision tree**
   with All default parameters and fixed “random state” get accuracy in unseen test set = 99.84%
B. **random forest**
   with All default parameters and fixed “random state” get accuracy in unseen test set = 99.76%
C. **SVM**
   with All default parameters and fixed “random state” get accuracy in unseen test set = 93.03%

1.1.2. Adapted Part

In Adapted Part I used forgetting Strategies (fixed window method) with 20,000 row window size and get new 2,000 records using Kafka API and loop for 50 times. Each iteration adds the new records in the bottom of my DataFrame and delete the same size for the top of DataFrame to be sure that after 10 iteration all old records was changed.

I choose decision tree Algorithm as a best performance in static part with following parameters

Decision Tree Model parameters in static and dynamic part in the two problems:

criterion='gini', **splitter**='best', **max_depth**=None, **min_samples_split**=2, **min_samples_leaf**=1, **min_weight_fraction_leaf**=0.0, **max_features**=None, **random_state**=None, **max_leaf_nodes**=None, **min_impurity_decrease**=0.0, **class_weight**=None, **ccp_alpha**=0.0

In this part I used two Models based on decision tree.

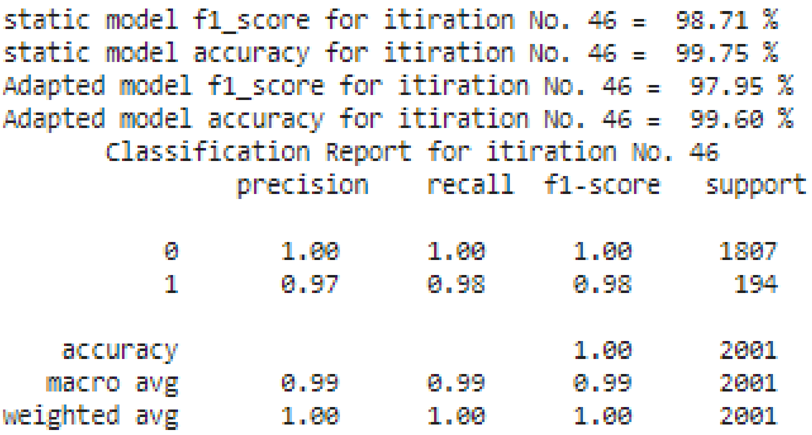
“static_model” that trained (fitted) on given dataset. and during iteration and sliding the fixed window predict each new data on it without fitting and append its value to list.

“Adapted_model” that trained (fitted) on the fixed window data 20,000 and train (fit) and during iteration and sliding the fixed window predict each new data and append its value to list. and train on the new window.

Finally, I get two prediction lists form the two models.

1.2. Algorithm evaluation

from the two prediction lists that I mentioned before I applied “Accuracy”, “F1_Score” as evaluation metric and print Classification report for each iteration. As shown in fig. 1



```
static model f1_score for itiration No. 46 =  98.71 %
static model accuracy for itiration No. 46 =  99.75 %
Adapted model f1_score for itiration No. 46 =  97.95 %
Adapted model accuracy for itiration No. 46 =  99.60 %
       Classification Report for itiration No. 46
                precision    recall  f1-score   support

           0       1.00      1.00      1.00      1807
           1       0.97      0.98      0.98       194

    accuracy                           1.00      2001
   macro avg       0.99      0.99      0.99      2001
weighted avg       1.00      1.00      1.00      2001
```

Figure**-Error! No text of specified style in document.**-1 terminal output of iteration no. 46 for problem 1

1.3. Results

After get Accuracy and F1_Score and classification report I think best way to show result to plot both line in same figure as shown in fig 2&3

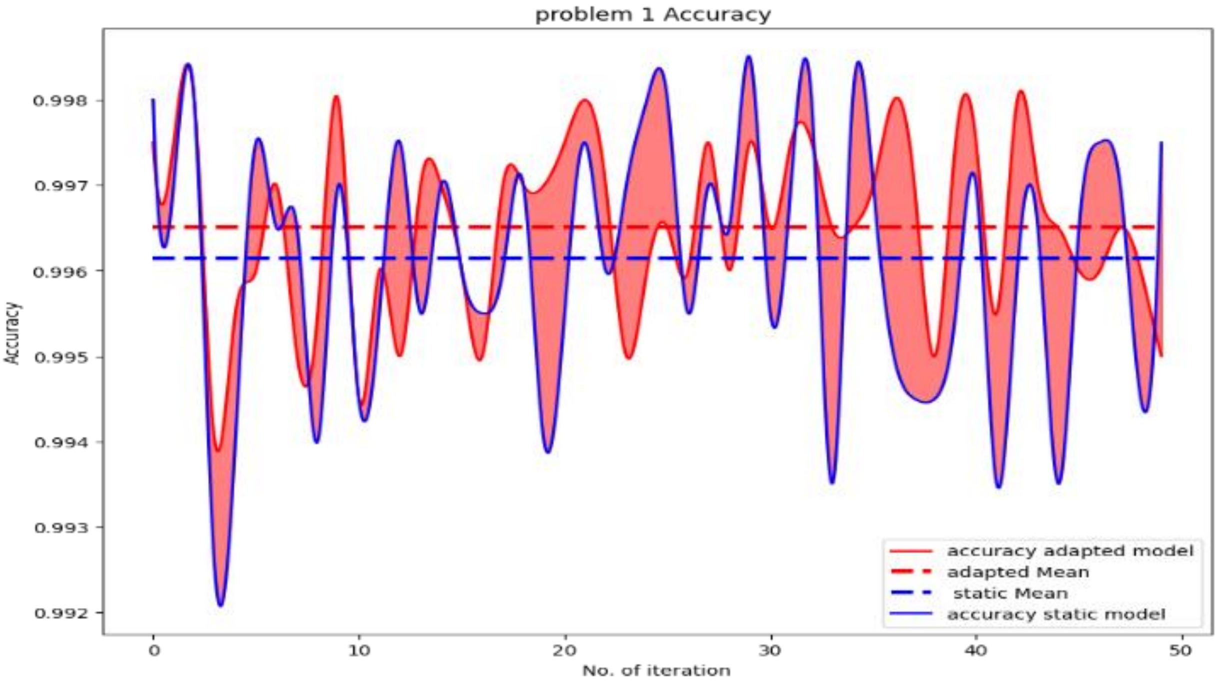


Figure 2 line chart between accuracy and no of iteration for problem 1

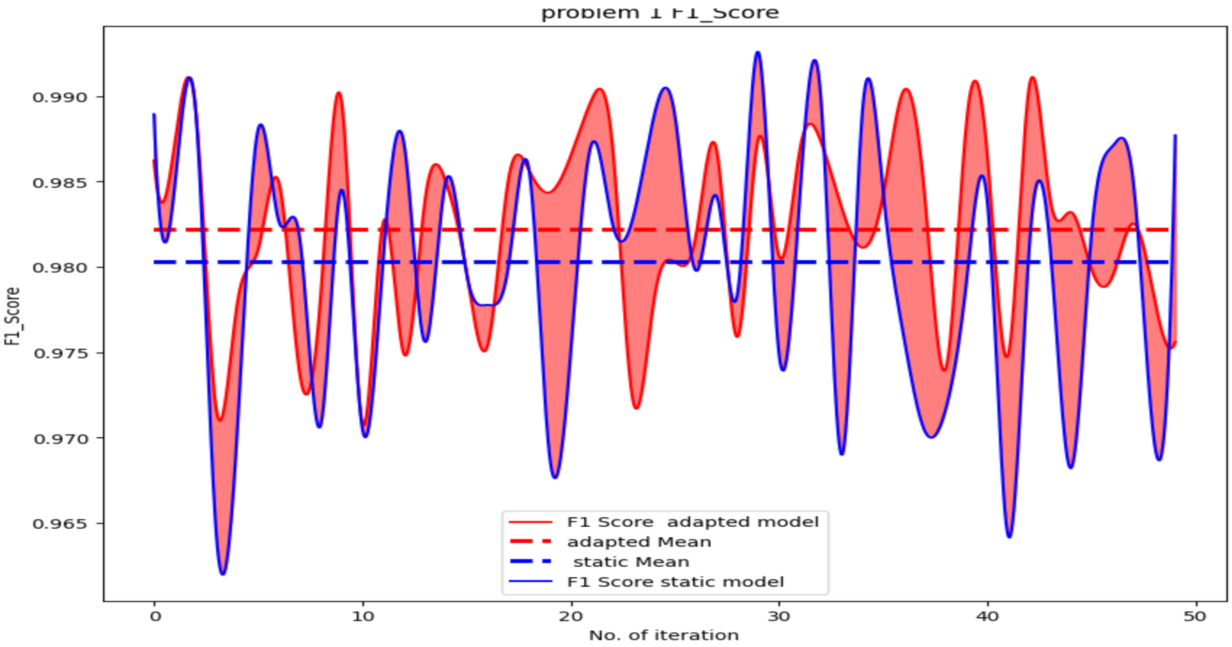


Figure 3-line chart between accuracy and no of iteration for problem 1

1.4. discussion

finally, after plot output charts I noticed adaptive Model get the best average performance and this shown in fig. 2&3 in dash lines. but it is small different between the two model it 0.2 % between averages.

I think this related to the given data is and train Static model on it have the same patterns stream data or refer to using fixed window strategy.

2. Multiclass Problem

2.1 Algorithm’s description

after load dataset I applied simple preprocessing to drop NAN and Inf values.

2.1.1 Static Part

In Multiclass Problem in static part, I splitted given dataset to 80 %training and 20 % testing part, I trayed 3 different algorithms with All default parameters and fixed “random state”:

D. **decision tree**
   with All default parameters and fixed “random state” get accuracy in unseen test set = 99. 90 %
E. **random forest**
   with All default parameters and fixed “random state” get accuracy in unseen test set = 99.88 %
F. **SVM**
   with All default parameters and fixed “random state” get accuracy in unseen test set = 87, 34 %

2.1.2 Adapted Part

In Adapted Part I used forgetting Strategies (fixed window method) with 20,000 row window size and get new 1,000 records using Kafka API and loop for 100 times. Each iteration adds the new records In the bottom of my DataFrame and delete the same size for the top of DataFrame to be sure that after 20 iteration all old records was changed.

I choose decision tree Algorithm as a best performance in static part. In this part I used two Models based on decision tree

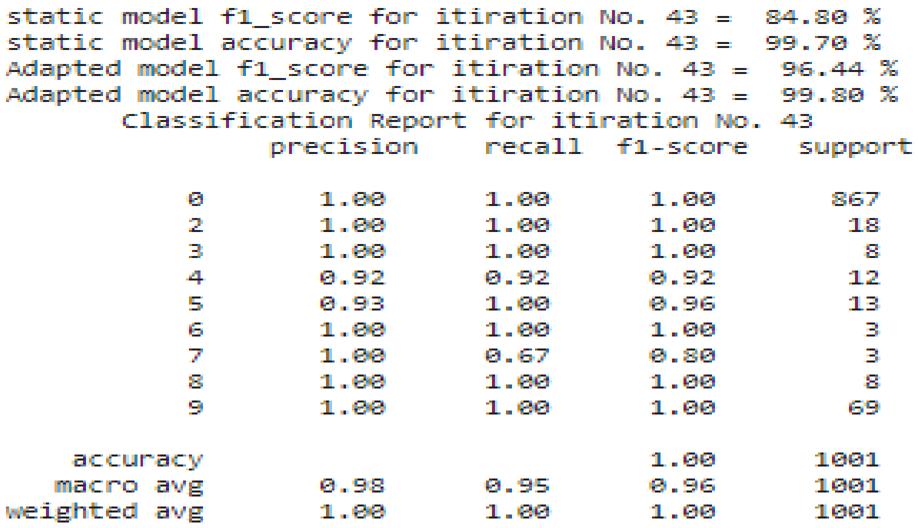
“static_model” that trained (fitted) on given dataset. and during iteration and sliding the fixed window predict each new data on it without fitting and append its value to list.

“Adapted_model” that trained (fitted) on the fixed window data 20,000 and train (fit) and during iteration and sliding the fixed window predict each new data and append its value to list. and train on the new window.

Finally, I get two prediction lists form the two models.

2.2 Algorithm evaluation

from the two prediction lists that I mentioned before I applied “Accuracy”, “F1_Score” as evaluation metric and print Classification report for each iteration. As shown in fig. 4



```
static model f1_score for itiration No. 43 =  84.80 %
static model accuracy for itiration No. 43 =  99.70 %
Adapted model f1_score for itiration No. 43 =  96.44 %
Adapted model accuracy for itiration No. 43 =  99.80 %
       Classification Report for itiration No. 43
                precision    recall  f1-score   support

           0       1.00      1.00      1.00       867
           2       1.00      1.00      1.00        18
           3       1.00      1.00      1.00         8
           4       0.92      0.92      0.92        12
           5       0.93      1.00      0.96        13
           6       1.00      1.00      1.00         3
           7       1.00      0.67      0.80         3
           8       1.00      1.00      1.00         8
           9       1.00      1.00      1.00        69

    accuracy                           1.00      1001
   macro avg       0.98      0.95      0.96      1001
weighted avg       1.00      1.00      1.00      1001
```

Figure- 4 terminal output of iteration no. 46 for problem 4

2.3 results

After get Accuracy and F1_Score and classification report I think best way to show final result to plot both line in same figure as shown in fig 5&6

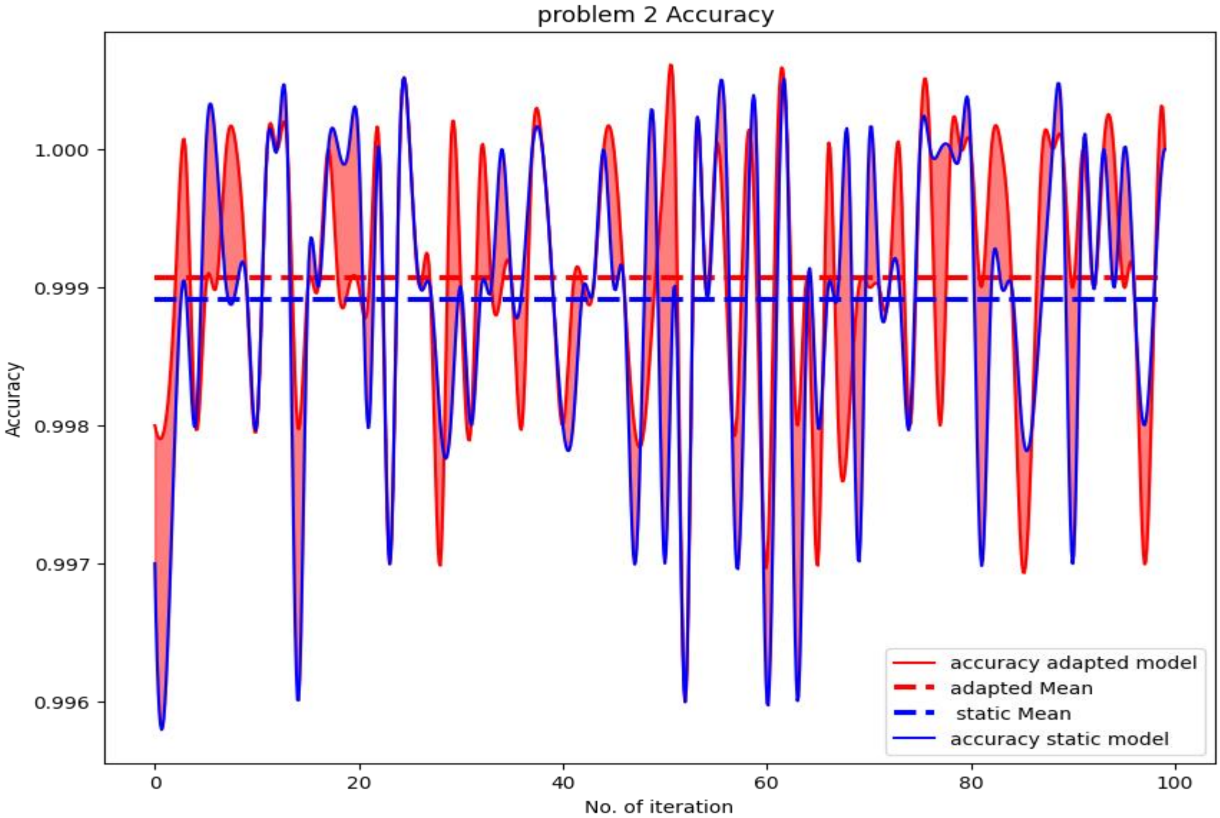


Figure 5-line chart between accuracy and no of iteration for problem 2

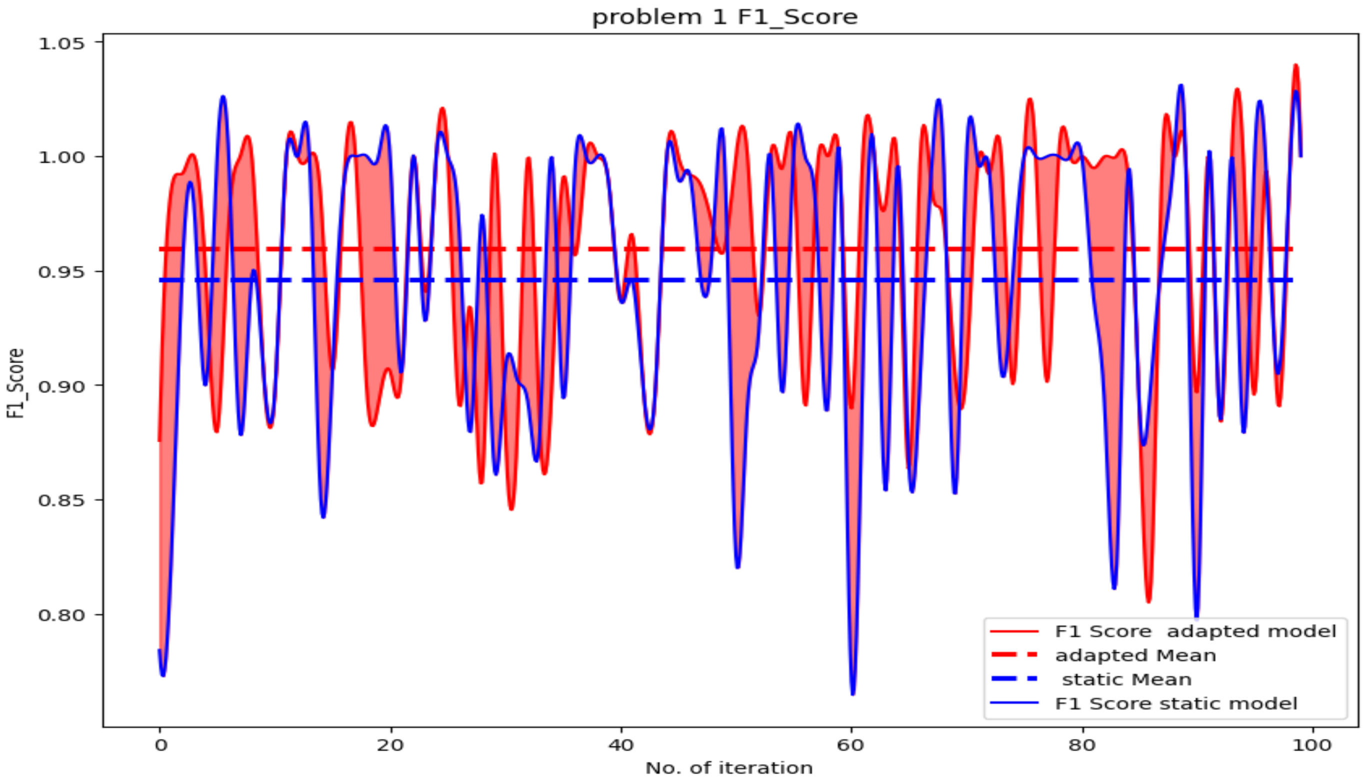


Figure 6 - line chart between accuracy and no of iteration for problem 2

2.4 discussion

finally, after plot output charts I noticed when traying and testing different size for window and streams packets when test on packets less than 1000 record we have some missing classes in output referred to imbalance packets and data set there may some classes didn't get on small packet size. So, I get to use 1000 record per iteration to make sure that will have all Class.

And the size of the adaptive Model gets the best average performance, and this shown in fig. 5&6 in dash lines. but it is small different between the two model it 0.2 % between averages. I think this related to the given data is and train Static model on it have the same patterns stream data or refer to using fixed window strategy. And imbalance Class.

limitations

Source in static and dynamic data is imbalance set I faced while traying when get data with small size some type of attacked didn't get so our dynamic model didn't fit well and this shown in the Classification report.

From charts all static and stream data have the same pattern so I didn't feel large different.