Kafka Assignment

Binary Problem

Overview

Network intrusion detection to check whether there is an attack or not using binary classification. In this section **CICIDS2017,** the dataset will be used. It contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs) [1].

Algorithms

This section will discuss the Two Different ensemble models that will be used.

Random forests

The random forests model Is an ensemble method that uses the bagging technique. It trains decision tree models with random subsample with replacement. Then it companies the predictions by average if it is a regression problem or by voting if it is a classification problem. [2]

AdaBoost

AdaBoost is boosting ensemble method. It tries to improve the asset of weak learners (decision tree). It is an improved decision tree by increasing the weight of misclassified samples of the previous decision tree. AdaBoost is suited for imbalanced datasets, but it is underperforming if there is noise in the data. [2]

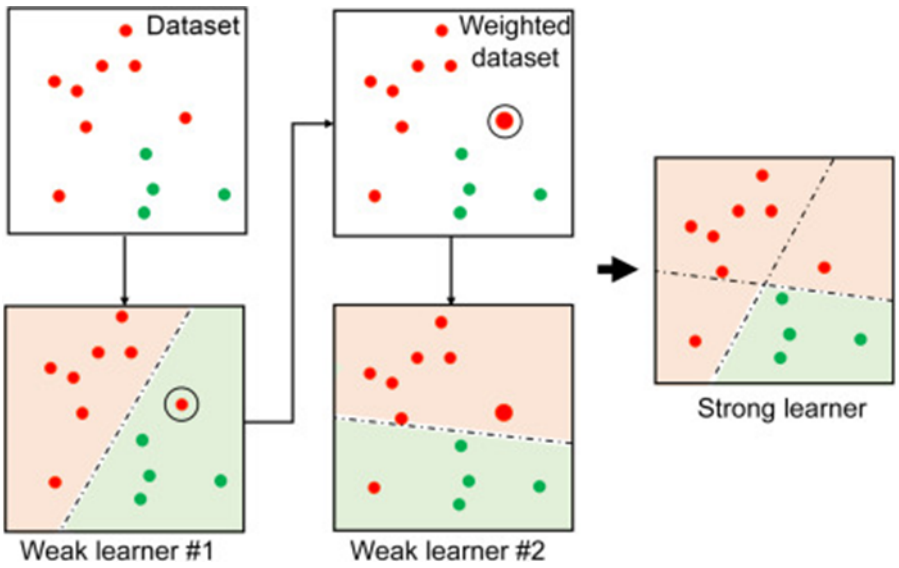

Fig. 1 AdaBoost has two weak learners, the weight of misclassified sample increases before being input for next learner then we can reach a strong learner.

Experiments

Static part

In this part, the models will be trained and evaluated using provided dataset [1]. Also, will decide which hyperparameters can increase the model performance.

Hyperparameters

Using grid search algorithm. The best hyperparameters score are shown in Table.1.

Random Forest		**AdaBoost**	
Name	**Value**	**Name**	**Value**
max_features	'auto'	learning_rate	1
min_samples_leaf	2	n_estimators	20
min_samples_split	2		
n_estimators	20		
max_depth	20		

Table.1. The hyperparameters used in the models Random Forest and AdaBoost after testing several parameters using grid search.

Evaluation Metrics used to evaluate the models are Precision, recall, and F1 Score.

	Precision	Recall	F1 Score
Random Forest	0.998	0.998	0.998
AdaBoost	0.994	0.994	0.994

Table.2 scores of the Two models after testing them on a subset of the given data set [1].

The Random Forest model is chosen to be used in the dynamic part because it is perform better as (table 2) shown.

Dynamic part

The dynamic part work in the following steps:

1. Every epoch the 1000 packets were retrieved from the Kafka server.
2. Both static and dynamic models were tested on the new 1000 packets.
3. New 1000 packets were added to the original data and the oldest 1000 samples in the dataset were deleted.
4. The dynamic model is retained on this new data.
5. The previous steps repeated until reaching 100,000 were retrieved from the server.

The F1 Score of the two models shown in Fig. 2

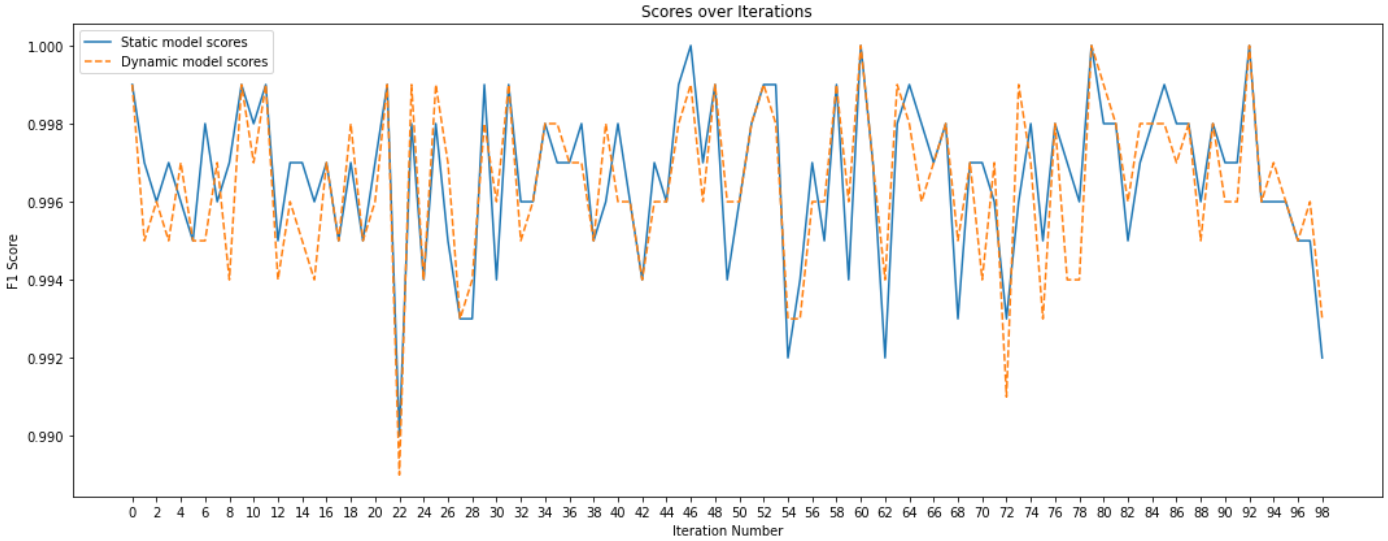

Fig 2. F1 scores of the two models along with the 100 iterations.

Evaluation Metrics used to evaluate the models are Precision, recall, and F1 Score.

	Precision	Recall	F1 Score
Random Forest	0.9966	0.9966	0.9966
AdaBoost	0.9964	0.9964	0.9966

Table.3 The average of all iteration's precision, recall, and F1 scores.

Conclusion:

The average f1 scores of the dynamic and static models are the same as (Table 3) shown. We see in (Fig. 2) that in some samples the static model performs better than the dynamic model as in iterations (1, 6, 12, 46, 64, 85). In this case, we can use other adaptation strategies so the model can learn without forgetting the other what it learned before because the data does not change so much over time as incremental learning.

Multiclass Problem

Overview

Network-based Detection of IoT Botnet Attacks to check whether traffic data are benign and Malicious (The malicious data can be divided into 10 attacks carried by 2 botnets) using Multiclass classification. In this section **detection_of_IoT_botnet_attacks_N_BaIoT** dataset will be used. gathered from 9 commercial IoT devices authentically infected by Mirai and BASHLITE. [4].

Experiments

Static part

In this part, the models will be trained and evaluated using provided dataset [4]. Also, will decide which hyperparameters can increase the model performance.

Hyperparameters

Using grid search algorithm. The best hyperparameters score are shown in Table.4.

Random Forest		**AdaBoost**	
Name	**Value**	**Name**	**Value**
max_features	'auto'	learning_rate	0.1
min_samples_leaf	2	n_estimators	20
min_samples_split	2		
n_estimators	10		
max_depth	10		

Tabel.4. The hyperparameters used in the models Random Forest and AdaBoost after testing several parameters using grid search.

Evaluation Metrics used are the average weighted Precision, recall, and F1 Score.

	Precision	Recall	F1 Score
Random Forest	0.999	0.999	0.999
AdaBoost	0.942	0.969	0.955

Table.5. Scores of the Two models after testing them on a subset of the given data set [4].

As (Table 5) shown Random Forest model performs better than AdaBoost, so the Random Forest model is used in dynamic.

Dynamic part

The dynamic part has the same steps as mentioned in the binary problem.

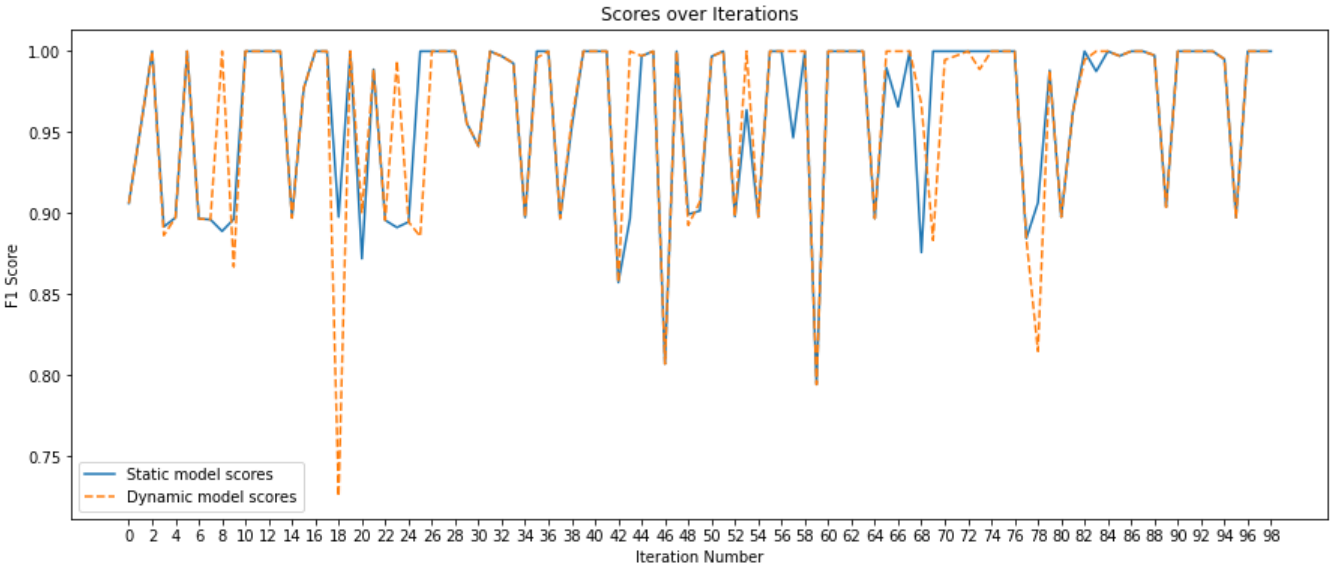

Fig 3. F1 scores of the two models along with the 100 iterations.

	Precision	Recall	F1 Score
Random Forest	0.9918	0.9645	0.9613
AdaBoost	0.9931	0.9641	0.9616

Table.6. The average of all iterations' precision, recall, and F1 scores.

Conclusion:

The average f1 scores of the dynamic and static models have almost the same value as shown in (Table 6). We see (in Fig. 3) some in some samples the dynamic model performance dropped as in iterations (18, 78). I think in these iterations some of the classes have vanished from the data set or the data set become more imbalanced in this case we can use other adaptation strategies so the model can learn without forgetting the other what it learned before because the data does not change so much over the time as incremental learning.

References

[1] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal.

[2] Misra, S., & Li, H. (2020). Chapter 9 - Noninvasive fracture characterization based on the classification of sonic wave travel times. In S. Misra, H. Li, & J. He (Eds.), *Machine Learning for Subsurface Characterization* (pp. 243–287). Gulf Professional Publishing. https://doi.org/https://doi.org/10.1016/B978-0-12-817736-5.00009-0

[3] Mason, L., Bartlett, P., Baxter, J., & Frean, M. (1999). *Boosting Algorithms as Gradient Descent*.

[4] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici 'N-BaIoT. (2018). Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders', IEEE Pervasive Computing, Special Issue - Securing the IoT.