**Evaluating Model and Analysing the Results – Andrew Robinson**

**Evaluation Overview**

After selecting and refining numerous algorithms three final models were selected as the focus of the evaluation. Pipelines were set up for each model before final refinements were made. Each model was trained using a version of the, previously mentioned, AWID dataset which was balanced to show equal numbers of attack and non-attack instances. The models were then evaluated using a testing dataset which was subjected to the same modifications and was made up of 40,158 observations (20,079 normal traffic (non-attacks) and 20,079 impersonation attacks).

In order to give a full overview of the performance of each model a number of performance measures have been calculated. The majority of these measures involve calculating the: True Positive (TP) values which show intrusions correctly identified as an attacks; True Negative (TN) values which show normal instances correctly identified as non-attack; False Positive (FP) values or normal instances incorrectly identified as an attack; and False Negative (FN) values showing Intrusions incorrectly identified as non-attack. These values were obtained from the classification report run on each pipeline.

A brief summary of each performance measure is outlined below:

In addition to the above measures Time to Build (TTB) in seconds will also be calculated for each model. In order to maintain a level playing field, all models will be run in the same environment using a mac OS with 1.4 GHz Intel i5 CPU and 8 Gb of RAM. Furthermore additional information on each performance measure above can be found in the DEMISe Techniques for IoT Intrusion Detection paper [2] should it be needed.

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pipeline 1 | Pipeline 2 | Pipeline 3 |
| Acc (%) | 99.08 | 99.08 | 98.63 |
| DR (Recall) (%) | 98.15 | 98.15 | 97.96 |
| Precision (%) | 100 | 100 | 99.29 |
| FAR (%) | 0 | 0 | 0.01 |
| FNR (%) | 1.85 | 1.85 | 2.04 |
| F1 (%) | 99.07 | 99.07 | 98.62 |
| MCC (%) | 98.17 | 98.17 | 97.26 |
| TTB (seconds) | 7.24 | 7.4 | 5.57 |

**Evaluation – Pipeline 1**

Pipeline 1 is made up of the following:

* Variance Threshold is used to remove variables with zero variance.
* Fclassif for feature selection.
* Adaboost as the algorithm.

The parameters of each element of the model were tuned using grid search.

As can be seen from the above results table Adaboost performed much better than logistic regression on all measures except time to build.

**Evaluation – Pipeline 2**

Pipeline 2 is made up of the same elements as Pipeline 1 with one addition, a Min/Max scaler is added to scale every variable between 0 and 1.

There is no impact of the scaler on performance other than to increase the time to build the model.

**Evaluation – Pipeline 3**

Pipeline 3 is made up of:

* Variance Threshold is used to remove variables with zero variance.
* Min/Max Scaler to scale variables between 0 and 1.
* Chi Square for feature selection.
* Logistic regression for the algorithm.

Whilst not performing at the same level as the Adaboost model. Logistic Regression still performed well on our test data. In addition, the time to build for a Logistic Regression was over 1.8 seconds faster the Adaboost model.

**Conclusion**

Following experimenting with multiple different algorithms it was clear that the Adaboost model performed significantly better than the other models that were trained on the AWID dataset. This was backed up when the final three models were used on the test data. Therefore, the pipeline 1 model is chosen as our best performing model for this project.

**References Evaluation**

[1] M. Aminanto, R. Choi, H. Tanuwidjaja, P. Yoo, K. Kim, "Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection", in IEEE Transactions on Information Forensics and Security, vol. 13, no. 3, March 2018.

[2] L. Parker, P. Yoo, T. Asyhari, L. Chermak, Y.Jhi, K.Taha, “DEMISe: Interpretable Deep Extraction and Mutual Information Selection Techniques for IoT Intrusion Detection” ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and Security, Article No. 98, August 2019.