**Exploring and selecting ML algorithms**

**Introduction**

After having preprocessed the dataset, used feature selection to reduce our dataset, categorised the problem and understood the data, we identified the algorithms that are applicable and practical to implement. Furthermore, before having set out to select the correct algorithm we chose the following 3 measurements to be the bases of our evaluation and point of comparison of other algorithms:

1. The accuracy of the model.
2. Time taken to make predictions using the model
3. Time taken to build, train, and test the model

**ML Algorithms**

Having set the groundwork with the phase1 and 2 the algorithm we decided to use as a baseline for our machine learning algorithm phase, was Xgboost, which in the paper [1] outline the benefits over many other machine learning models.

Xgboost is a greedy gradient boosting algorithm, which is a decision-tree ensemble method that performs well on structured data. A common issue that arises with tree learning is finding the best split that leads to optimal results indicated by equation in [2]. Xgboost tackle this problem by using a depth-first approach, which is a tree pruning method that has shown to significantly increase performance whilst reducing the time complexity of the algorithm [2].

The following algorithms Catboost, logistic regression, naïve Bayes, Random Forest and decision tree classifier (Adaboost) are the five extra algorithms we decided test.

Catboost is a unbiased gradient boosting algorithm having shown to outperform many of the latest and best algorithms [3], this was a step up from Xgboost, however, it took a significant amount longer for making predictions. Logistic regression, Random Forest, and naïve Bayes have previously been proven to work well with AWID dataset, as shown in the paper by Dr Paul and his counterparts [4] where tree learning and simple algorithms like logistic regression proved to be both interpretable and predict with high accuracy

**Classifier**

Boosting is generally a technique that derives from Probably Approximately Correct (PAC) models – a theoretical learning framework with the aim to show that an algorithm achieves low empirical risk (ER) error with high probability [6]. However, for this model we are adopting a more general learning framework where we use Adaboost – an ensemble boosting algorithm created by Yoav Freund and Robert Schapire in 1996 [5] to tackle the problem of overfitting, increase accuracy through combination of weak learners and an algorithm which is simple to implement.

The formula for classifying with Adaboost is the following:

A picture containing object

Description automatically generated

Eq1

Where f(x) is the function representing the mth classifier and theta is the mth corresponding weight. Thus, this formula is the combination of all the weights of all mth classifiers (ensemble method). Adaboost trains the classifiers on the training data and then places weights on each case depending on how accurate the predictions were (the larger weight are placed on misclassified ones). Lastly, the final classifier is made up of linear combination of each iteration (different classifiers at each stage), which uses an average to reduce the algorithm to overfit the data [6].

The best performing classifiers were Adaboost and logistic regression, with the former proving to outperform the latter. The reason for this is two folds: with the first and obvious attribute of adaboost is the ability to adjust to non-linearity within the relationship of the data. Secondly, adaboost uses a forward stagewise additive logistic regression model, which allows it minimize the exponential loss and increase predictive performance [6]. This seems to fit well with Breiman’s [1996] remarks during a NIPS workshop where he said Adaboost is “the best-of-the-shelf classifier in the world” [6]. In conclusion, after thorough investigation the Adaboost classifier was chosen as a the classifier for our model, which is evaluated in the evaluation section.