Project Report

**Machine learning and DDos attack detection**

Badr Elhattab Elibrarhimi

**Introduction**

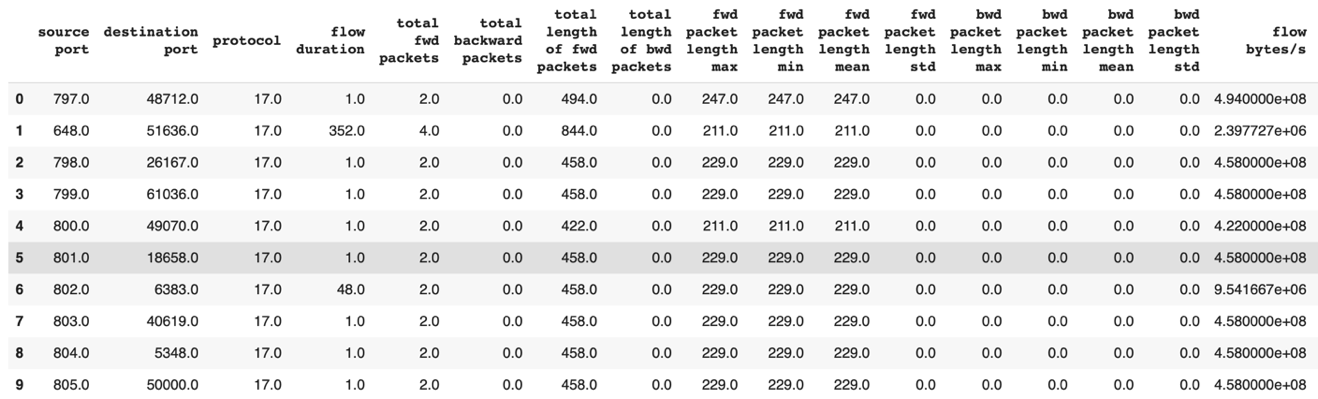
By carrying out a DDOS attack also called a "denial of service attack", the attackers aim to flood servers or computer systems with a large number of requests until the latter make their services and websites unavailable.

And with the rapid development of computer and communication technologies, the damage caused by DDoS attacks is becoming more and more serious. Of course, a lot of research has been done and progress has been made. However, due to the diversity of DDoS attack modes and the varying size of attack traffic, there is not yet a detection method with satisfactory accuracy.

During this project we focused on machine learning techniques to detect DDoS attacks in network communication streams using various learning algorithms that learn the normal pattern of network traffic, behavior of network protocols and identifies compromised network flows.

**Data set**

To be able to use our machine learning methods we first needed a set of data on which we could apply our learning models, and for that we used the CICDDoS2019 which is a set of data proposed by the Canadian Institute of Cyber Security (based at the University of New Brunswick) which contains a mixture of benign (normal) traffic and various DDoS attacks. The dataset includes network traffic captures analyzed using the CICFlowMeter tool with tagged flows based on timestamp, source and destination IP addresses, source and destination ports, protocols, and type of attack, all in a csv file.



**Figure # 1**: Overview of the dataset used (does not contain all attributes)

Given the large size of the dataset which is about 25GB, some concern was encountered with the RAM of the system used (Google Colab) and so I decided to divide the dataset into 7 subsets, apply my experience to each of these subsets and finally get my real results by calculating the average of the results obtained on each subset.

**Experimentation plan**

I did two types of experiments the first is a binary classification where my model was just trying to identify if an attack took place or not and so I had 2 classes, benign (which means no attack) and non\_benign (which means that there has been an attack).

In my 2nd type of experiment the classification was multi-class, once again I had a benign class (i.e. no attack), but also several other classes which said that there was an attack and specified what kind of DDoS attack it was, for example if it was (MSSQL, Portmap, UDP, UDPLag, ect ...)

Each of my experiments consisted of 3 steps: the first is the feature selection, the second is the classification using various supervised classification algorithms, and the third is the evaluation of these same algorithms using different metrics.

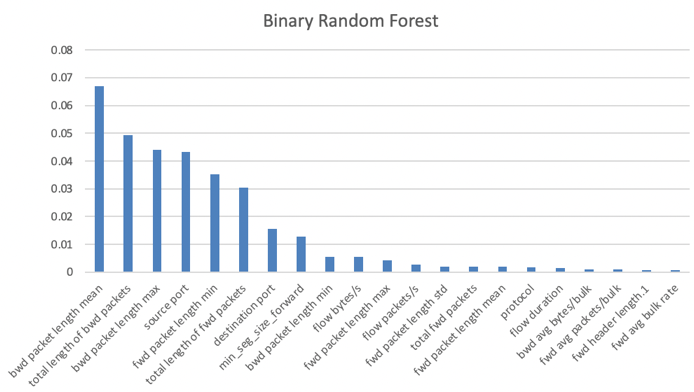
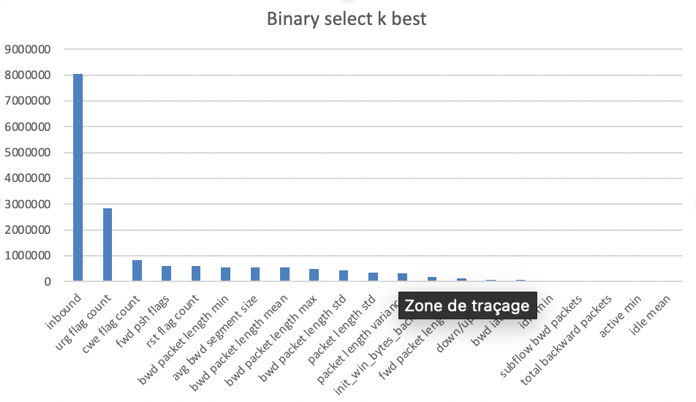
**First step : Features selection**

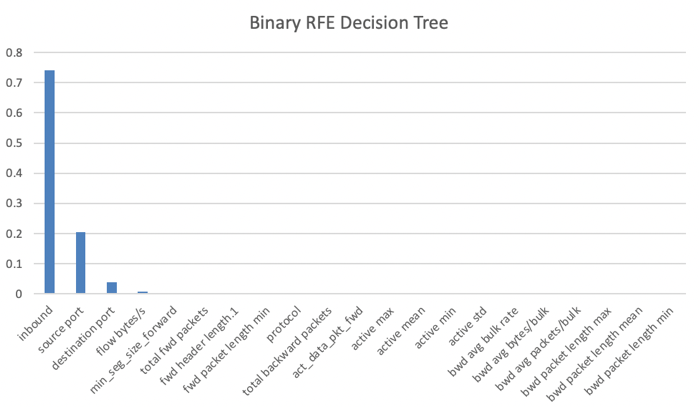
The attributes we use to train our machine learning models have a huge influence on the performance we can achieve. And therefore the use of irrelevant or even partially relevant attributes can have a negative impact on the performance of the model.

The attribute selection process helps us avoid these kinds of issues by selecting the attributes that contribute the most to our predictor or output that interests us. The latter can also bring many benefits when modeling our data such as: reduction of overfitting, improvement of precision, and reduction of learning time.

Regarding the methods used for binary or multi-class classification, we tried three methods which are: the select k best; the random forest; and the rfe (recuresive feature eleminator) with the decision tree as estimator.

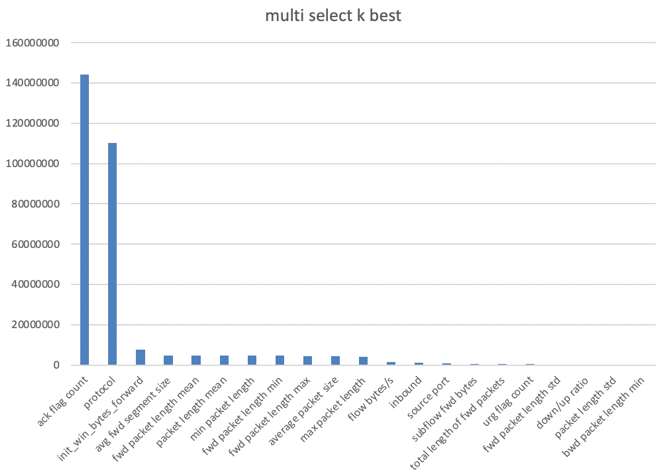
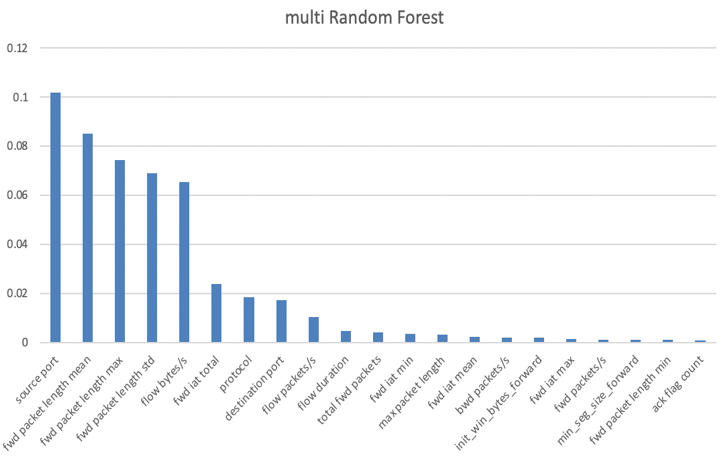
**Results of features selection (Binary classification)**

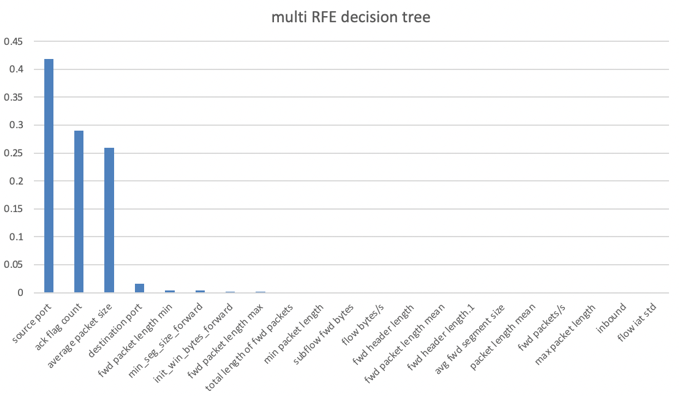
****

****

Here are the results I got when selecting attribute for the binary experiment, my dataset had 88 attributes, and I decided to keep the best 40 using the different methods mentioned, and even after doing that it can be seen that only a small portion of the attribute has a significant score. (Note: in the graphs only 20 attributes are represented)

**Results of features selection (multiclass classification)**

** **

****

Here are the results obtained when selecting attribute for the multi-class experiment, where again we can see that only a handful of attributes obtained a significant and dominant score compared to the remaining attributes which had almost a null score.

After carrying out preliminary tests on a sample of data, I decided to use the attributes selected by the random forest method and apply them to the second step which is classification.

But in future research, the experiments performed should also be extended to other methods of selecting attributes to see if there is a difference in performance.

**Second step : Classification**

For the binary classification we have divided our data set into a training set and an evaluation set by applying the K fold cross validation method.

We have applied this method on three binary classification algorithms which are logistic regression, vector machine support, and random forest.

Regarding the multi-class classification, we have again divided our data set into a training set and an evaluation set, but this time using 2 different methods. The first is the K fold cross validation that we have already mentioned, and the second is the stratified K fold cross validation, this is a slight variation of the K fold cross validation technique, so that each fold contains approximately the same percentage of samples from each target class as the full set.

we have applied each of these 2 methods on 3 multi-class classification algorithms which are the Decision tree, the random forest, and the naive bayes.

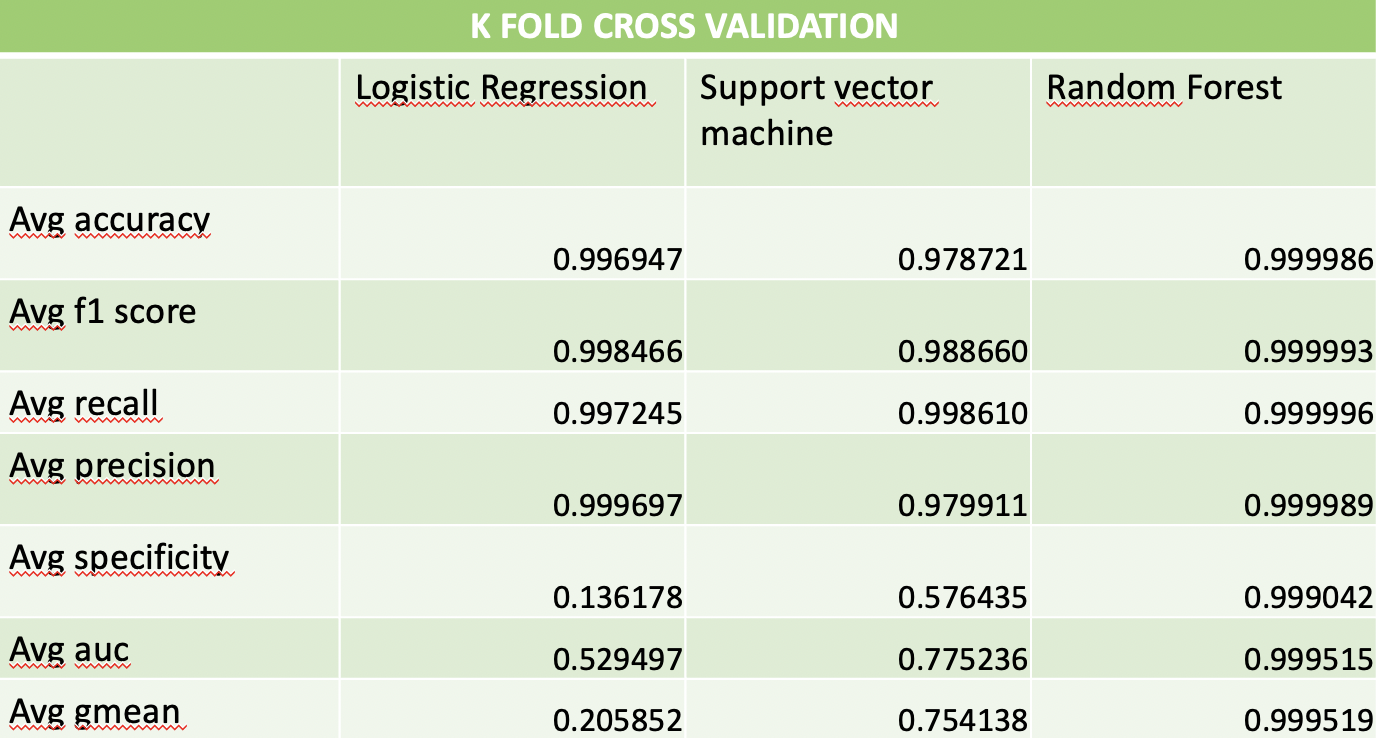
**Third step: Evaluation of results**

To evaluate our models, different metrics were used for each type of classification.

For the binary classification we used the accuracy, the F1 score, recall, precision, specificity, the area under the curve, and the geometric mean.

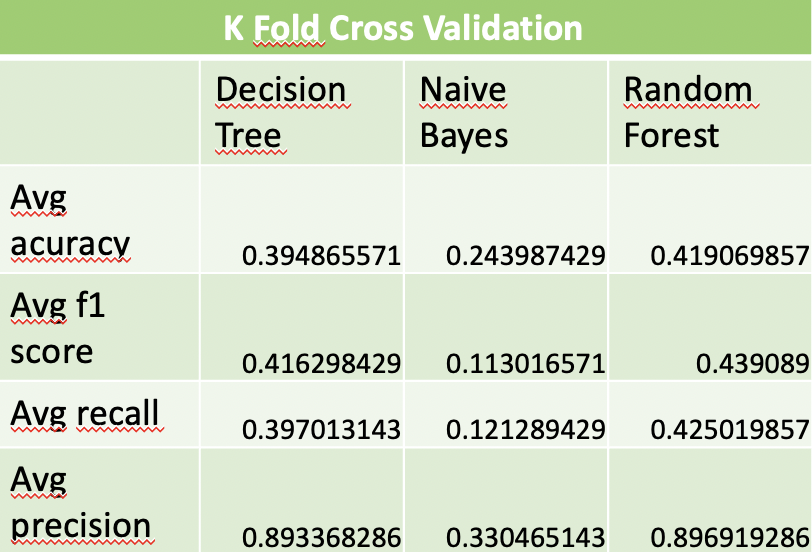
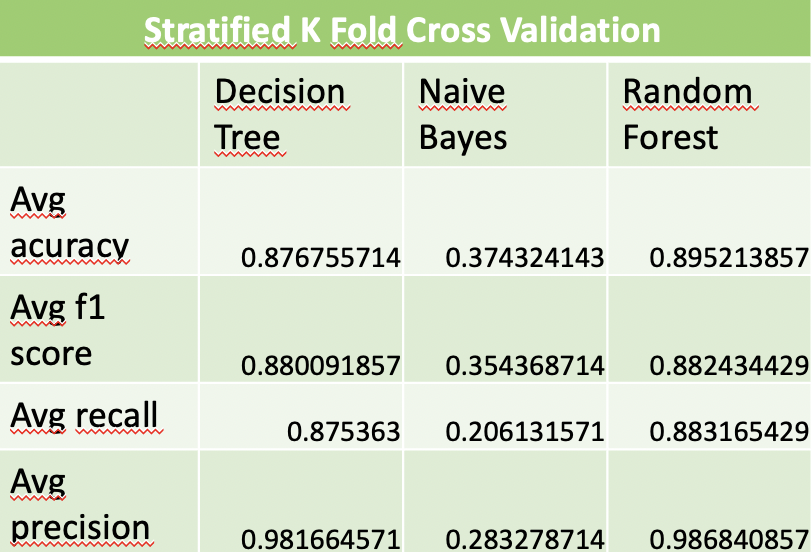
And for the multi-class classification we just used the accuracy, recall, precision, and again the F1 score.

**Evaluation of results (Binary classification)**

****

Here are the results obtained during the evaluation for the binary classification, we can see that the random forest is the one that obtained the best score (almost a perfect score on all the metrics), while the linear regression had scores relatively low in the area under the curve and the geometric mean.

**Evaluation of results (multiclass classification)**

Regarding the evaluation results for the multi-class classification, we notice that the stratified k fold cross validation method has greatly improved the scores of our models in comparison with the k fold cross validation, which means that the folds of the latter contained an unbalanced percentage of samples from each target class.

We can also notice that once again the random forest algorithm is the one that performed the best, there is also the decision tree algorithm which also had very satisfactory results (almost as good as the random forest), on the other hand, the naive Bayes, unfortunately, did not perform that well.

**Conclusion**

To conclude, we can see that different methods and algorithms were used in order to solve our attack detection problem. So for the binary classification we saw that the best algorithm used was the random forest combined with the k fold cross validation method. While for multi-class classification we can either use the random forest or the decision tree combined with the stratified k fold cross validation method.

During the development of this project, the biggest challenge was the management of our data set which contained a very large amount of data, we had to find a safe and efficient way to apply our different methods and algorithms on this set while avoiding going beyond the RAM the system uses (google colab).

Regarding the next steps of this project, we can make our detection system a module or an application, which can be permanently supplied with data. The trained model can trigger an alarm to the network administrator to react when an attack is detected. We can also improve our system to ensure that attacks are detected as early as possible.