Detecting DDoS attacks using machine learning.

Background

In any IP network infrastructure securing the data is paramount and we have seen many instances of DDoS attacks that go unnoticed by the monitoring tools or by the IT staff. By the time we realize DDoS has impacted the network, it becomes too late and data is already compromised.

It is very important to identify DDoS attacks and take actions to prevent/block those attacks so that user data from any services can be protected by the organization

This project aims to predict and detect DDoS attacks based on existing labeled data from sflow using various supervised machine learning models. The accuracy of various Machine Learning (rest of the documents will reference as ML) models are compared and the most accurate model for a given data set is used to predict and plot DDoS traffic.

Implementation

The app.py implements 4 different ML algorithms such as

1. Logistic Regression,
2. Logistic Regression with Correlation matrix
3. Naive Bayes,
4. KNN Classifier
5. Random_Forest.

The data set used for ML is obtained from this publicly available information. Here are the properties of this data set.

1) 225K + rows
2) 80+ features
3) 1 class label

Total of 80 features are used for ML training. The data here is a labeled data and supervised learning algorithms will be used. This data demands for binary classification i.e. DDoS vs Benign traffic identification.

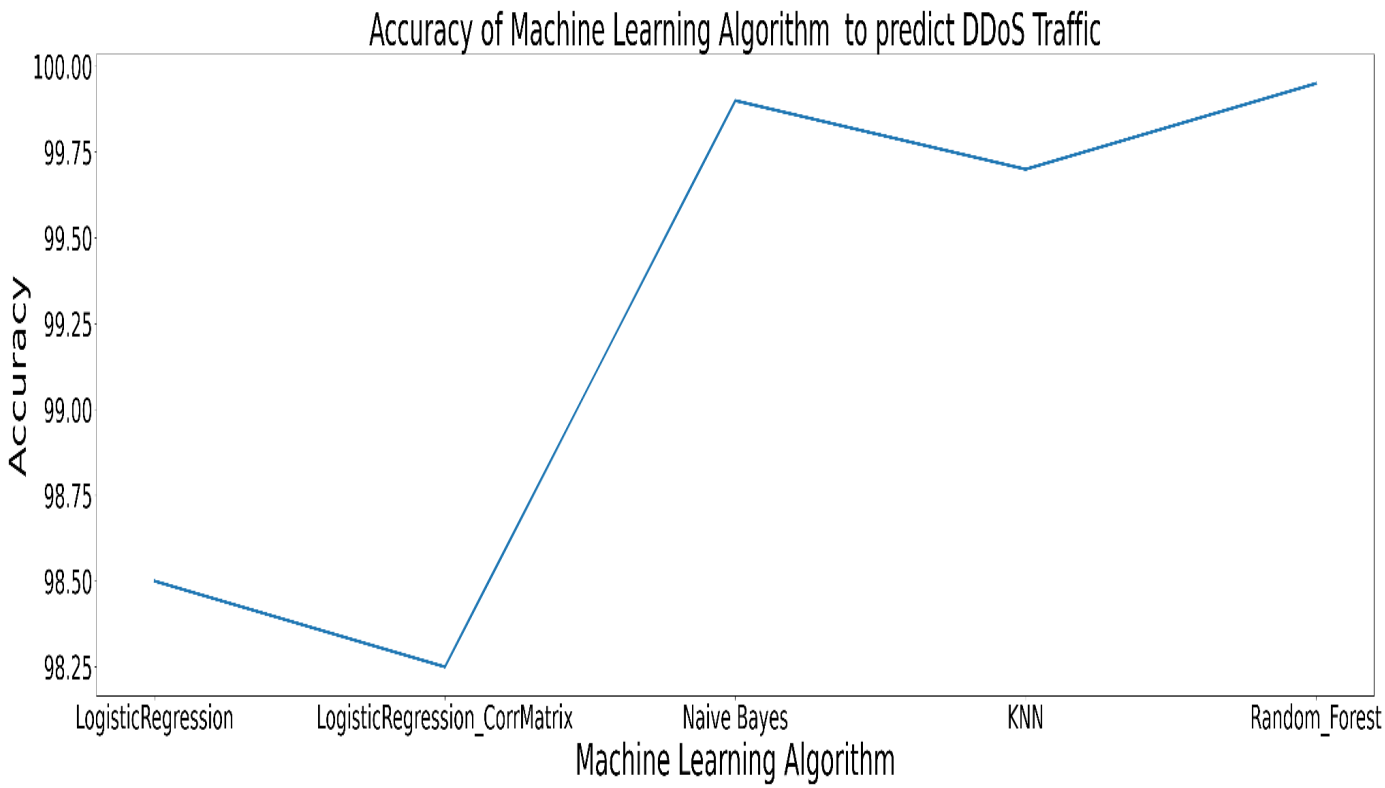
First the data is pre-processed using a separate function that takes csv data as input and selects 2000 flow records (1000 for Benign and 1000 for DDoS). The flow records are randomly distributed such that when the data is split using train_test_split function at 50%, the training data gets to see the pattern for both DDoS and Benign flows.

This way the model training is optimal and the performance of the model after training is validated using the test data. The trained object of a given model can be again run against the rest of the 200K+ data to further check the performance of these different models against validation data.

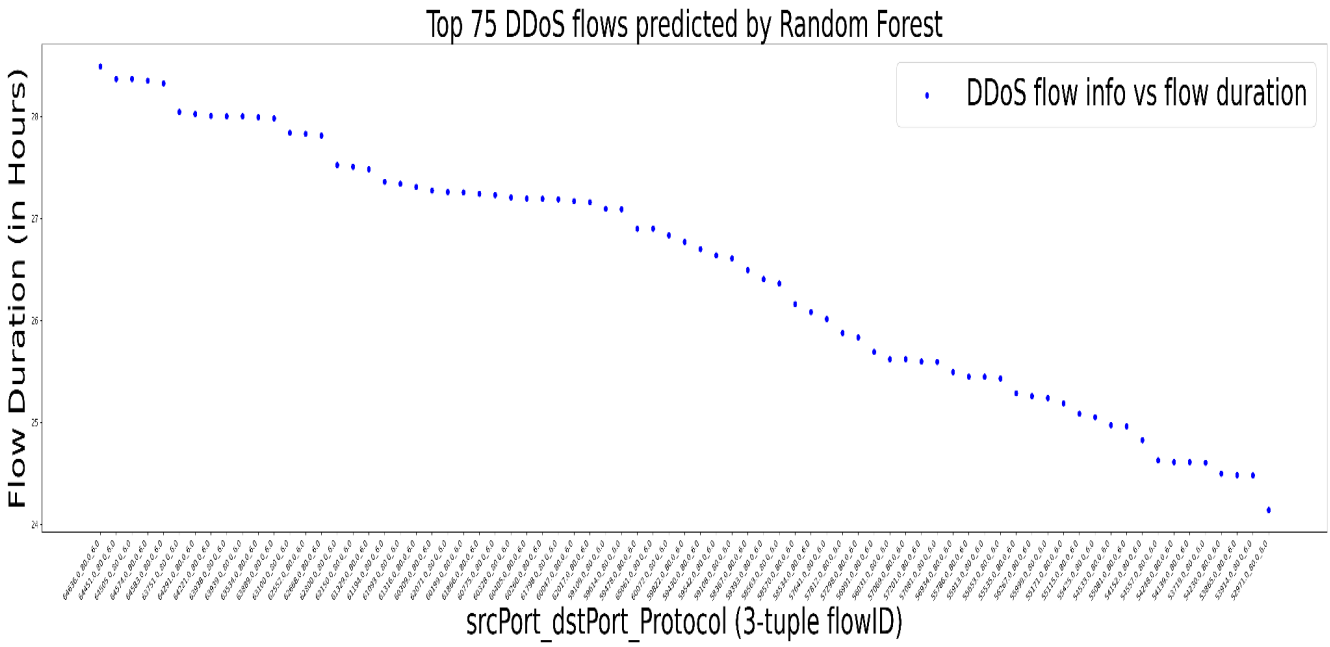
The end goal of this project is such that in a given network, when traffic from the edge nodes are sampled and streamed to the sflow collector, the collector runs these trained ML models and make binary classification if DDoS attack is active in the network. The Sflow collector can further talk to other traffic engineering systems within the network to take action based on the detected DDoS traffic pattern. The actions can include blocking the source/dest IP pair on the edge, closing the dst port on the target machine which is underattack if possible or implement IP table rules on the target machine to block these DDoS attacks.

Visualization

Here are two graphs for this project, graph 1 compares the overall performance of various ML model against the given data set used for training.



Graph 2 shows the predicted DDoS traffic flow sorted based on the flow that had the longest flow duration i.e. the top 75 DDoS flows that had the highest attack time in the network.



Results

Here are the results from running 5 different instances of ML algorithms.

#1) Logistic Regression

Total features used for training is: 80
Accuracy from Logistic regression model is: 98.50%
True Positive Rate is (Rate of DDoS flows predicted as DDoS by the model) : 100.0
False Positive Rate is (Rate of Benign flows predicted as DDoS by the model) : 3.0

%%% Dropping high positive Correlated features

Total features used for training is: 39
Accuracy from Logistic regression model is: 98.25%
True Positive Rate is (Rate of DDoS flows predicted as DDoS by the model) : 100.0
False Positive Rate is (Rate of Benign flows predicted as DDoS by the model) : 3.0

#2) Naive Bayes

Total features used for training is: 80
Total Accuracy of the NB Decision Tree Classifier is: 99.65%
True Positive Rate is (Rate of DDoS flows predicted as DDoS by the model) : 100.0
False Positive Rate is (Rate of Benign flows predicted as DDoS by the model) : 1.0

#3) KNN Classifier

Total features used for training is: 80
The best accuracy seen with KNN=2 is 99.7%
True Positive Rate is (Rate of DDoS flows predicted as DDoS by the model) : 100.0
False Positive Rate is (Rate of Benign flows predicted as DDoS by the model) : 2.0

#4) Random_Forest

Total features used for training is: 80
The best accuracy seen with Random Forest using estimator 8 and max depth 5: 99.8%
True Positive Rate is (Rate of DDoS flows predicted as DDoS by the model) : 100.0
False Positive Rate is (Rate of Benign flows predicted as DDoS by the model) : 0.0

ML Model	Features used	Accuracy Rate	TPR	FPR
Logistic Regression	80	98.50	100	3
Logistic Regression with Correlation Matrix	39	98.25	100	3
Naive Bayes	80	99.65	100	1
KNN Classifier (KNN=2)	80	99.7	100	2
Random_Forest (estimator =8, max depth = 5)	80	99.8	100	0

Conclusion

From this project it is evident that the ML model performance is improved when the data is scaled using simplescalar before being fit into the model. For instance logistic regression with 80 features was giving accuracy of 92%, but post scaling the accuracy increased to 98%.

The correlation matrix is in this picture to see if there are improvements in ML model accuracy when highly correlated features are removed. Before data scaling, removing highly correlated features did provide some level of improvement in the accuracy.

Finally KNN and random forest gives the best accuracy because we are able to try with different hyper parameters for both these models and we get good distribution of accuracy to compare against. The winning ML model is random forest with estimator =8 and depth = 5. This is also proven in previous studies where random forest algorithm works well for prediction DDoS attack patterns.