TOPIC:

APPLY MACHINE LEARNING TO IDENTIFY ABNORMAL NETWORK TRAFFIC

Advisor: PhD. Cao Văn Lợi

Team member:

1. Lưu Minh Quân
2. Trần Hoàng Vũ
3. Trần Tấn Thành

Hanoi, 12-2020

Table of Contents

CHAPTER 1

INTRODUCTION

1.1. Problem statements

Cybersecurity is developing and the rate of crime on the Internet is increasing. This constant evolution also calls for innovation in the cybersecurity defense.

Network Intrusion Detection and Prevention Systems (IDS/ IPS) are widely used for malicious activity or policy violations. Signature-based IDS depend on known patterns and work well at detecting malwares that fit known signatures, while Behaviour-based IDS learns what is normal for a system and reports on any different traffic. However, both types have some weaknesses. Signature-based systems rely on known threats and are consequently ineffective for zero-day attacks.

Machine Learning has gained wide attention on many applications. Both of the Machine Learning approach, Supervised and Unsupervised Learning, can be applied in Cybersecurity for analyzing malware in near real-time, thus eliminating the weaknesses of traditional detection methods

Our approach uses NetFlow data for analysis. NetFlow records provide enough information to identify traffic using 15 attributes

1.2. Objectives

The main objective of our project is to detect botnet traffic based on NetFlow dataset using various Machine Learning approaches. In specific, our proposal attempts to:

- Take any NetFlow dataset, then classify either normal or attack traffic
- Compare popular Machine Learning methods to figure out the proper model for a specific use case

CHAPTER 2

REVIEW OF BACKGROUND

2.1. NetFlow

NetFlow is a feature developed by Cisco that characterizes network operations. Network devices can be used to collect IP traffic information on interfaces where NetFlow is enabled. NetFlow data can be utilized to (1) identify network traffic, and (2) detect network anomalies. It used to identify devices that are finding bottlenecks, and improving network efficiency, There is also a large number of threat and anomaly detection tools that use NetFlow traffic
Traffic Flow data
Flows are defined as sequences of packets with some common properties. By investigating flows, we can find meaningful traffic patterns. The output NetFlow data has the following attributes:

- StartTime: the start time of the recorded flow
- Dur: duration of the flow
- Proto: protocol used (TCP, UDP, etc)
- SrcAddr: Source IP address
- Sport: Source Port
- Dir: Direction of communication
- DstAddr: Destination Address
- Dport: Destination Port
- State: the Protocol state
- sTos: source Type of Service
- dTos: destination Type of Service
- TotPkts: total number of packets exchanged
- TotBytes: total bytes exchanged
- SrcBytes: number of bytes sent by source
- Label: label assigned to this netflow

2.2. NetFlow for Anomaly Detection

Traditional anomaly detection methods such as intrusion detection require patterns published by manufacturers. NetFlow data does not contain private information and is widely used by network operators. As a consequence, NetFlow has been a popular source of information for anomaly detection. On the flip side, NetFlow has to do with volume of data and thus have an adverse effect on the performance and the amount of processing to be handled.

2.3. Main issue with Network Security Data

There are a number of problems when working with Network Security dataset. Firstly, the data is extremely imbalanced, which leads to the difficulty in learning. Furthermore, traffic analysis deals with communications which are time dependent and links between servers may appear and disappear with new requests and new users in the network. That is why detecting new unknown botnets is a real challenge in network security

CHAPTER 3

METHODOLOGY

3.1. Overall solution

To build the model capable of identifying the abnormal traffic, we follow the methodology as described below:

1. Select a Dataset: The first section is to collect traffic flow data. It can be accomplished by sourcing actual data traffic and converting them into NetFlow. Due to the lack of actual data, a well-known public domain dataset is another good option. In our proposal, CTU-13 (Sebastian Garcia, 2014) is selected, which is highly available and has been used in a considerable number of research studies. We discuss in detail about the dataset in the next section. But, in general, the features of data - StartTime, Duration, Proto, SrcAddr, Sport, Dir, DstAddr, Dport, State, sTos, dTos, TotPkts, TotBytes, SrcBytes and Label.
2. Feature Extraction: Once a dataset is selected, we then identify and extract the features, which is a crucial part in our methodology. NetFlow data includes categorical features that have to be encoded into numerical values. One popular solution to transform categorical values into numeric features is using one-hot encoding. In spite of the expansion of required memory to store, our hardware is capable of storing a considerably large amount of data since there are seven categorical features in our dataset.
3. Comparison of Supervised approaches. In the former solution, our model is trained to classify between the normal and abnormal samples. On the other hand, the latter approach is trained on data only containing normal samples; as a result, the intrusions will be detected based on the deviation between them and the model. The abnormal traffic is then classified thanks to the multi-classification model in order to figure out the specific attack.
4. Botnet detection: The last step in our methodology involves testing our model to see if it has capability of detecting botnet traffic from CTU-13 dataset. The overall performance of botnet detection is figured out based on aforementioned scores

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1. Data Analysis

4.1.1. CTU-13 Dataset

The CTU-13 Dataset is a labeled dataset used in the literature to train botnet detection algorithms. CTU-13 captures real botnet traffic combined with normal traffic and background traffic. In this section, the dataset is described to find the relevant features to extract, and train models
CTU-13 includes 13 different scenarios in which various botnets are used to simulate the abnormal traffic. The table below described specific botnet is used in each scenario and its percentage

	Botnet	Dataset	
		Size	Botnets
Train	Neris - Scenario 1	2 226 720	1.28%
	Neris - Scenario 2	1 431 539	1.45%
	Rbot - Scenario 3	2 024 053	4.99%
	Rbot - Scenario 4	470 663	2.36%
	Virut - Scenario 5	63 643	3.46%
	DonBot - Scenario 3	220 863	5.57%
	Sogou - Scenario 7	50 629	1.38%
	Total	**6 488 110**	**2.72%**
Test	Murlo - Scenario 8	1 643 574	6.8%
	Neris - Scenario 9	1 168 424	12.51%
	Rbot - Scenario 10	559 194	9.67%

	Rbot - Scenario 11	61 551	2.76%
	NSIS.ay - Scenario 12	156 790	10.2%
	Virut - Scenario 13	1 294 025	7.57%
	Total	**4 883 558**	**8.75%**

Table 1. Botnet distribution in CTU-13 dataset

Our data analysis is focused on the first scenario, which uses *Neris*. The Netflow file includes 2 824 636 bidirectional communications characterized by 15 features:

1. StartTime: Date of the flow start (from 2011/08/10 09:46:53 to 2011/08/10 15:54:07)

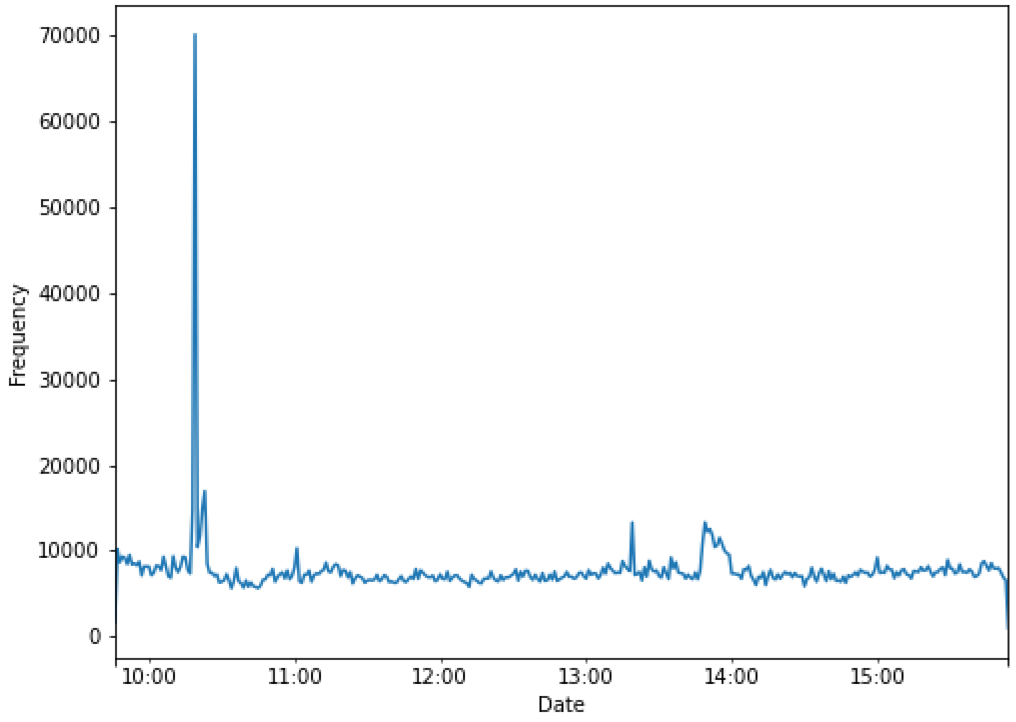


Figure 1. Frequency of Flow w.r.t Time

The Start times of NetFlow seems uniformly distributed except at around 10:20 due to starting of Netflow

2. Dur: Majority of the communications last less than 1ms

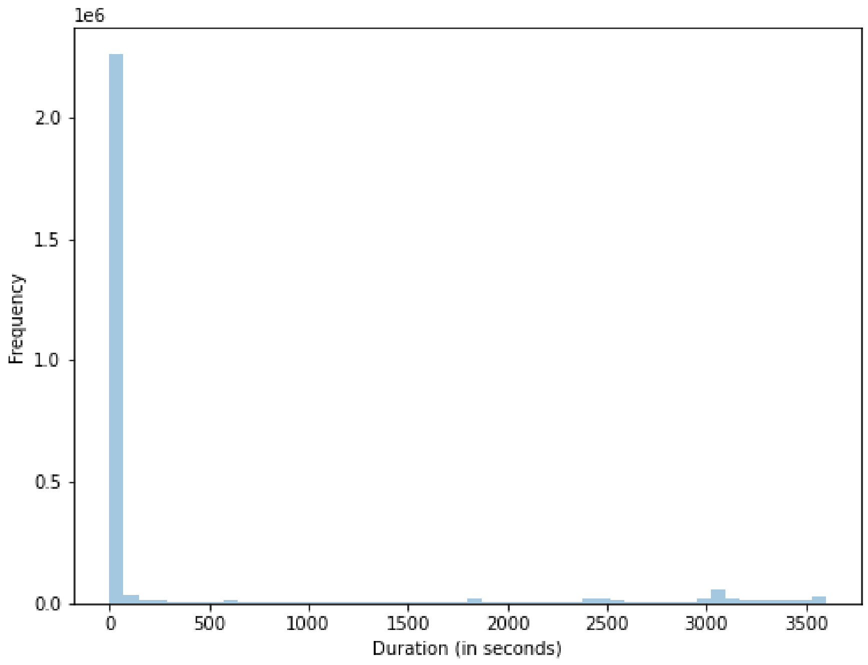


Figure 2. Frequency of Flow w.r.t Duration

3. Label

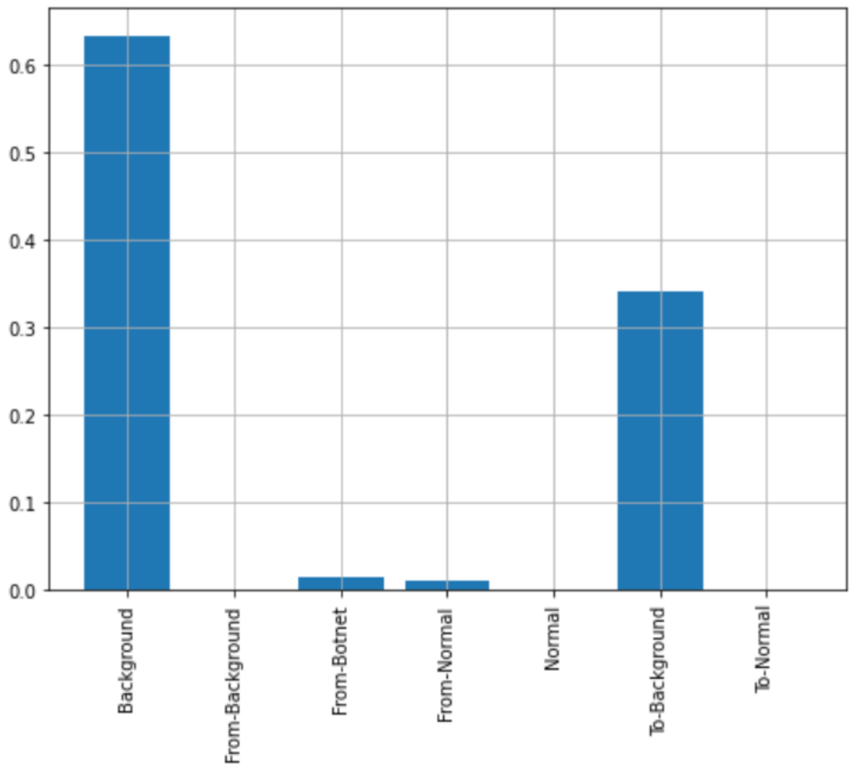


Figure 2. Distribution of Labels

4.1.2. Analysis of attack traffic sample

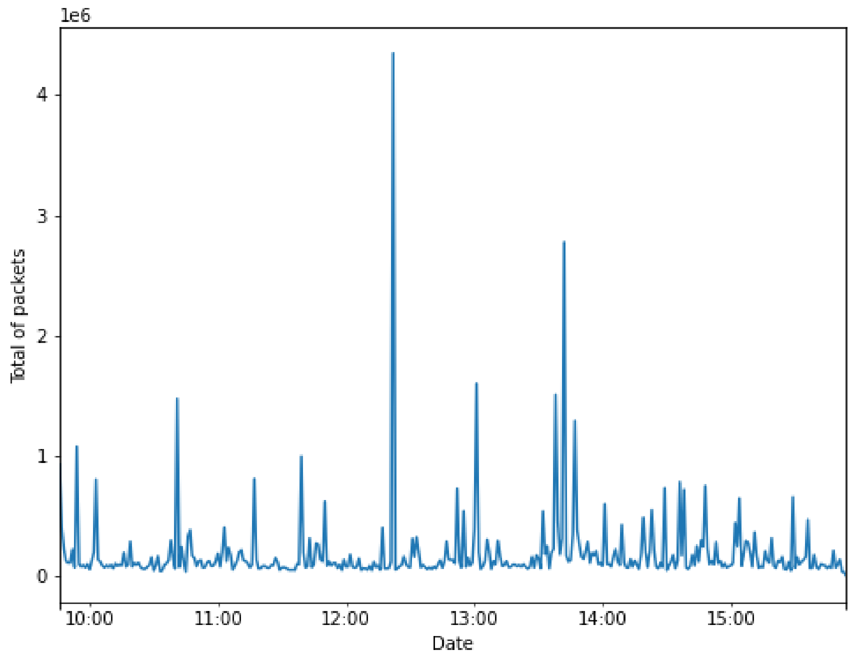


Figure 3. Frequency of Flow w.r.t Total of Packets

The figure 12 describes the total of packets with respect to time, which is grouped by every five minutes. The total number of packets increases suspiciously in the range between 12:00 and 13:00, which indicates the signal of botnest.

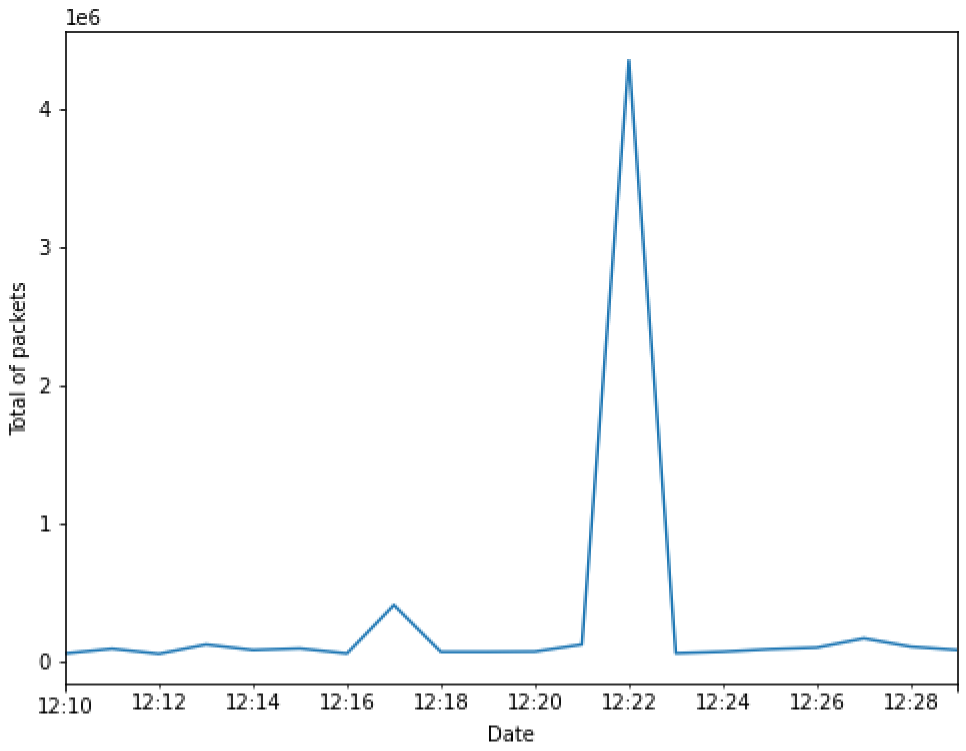


Figure 4. Frequency of Flow w.r.t Total of Packets

The figure 13 depicts the total of packets with respect to time in the time range which has suspicious traffic. Obviously, the total number of packets has the peak of 4 664 716 packets at 12:22.

4.2. Results

4.2.1. Metrics

To compare the results of two approaches, some metrics need to be defined to evaluate the performance of methods. The popular way is to look at the number of false positives (i.e., number of normal traffic classified as botnets), false negatives (i.e., number of abnormal traffic labeled as usual traffic) and accuracy of the multi-classification model which identify among the botnet once abnormal traffic is detected
To do this, five scores can be used:

- The recall: $R = \frac{TP}{TP + FN}$
- The precision: $P = \frac{TP}{TP + FP}$
- The F1 score: $F_1 = \frac{2}{P^{-1} + R^{-1}}$
- AUC score: area under ROC which is the curve of True Positive Rate with respect to False Positive Rate
- False Alarm Ratio (FAR): the number of false alarms per the total number of warnings or alarms $FAR = \frac{1}{2}(FPR + TPR) = \frac{1}{2}(\frac{FP}{FP + TN} + \frac{FN}{FN + TP})$

4.2.2. Algorithms

We have 2 approach:

a. Supervised learning: The Decision tree is involved to solve the binary classification problem. Label 0 is assigned to normal traffic, while botnet have the label of 1

Decision Tree

Use Grid Search to find the best parameter for Decision Tree with weight for class 0: 0.05 - 1: 0.95 and “gini” function to to measure the quality of a split.

- Max depth: 14
- Min sample split: 6
- Min sample leaf: 13

The result shows in figure below:

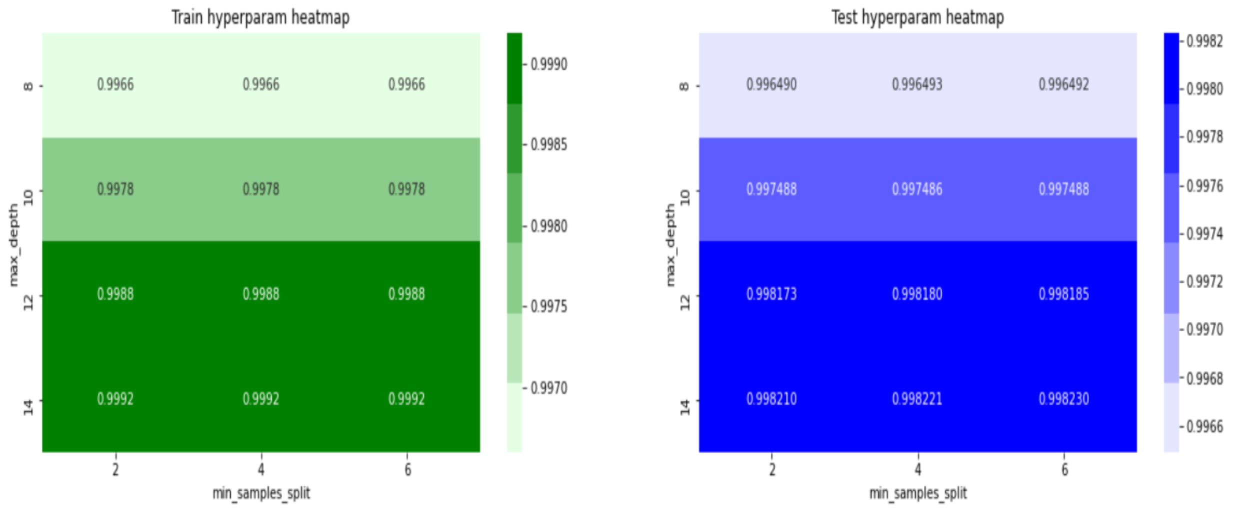


Figure 5. Max depth and min sample split tuning

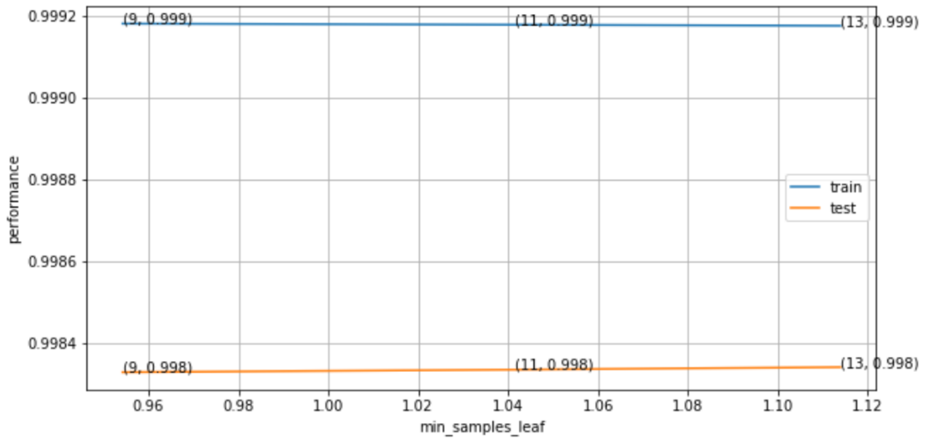


Figure 6. Min sample lead tuning

With best parameter: metrics score on training set and test set

```
+---------+-------+--------------------+--------------------+----------------------+
| Dataset | Model |        AUC         |      F1-score      |   False Alarm Rate   |
+---------+-------+--------------------+--------------------+----------------------+
|  Train  |   DT  | 0.9750622058833952 | 0.9546562500000001 | 0.024937794116604774 |
|   Test  |   DT  | 0.9658627665954107 | 0.9367598891408415 | 0.034137233404589296 |
+---------+-------+--------------------+--------------------+----------------------+
```

Table 2. Score on training and test set

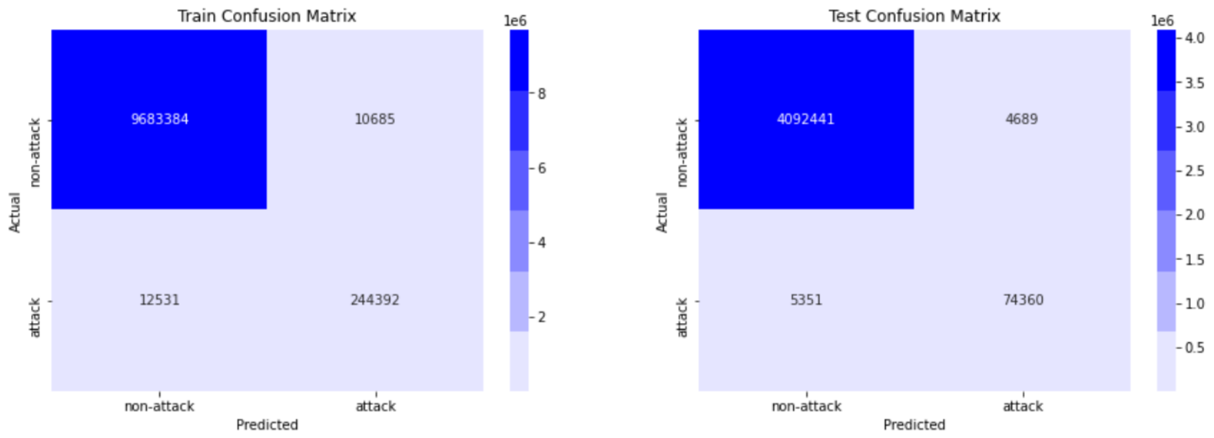


Figure 7. Confusion matrix of Decision tree with tuned parameter

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4097792
1	0.93	0.94	0.94	79049
accuracy			1.00	4176841
macro avg	0.97	0.97	0.97	4176841
Weighted avg	1.00	1.00	1.00	4176841

Table 3. Result of Decision Tree Model

Random Forest:

We will tune Random Forest hyperparameters with train and cross-validation data and use loops.

Use Grid Search find best parameter for Random Forest:
- N_samples: 20
- Criterion: Entropy

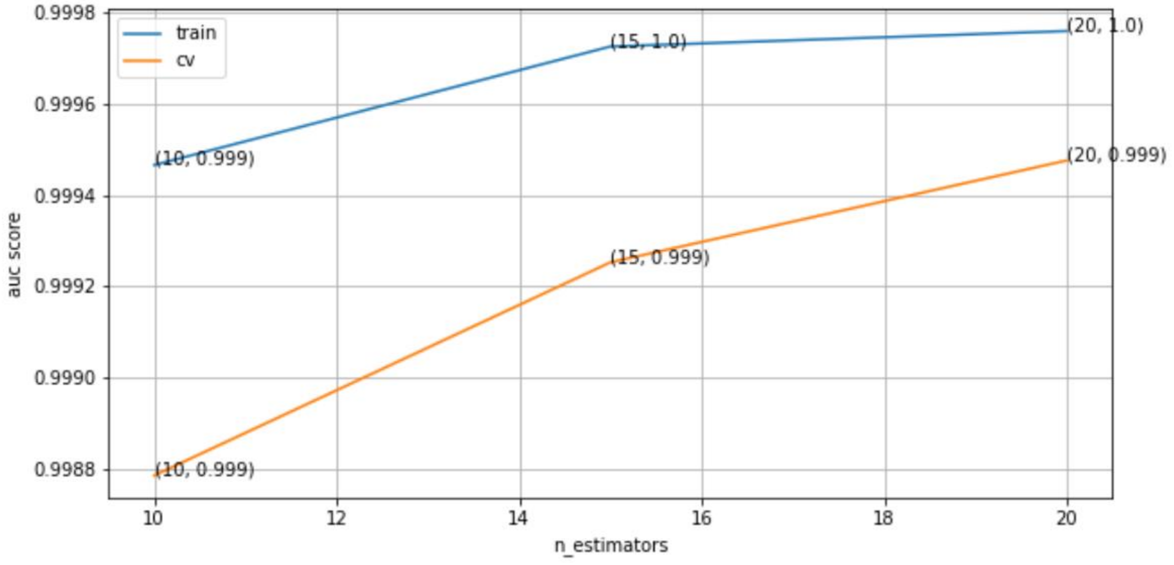


Figure 8. Number of estimator tuning

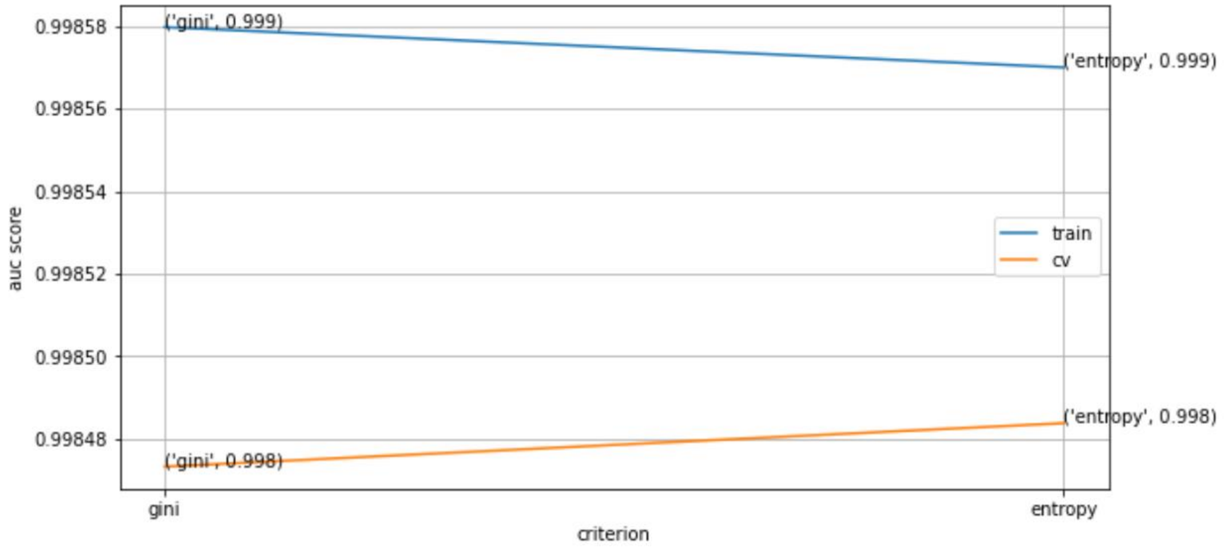
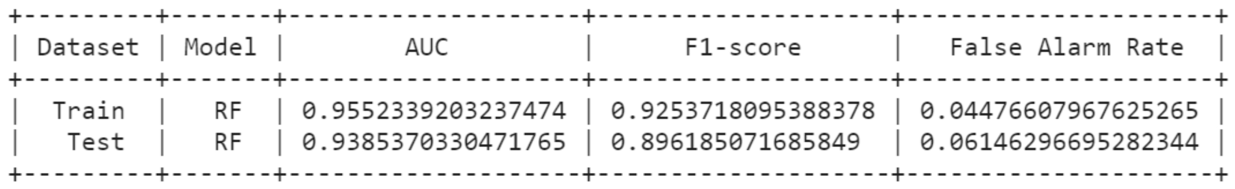


Figure 9. Criterion tuning



```
+---------+-------+--------------------+--------------------+---------------------+
| Dataset | Model |        AUC         |      F1-score      |   False Alarm Rate  |
+---------+-------+--------------------+--------------------+---------------------+
|  Train  |   RF  | 0.9552339203237474 | 0.9253718095388378 | 0.04476607967625265 |
|   Test  |   RF  | 0.9385370330471765 | 0.896185071685849  | 0.06146296695282344 |
+---------+-------+--------------------+--------------------+---------------------+
```

Table 3. Scores on training and testing set with tuned parameter

Involvement of class weight has the obvious impact on the performance of Decision Tree

name	auc	f1-score	far
DT	0.9669	0.9368	0.0341
RF	0.9385	0.8962	0.0615

Botnet classification: Rbot, Virut, Murlo, Neris, NSIS, SoGou, DonBot are the botnets to categorize. Only botnet samples are retained in the sub-dataset to classify the category of botnet. Distribution of each botnets in the sub-dataset:

- Murlo: 6127 samples
- Virut: 184987 samples
- Rbot: 40904 samples
- Neris: 33474 samples

Multiple Layer Perceptron (MLP) is used to train classification models. Specifically, our MLP uses 6 hidden layers: number hidden nodes associated with each layer: 128, 1024, 512, 128, 128, 10. ReLU is used in every hidden layer and Softmax is involved in the end to determine the probability distribution of all classes.

	precision	recal	f1-score	support
0	1.00	0.98	0.99	1852
1	0.89	0.92	0.91	55486
2	0.99	0.44	0.61	12306
3	0.60	0.88	0.71	10004
accuracy			0.84	79648
macro avg	0.87	0.81	0.80	79648
weight avg	0.87	0.84	0.84	79648

Table 3. Result of MLP Model

result of model

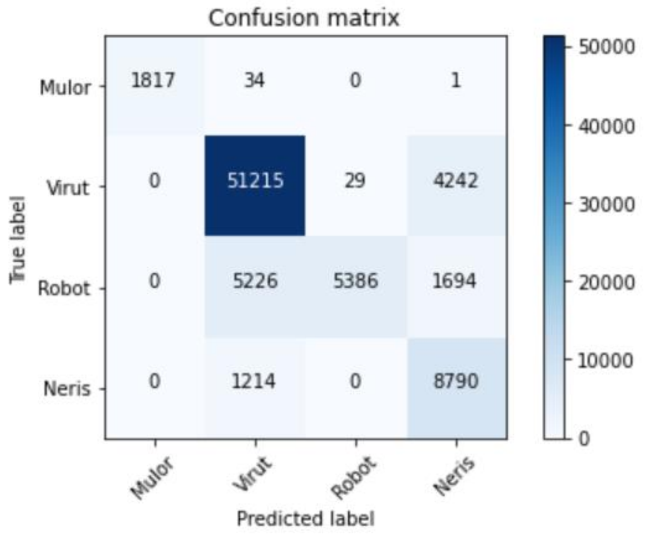


Figure 10. Confusion matrix of MLP model

References

Sebastian Garcia, M. G. (2014). *An empirical comparison of botnet detection methods*, 100-123.