Using the ISOT HTTP Dataset to Identify Botnet Anomalies in Network Traffic.

The Data.

The data was generated from an experiment conducted by the Canadian Institute for Cybersecurity in the University of New Brunswick in 2017. The experiment involved connecting multiple end-user virtual machines to a single DNS server via a firewall. One section of the virtual machines where each were actively running one distinct and known desktop user application and web service on Windows 7. The applications included the following:

- Dropbox
- Avast
- Adobe Reader
- Adobe Software Suite
- Chrome
- Firefox
- Malwarebyte
- WPS office
- Windows update
- utorrent.com bittorrent.com
- fosshub.com audacity
- Bytefence-com
- Thunderbird Mozila
- Skype
- Facebook massager
- CCleaner
- Win update
- Hitmanpro.com
- background data from windows
- time.windows.com
- time.microsoft.akadns.net
- dns.msftncsi.com

The other section of virtual machines were infected with known botnet malware running on Windows XP as their base operating system. The botnet services were as follows:

- zyklon.botnet.isot
- blue.botnet.isot
- liphyra.botnet.isot
- gaudox.botnet.isot gdox.botnet.isot

- dox.botnet.isot
- blackout.botnet.isot
- citadel.botnet.isot
- be.botnet.isot black energy
- zeus.botnet.isot

Data Preparation.

The data sourced from the Canadian Institute for Cybersecurity website came in as pcap file.

Wireshark was used to read and extract statistically analysed UDP and TCP packets from the files. The analysis from Wireshark was exported to 2 csv files for further processing, one for the concatenated botnet patterns and another for the concatenated applications and DNS patterns into as single file named 'normal'. The files had the features in the following order:

Attributes	Description
Address_A	Source IP address.
Port_A	Source port.
Address_B	Destination IP address.
Port_B	Destination port.
Total_Packets	Total number of packets sent during a full TCP or UDP conversation.
Total_Bytes	Total amount of data transmitted in in a full conversation in bytes.
Packets_Forward	Number of packets sent by the source IP to the destination IP.
Bytes_Forward	Number of bytes transmitted by the source to the destination IP.
Packets_Backward	Number of packets sent to the source IP from the destination IP.
Bytes_Backward	Number of bytes transmitted to the source from the destination IP.
Rel_Start	The relative start time the transition was captured.
Duration	The time period take for each transition.
Bits/s_Forward	Number of bits transmitted by the source to the destination IP.
Bits/s_Backward	Number of bits transmitted to the source from the destination IP.

Data Modelling.

Both the botnet and normal datasets were pre-processed separately. The columns 'Address_A' and 'Address_B' were dropped from the data due to their lack of the consistent information required for any further analysis. 'Port_A', 'Port_B' and 'Rel_Start' where also excluded from any further analysis but maintained on the far left within the data in their listed order above.

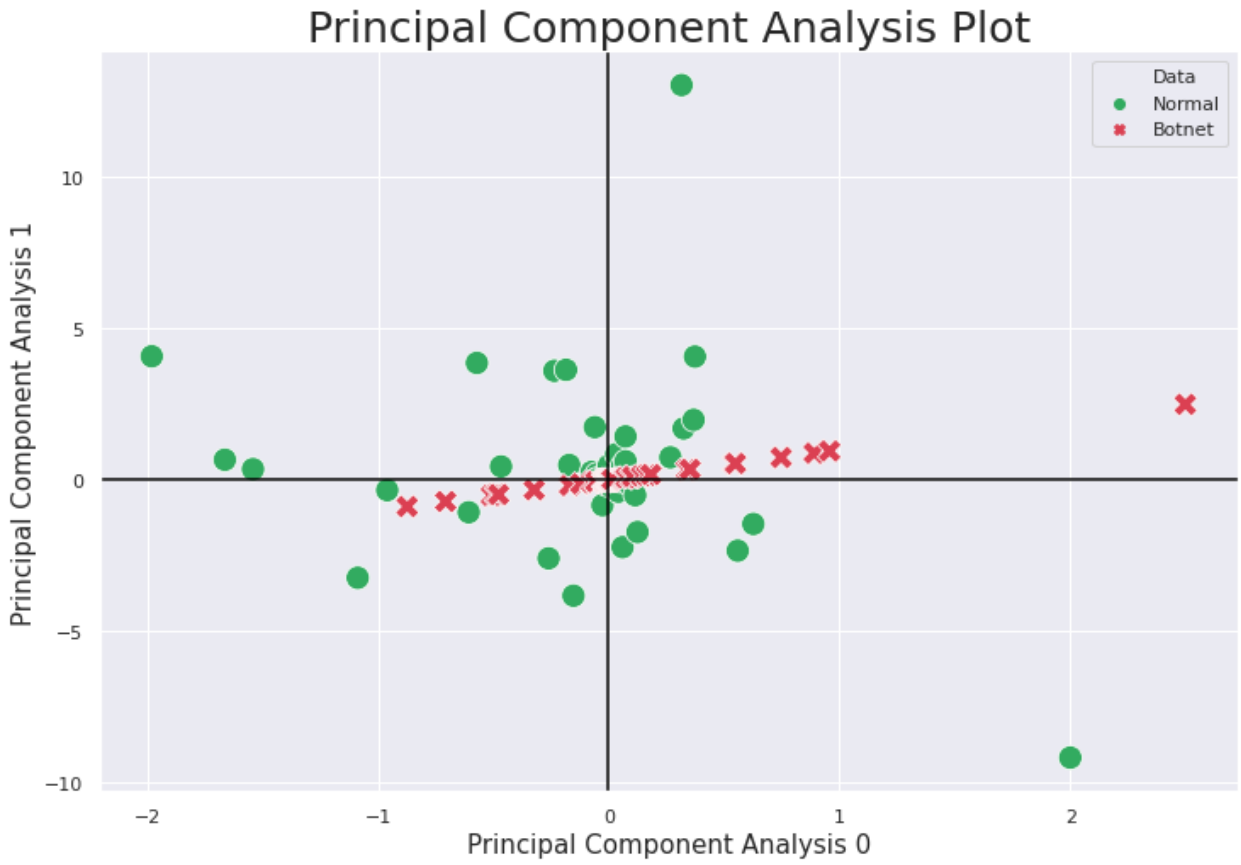
The Download_Upload_Ratio was then calculated from dividing the Bits/s_Forward values from the Bits/s_Backward. As for the reminder of the attributes in the data, including the calculated Download_Upload_Ratio, underwent the following statistical analysis and added into the original dataset as a column:

Function	Process
Mean	Arithmetic average between two flows.
Exponential Mean	Exponential moving average between two flows.
Standard Deviation	Standard deviation between two flows.
Delta	Difference between two flows.
Sum	Addition between two flows.
Change	Percentage change between two flows.
Max	Maximum value between two flows.
Min	Minimum value between two flows.

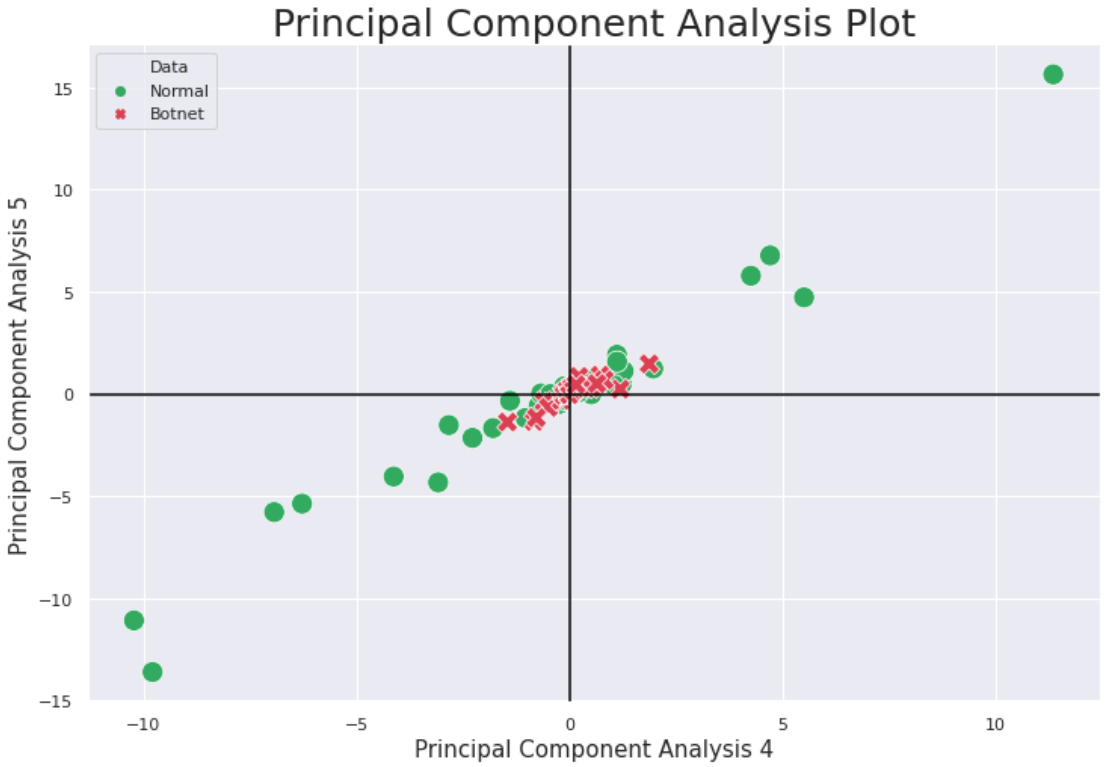
In total, 93 features were formed from the data including the original 12 that were maintained from the raw data. All sections in either of the datasets with Nan values were imputed with a 0.

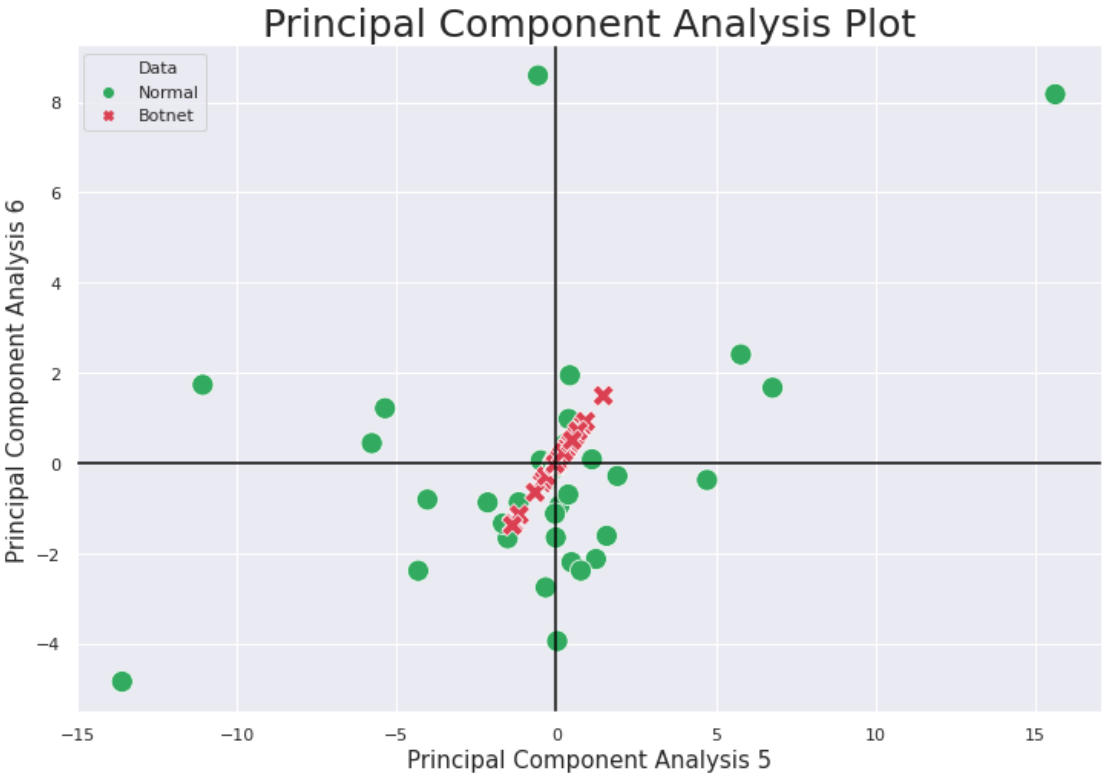
The data was then scaled and decomposed using Principle Component Analysis with the number of components set to 93. This was necessary in order to centre the values and reduced variation in the data. All the 93 components identified from the PCA where used in the model training.

This also unlocked a number of unique and distinctive characteristics between the values in the botnet and normal dataset as shown below.

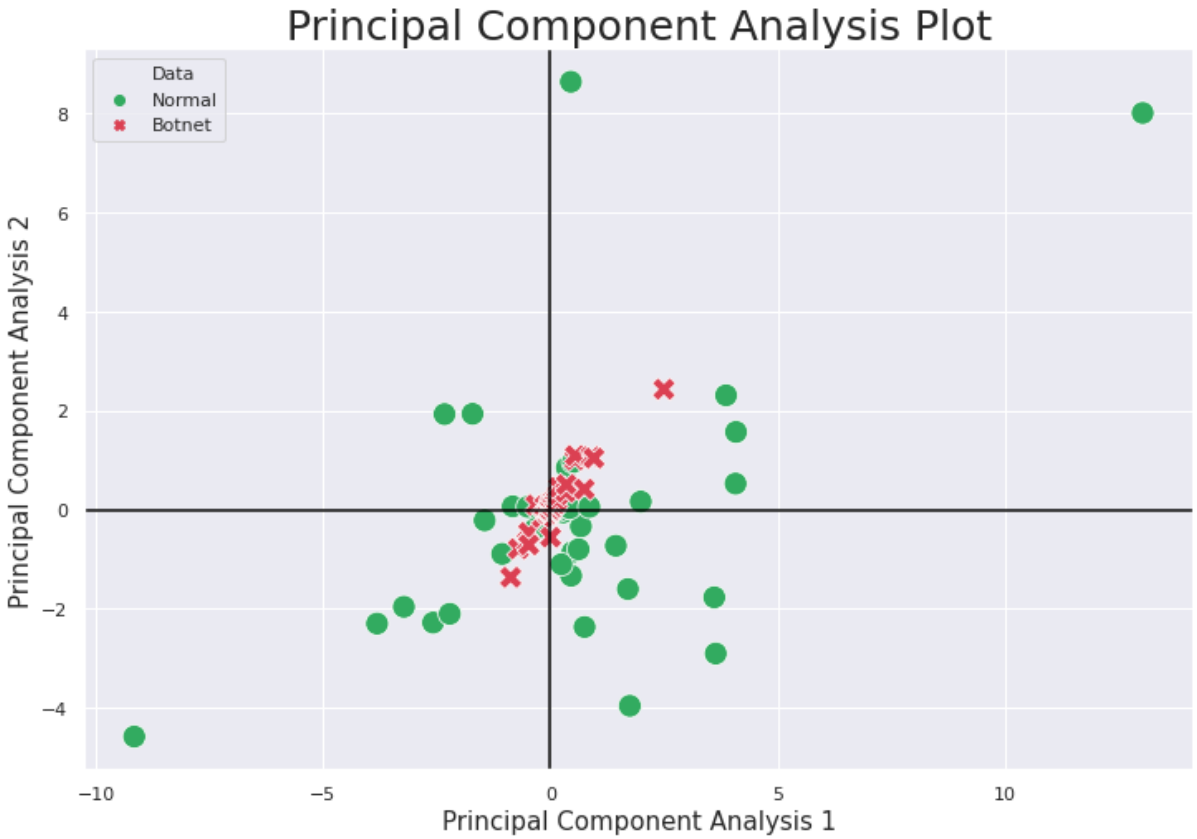


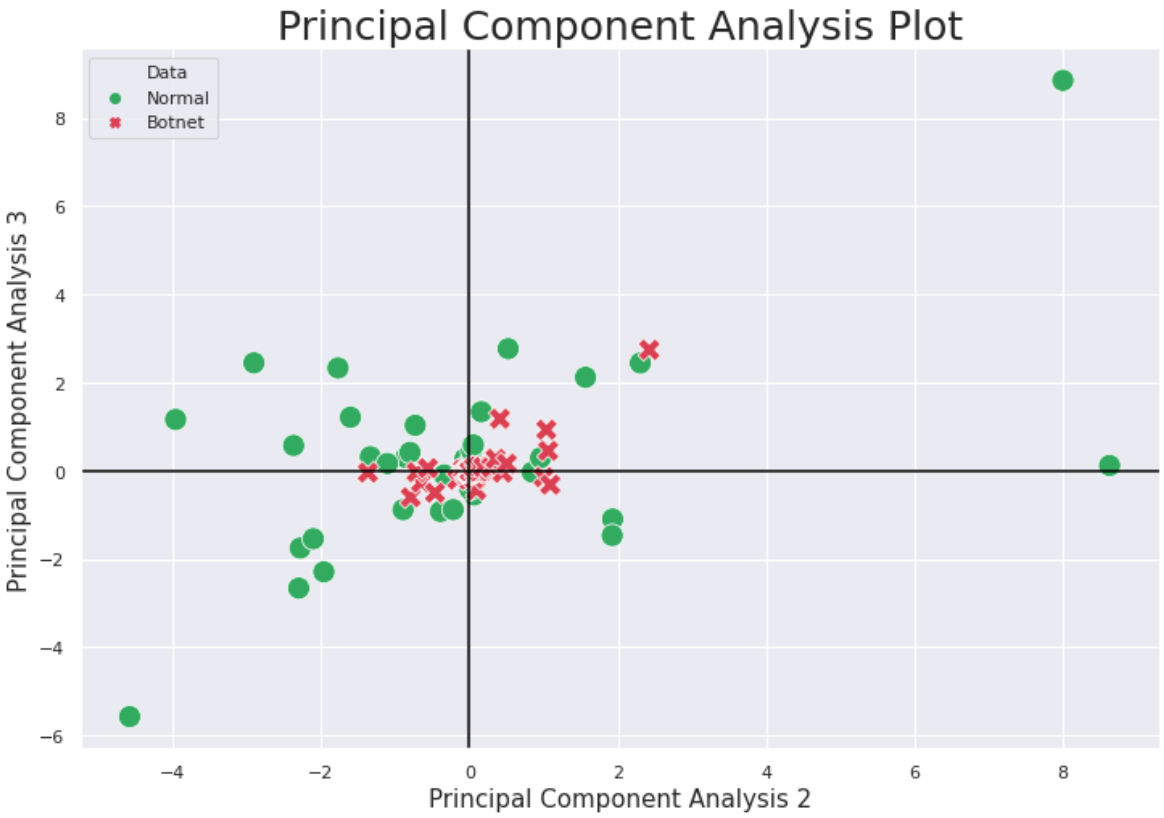
A common trend in the analysis observed was that the values across the 93 features in the botnet dataset where highly correlated and showed a consistent behaviour pattern within them as shown in the above and below graphs.

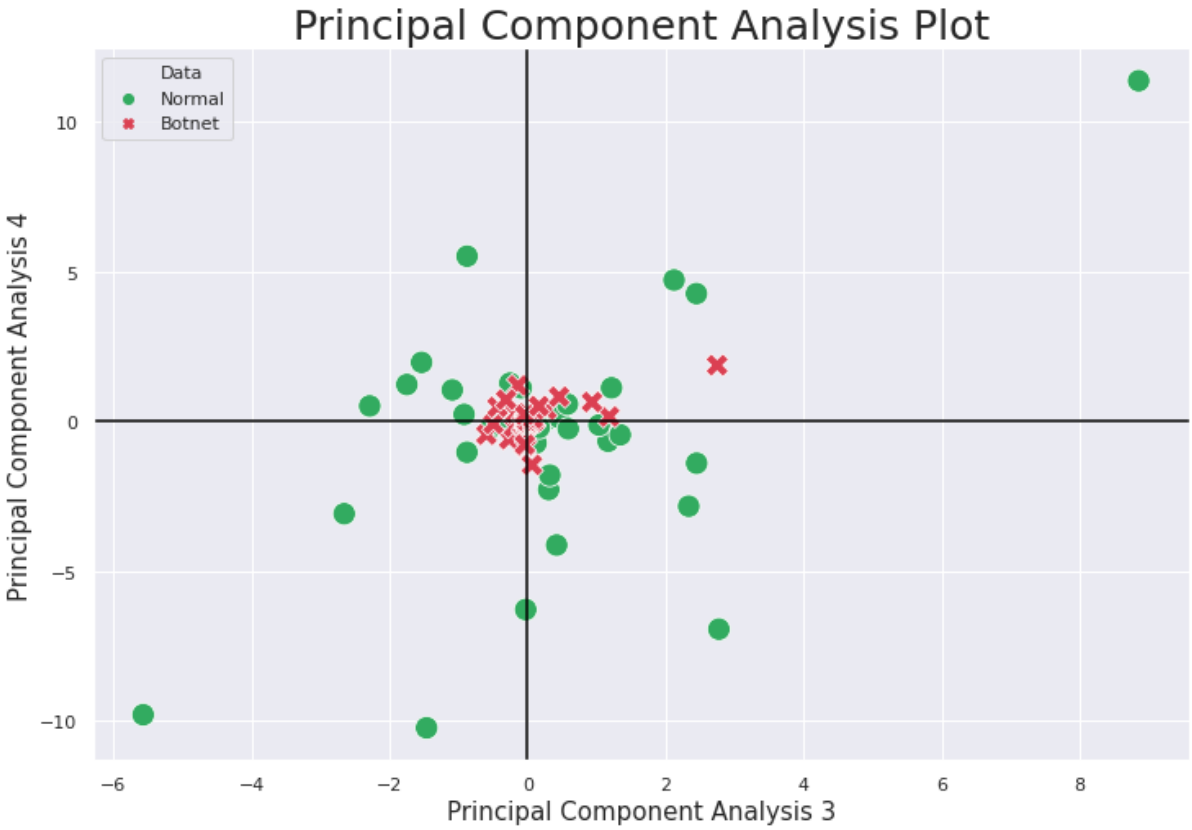




However, this was not always consistent when certain components where paired together but the botnet pattern where observed to be significantly distinctive from those exhibited by normal network traffic.




Principal Component Analysis Plot
Data
Normal
Botnet
Principal Component Analysis 3
Principal Component Analysis 2
8
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
8


Principal Component Analysis Plot
Data
Normal
Botnet
Principal Component Analysis 4
Principal Component Analysis 3
10
5
0
−5
−10
−6
−4
−2
0
2
4
6
8

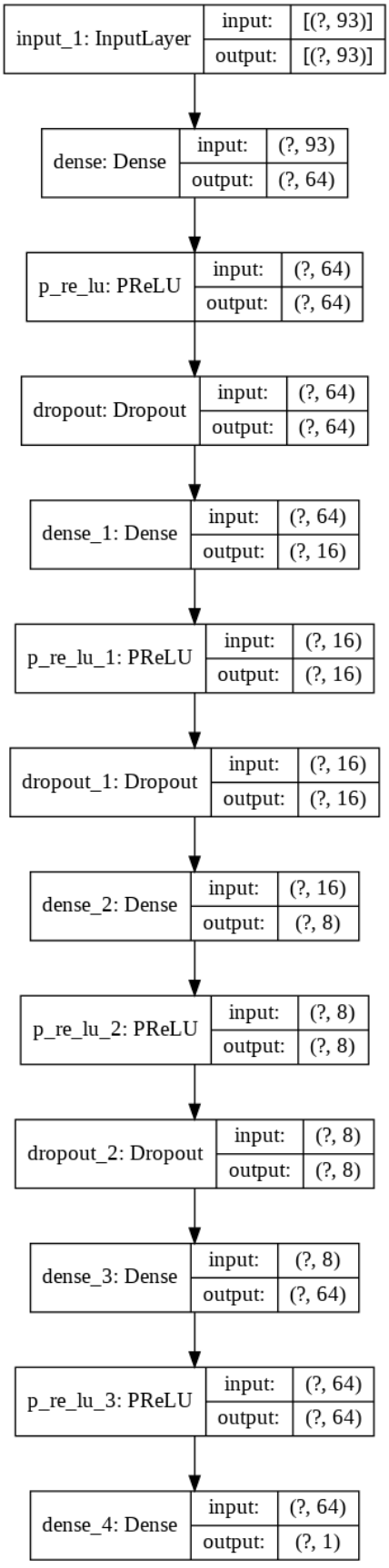
The column ‘Data’ was added to both the botnet and normal dataset labelling each observation ‘Botnet’ and ‘Normal’ respectively. At this point, both datasets where then concatenated. Finally, the labels in the ‘Data’ column were label encoded to 0 and 1 representing ‘Normal’ and ‘Botnet’ respectively.

The Model.

As a result of the decomposition analysis leading to the distinctive parameters between the botnet and normal network traffic generated above, a complex model was not required. Therefore, the model used was a Multi-Layer Perception binary classifier neural network with, hence it uses the sigmoid function to shown below with its pre-set threshold set to 0.5.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

The model structure demonstrated by the graph below:

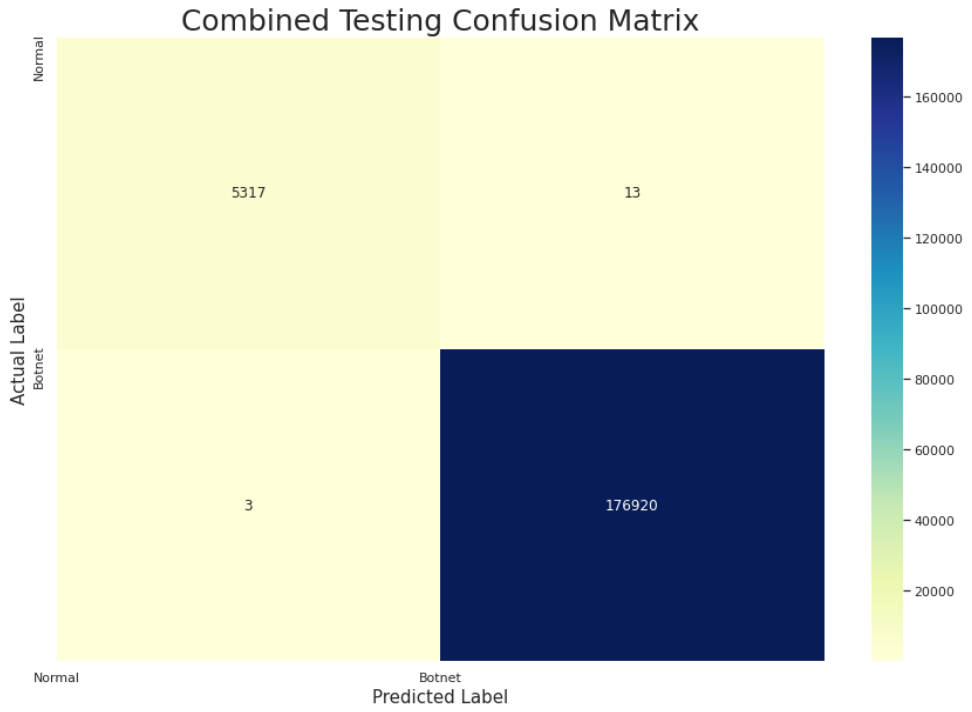

input_1: InputLayer
input: [(?, 93)]
output: [(?, 93)]
dense: Dense
input: (?, 93)
output: (?, 64)
p_re_lu: PReLU
input: (?, 64)
output: (?, 64)
dropout: Dropout
input: (?, 64)
output: (?, 64)
dense_1: Dense
input: (?, 64)
output: (?, 16)
p_re_lu_1: PReLU
input: (?, 16)
output: (?, 16)
dropout_1: Dropout
input: (?, 16)
output: (?, 16)
dense_2: Dense
input: (?, 16)
output: (?, 8)
p_re_lu_2: PReLU
input: (?, 8)
output: (?, 8)
dropout_2: Dropout
input: (?, 8)
output: (?, 8)
dense_3: Dense
input: (?, 8)
output: (?, 64)
p_re_lu_3: PReLU
input: (?, 64)
output: (?, 64)
dense_4: Dense
input: (?, 64)
output: (?, 1)

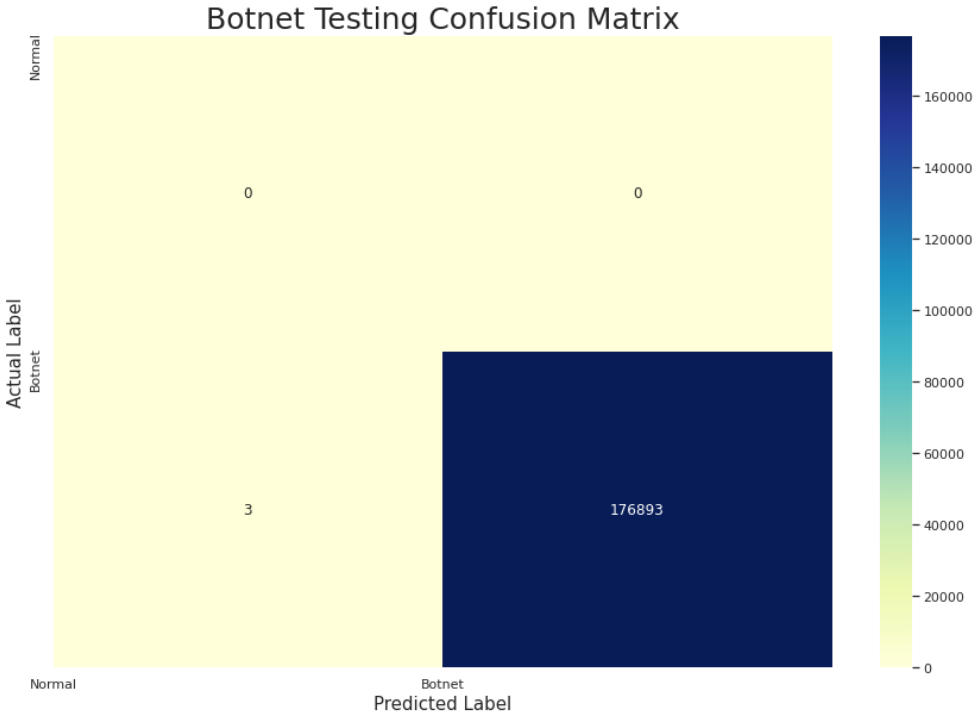
Model Testing and Performance Analysis.

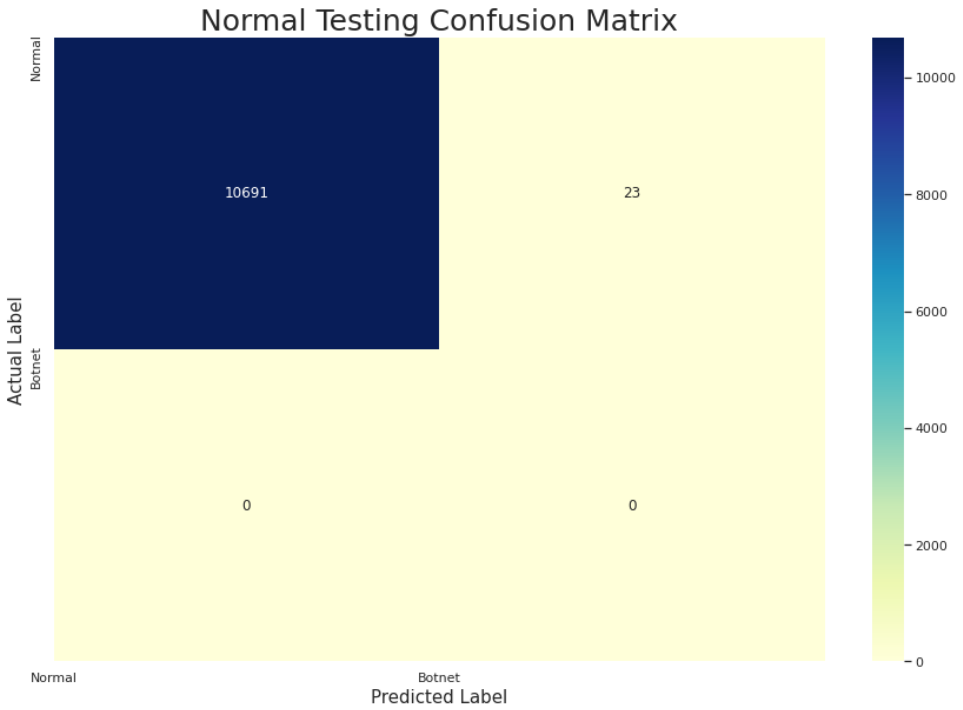
The model gave a general minimum accuracy of 99 % and a maximum accuracy of 99.94%. The model was also tested on several accuracy parameters as well as on different sets of data marked as train and test. The results obtained are as follows:

	Train	**Test**
Precision	99 %	99 %
Recall	99 %	99 %
F1 Score	99 %	99 %

In order to demonstrate the models performance and to countercheck the mentioned accuracy levels above, a confusion matrix was used to identify how well the model was able to perform its prediction given different datasets. This resulted to the following outcomes:


Combined Testing Confusion Matrix
Normal
Botnet
Actual Label
5317
13
3
176920
Normal
Botnet
Predicted Label
160000
140000
120000
100000
80000
60000
40000
20000


Botnet Testing Confusion Matrix
Normal
Botnet
Actual Label
0
0
3
176893
Normal
Botnet
Predicted Label
160000
140000
120000
100000
80000
60000
40000
20000
0



As observed above, the model was not only able to perform well when both the botnet and normal datasets where combined but also when the model was required to make a distinction when presented with either the isolated botnet or normal datasets.

In conclusion, the model demonstrated that it was able to generalize on the patterns on the data as expected despite the imbalance in the data set.

The Application.

The app runs on a Command Line or Terminal interface. It is executed as such in the example below:

```
$ python botnet.py --file sample_data/network_traffic.csv
```

The execution command includes:

Command	Description
python	Application runtime.
botnet	Application filename.
.py	Application file extension.
--file	File declaration argument.
Sample_data/network_traffic	File path and filename to be analysed.
.csv	File extension. The file has to be either a text or csv format.

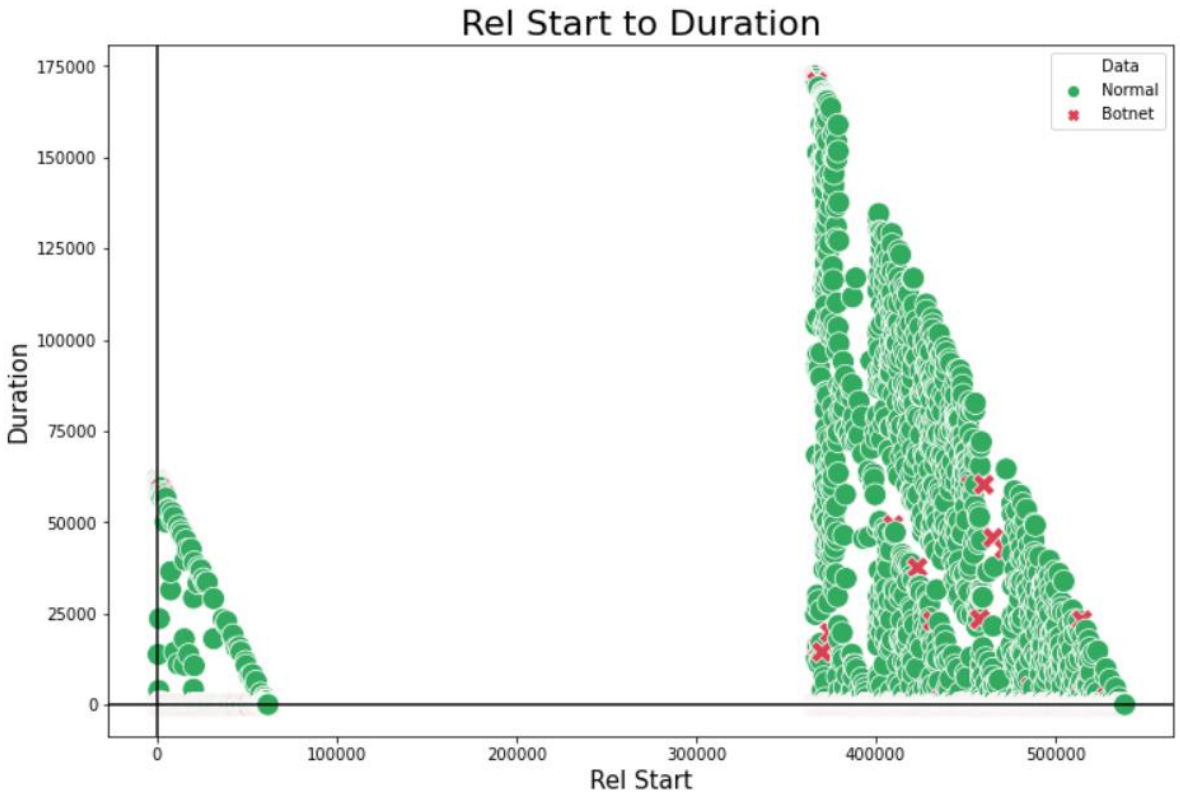
Each of the mentioned elements in the command mentioned above must be declared for the application to run as expected in the order mentioned. The application takes in a csv file with the statistically analysed TCP/UDP conversations in a pcap file using Wireshark. The app analyses the data through all the data analysis steps discussed and feeds the parameters to the model in order to identify any suspected botnets. The app then filters any predicted botnets from the data so as to identify only the unique instances of the suspected botnets as well as the targeted IP address and produces its results in the following as shown below:

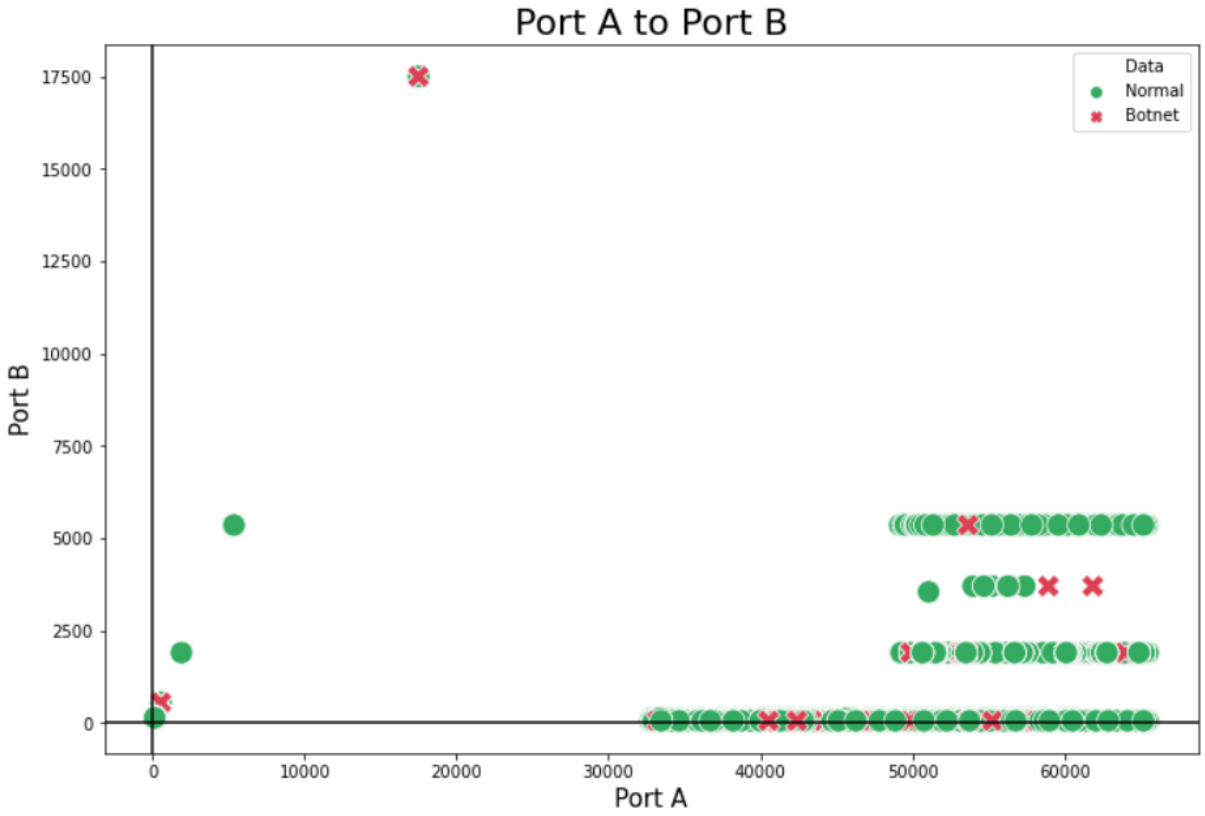
```
Suspected Botnet IP Address:      192.168.50.51
Targeted IP Address:              224.0.0.252
Suspected Botnet IP Address:      192.168.50.19
Targeted IP Address:              255.255.255.255
Suspected Botnet IP Address:      192.168.50.19
Targeted IP Address:              192.168.50.255
Suspected Botnet IP Address:      ff02::1:2
Targeted IP Address:              fe80::e14d:8fc5:b840:50b8
Suspected Botnet IP Address:      fe80::e40e:a6ad:3c4d:d40d
Targeted IP Address:              ff02::1:3
Suspected Botnet IP Address:      192.168.50.59
Targeted IP Address:              224.0.0.252
Suspected Botnet IP Address:      fe80::e14d:8fc5:b840:50b8
Targeted IP Address:              ff02::1:3
Suspected Botnet IP Address:      192.168.50.59
Targeted IP Address:              239.255.255.250
Suspected Botnet IP Address:      192.168.50.88
Targeted IP Address:              8.8.8.8
Suspected Botnet IP Address:      192.168.50.54
Targeted IP Address:              224.0.0.252
Suspected Botnet IP Address:      192.168.50.56
Targeted IP Address:              224.0.0.252
Suspected Botnet IP Address:      fe80::354c:4ae3:193:37a2
Targeted IP Address:              ff02::1:3
Suspected Botnet IP Address:      fe80::f42d:b31d:a4b2:df73
Targeted IP Address:              ff02::1:3
Suspected Botnet IP Address:      ff02::1:2
Targeted IP Address:              fe80::e40e:a6ad:3c4d:d40d
Suspected Botnet IP Address:      192.168.50.69
Targeted IP Address:              224.0.0.252
Suspected Botnet IP Address:      fe80::8932:730f:c9d8:6fca
Targeted IP Address:              ff02::1:3
Suspected Botnet IP Address:      192.168.50.55
Targeted IP Address:              224.0.0.252
```

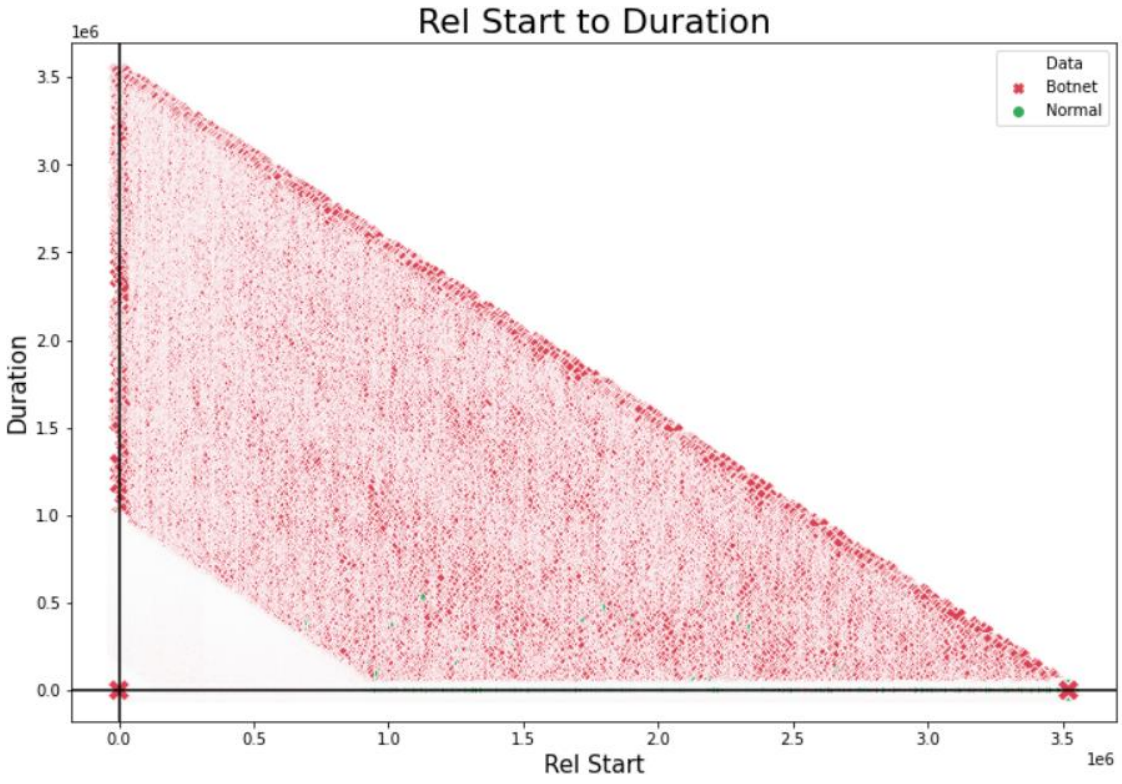
In addition, the app also plots the following parameters and displays 6 scatter graphs in to visualize the predictions:

1. Port_A to Port_B
2. Total_Packets to Total_Bytes
3. Bytes_Forward to Bytes_Backward
4. Packets_Forward to Packets_Backward
5. Rel_Start to Duration
6. Bits/s_Forward to Bits/s_Backward

The following demonstrates examples of the plots from the app:




Port A to Port B
Data
Normal
Botnet
Port B
Port A
17500
15000
12500
10000
7500
5000
2500
0
0
10000
20000
30000
40000
50000
60000


Rel Start to Duration
1e6
Data
Botnet
Normal
Duration
Rel Start
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e6

Drawbacks.

There were some drawbacks encountered during the development of the data pipeline and the model. They include:

- The data needed some initial pre-processing using Wireshark as discussed rather than directly handled by its data pipeline from its raw pcap files.
- The data used was generated in 2017 hence can be considered outdated considering the above setups is to be used in a network security setting.
- The data was generated in a lab setting. Therefore the parameters within it may not be true to a real life setting.
- The data used to train the model was highly imbalanced, therefore there is a high bias in the model to identifying botnets in the data. Hence, false positives are to be expected from the model.

Recommendation.

Taking into consideration the current setup and its drawbacks, the following is recommended:

- The model requires regular and constant updating by retraining it on updated data given the fact it is to be used in a security related application setup.
- A separate program to analyse the pcap files to extract the required information useful to the model.
- Using data that can be attributed as normal traffic such as log files rather than generically collected data can help ensure the data is more realistic for a real life setting. This can ensure the setup can be used in a production setting.

Conclusion.

The results have helped identify that network anomalies such as those in the botnet data can be differentiated from known applications and DNS traffic when analysed statistically.

Appendix.
Mathematical formulas used:

- Mean:

$$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$ where: i = 1, N = 2

- Exponential Mean:

$$(x_i - x_n) \times (2 \div (n + 1)) + x_n$$ where i = 1, n = 2

- Standard Deviation:

$$s = \sqrt{\frac{\sum (x - \overline{x})^2}{n - 1}}$$ where: N = 2

- Delta:

$$(x_i - x_n)$$ where i = 1, n = 2

- Sum:

$$\sum_{i=1}^{N} x_i$$ where i = 1, n = 2

- Change:

$$(x_i \div x_n)$$ where i = 1, n = 2

- Max:

$$\mathbf{Max}\ (x_i, x_n)$$ where i = 1, n = 2

- Min:

$$\mathbf{Min}\ (x_i, x_n)$$ where i = 1, n = 2