

Configuration Manual

MSc Internship Cyber Security

Sheriff Agboola _{x18123171}

School of Computing National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland Project Submission Sheet School of Computing



Student Name:	Sheriff Agboola
Student ID:	x18123171
Programme:	Cyber Security
Year:	2019
Module:	MSc Internship
Supervisor:	Vikas Sahni
Submission Due Date:	12/12/2019
Project Title:	A Comparative Analysis of Base Learning and Ensemble
	Learning for Botnet Detection
Word Count:	XXX
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).					
Attach a Moodle submission receipt of the online project submission, to					
each project (including multiple copies).					
You must ensure that you retain a HARD COPY of the project, both for					
your own reference and in case a project is lost or mislaid. It is not sufficient to keep					
a copy on computer.					

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only						
Signature:						
Date:						
Penalty Applied (if applicable):						

Configuration Manual

Sheriff Agboola x18123171

This configuration manual describes the features and capabilities of the tools utilised during the course of this research. It provides detailed instructions on how best to replicate the experiment carried out

1 Dataset

1.1 ISCX Botnet Dataset

ISCX Botnet dataset is developed in 2014 which is a detailed, packet capture (pcap) dataset containing traces of the operation and valid traffic of 16 kinds of botnets. The researchers merged traces from the ISOT Botnet data set, ISCX 2012 IDS data and Malware Capture Facility Project set of data to create the dataset. The dataset is categorised into train set of about 4.9 GB and test set of about 2.0 GB training set comprises traffic produced by 7 types of botnets, while the test set includes traffic generated by 16 types of botnet. The data-set includes the network traffic (PCAP-file), as well as flow (sub-)classification information in XML or text folder format, as well as non-botnet and botnet traffic which is available on University of New Brunswick(UNB)¹ website.

1.2 Dataset Conversion

The flow features from the ISCX botnet dataset which are in pcap file format will be extarcted using Flowtbag 2 developed by Daniel Arndt.

2 Environment

2.1 Anaconda

This section describes how to install Anaconda on a Windows operation system. Anaconda which is an open-source package manager designed for data analysis and machine learning. The steps outlined below is similar to installing the software in Windows and Mac OS.

2.2 Jupyter Notebook

In order for the installation of Anaconda framework and Jupyter notebook a detailed step is given in the youtube video by AP-Monitor.com 3 Jupyter notebook can be launched

¹http://205.174.165.80/CICDataset/ISCX-Bot-2014/

²https://github.com/DanielArndt/flowtbag

³https://www.youtube.com/watch?v=LrMOrMb8-3s

using the following steps:

open command prompt with administrator privileges type jupyter notebook

C\WINDOWS\system32\cmd.exe - jupyter notebook	-		Х
Microsoft Windows [Version 10.0.18362.476]			,
(c) 2019 Microsoft Corporation. All rights reserved.			
C:\Users\Noir>jupyter notebook			
[I 02:27:37.610 NotebookApp] The port 8888 is already in use, trying another port.			
<pre>[I 02:27:37.838 NotebookApp] JupyterLab extension loaded from C:\Users\Noir\Anaconda3\lib\site-packages\;</pre>	jupyte	rlab	
[I 02:27:37.838 NotebookApp] JupyterLab application directory is C:\Users\Noir\Anaconda3\share\jupyter\la	ab		
[I 02:27:37.842 NotebookApp] Serving notebooks from local directory: C:\Users\Noir			
[I 02:27:37.842 NotebookApp] The Jupyter Notebook is running at:			
[I 02:27:37.842 NotebookApp] http://localhost:8889/?token=e19da90948e40228b4a7fb33218e0c5ada3cc6d9885fb7	ef		
[I 02:27:37.842 NotebookApp] or http://127.0.0.1:8889/?token=e19da90948e40228b4a7fb33218e0c5ada3cc6d988	ofb7ef		
[I 02:27:37.842 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip (confir	mation)).
[C 02:2/:3/.9/4 NotebookApp]			
To access the notebook onen this file in a browser			
file:///C:/Users/Noir/AonData/Roaming/jupyter/runtime/nbserver-16608-open.html			
Or copy and paste one of these URLs:			
http://localhost:8889/?token=e19da90948e40228b4a7fb33218e0c5ada3cc6d9885fb7ef			
or http://127.0.0.1:8889/?token=e19da90948e40228b4a7fb33218e0c5ada3cc6d9885fb7ef			

Figure 1: Launching Jupyter notebook

Jupyter notebook uses the default browser of the system in use hence, a browser tab is opened which will represent the jupyter notebook environment.

```
double click on the desktop folder locate and double click the botnet folder
```

localhost:8889/tree/Deskton		0 💠 🖪 :
• Iodamostooos, recipiciskop		
	Files Running Clusters	
	0 - Desktop	i i
	b	
	backup	
	botnet	
•	cloud	
	final documentation	
	ICT	
	📮 🖿 ImageJ	

Figure 2: Folder selection

There are two files with the ipynb extension which are Preprocessing and BotnetDetection. The preprocessing file contains the code for importation of the dataset, required libraries,data cleaning and data balancing while the botnet detection file contains the implemented algorithms, results and the visualisation of the results.

seconds ago	
8 hours ago 4.44 MB	
a day ago 90.6 kB	
a day ago 72 B	
16 days ago 460 B	
16 days ago 62 B	
16 days ago 21.8 MB	
16 days ago 28 B	
16 days ago 66 B	
8 days ago 3.68 MB	
a day ago 2.78 MB	
16 days ago 3.66 MB	
	UploadNew -2Name +Last ModifiedFile sizeSeconds agu-8 hours ago9.0.6 kBa day ago90.6 kBa day ago72 B16 days ago62 B16 days ago21.8 MB16 days ago28 B16 days ago3.68 MB8 daya ago2.78 MB16 days ago2.78 MB16 days ago3.66 MB

Figure 3: Folder selection

				Widge					
		► Run							
Read	d Fea	atures							
#Imports from collect import pand import nump from sklear	tions imp as as pd y as np n.model_s								
['srcip', 'sr _fpktl', 'max 'min_biat', 'n le', 'max_idin _cnt', 'total	port', 'dst fpktl', 'st mean_biat', ", 'std_idi fhlen', 'to	ip', 'dstport', d_fpktl', 'min_b 'max_biat', 'std e', 'sflow_fpack tal_bhlen', 'dsc	'proto', ' pktl', 'me Lbiat', 'd ets', 'sf] P']	total_fp an_bpktl luration ow_fbyte	packets', 'total_f l', 'max_bpktl', ' ', 'min_active', ' es', 'sflow_bpacke	volume', 'total_ std_bpktl', 'min mean_active', 'm ts', 'sflow_bbyt	bpackets', 'total_ _fiat', 'mean_fiat ax_active', 'std_a es', 'fpsh_cnt', '	bvolume', 'min_fp ', 'max_fiat', 's ctive', 'min_idle bpsh_cnt', 'fung_	ktl', 'mean td_fiat', t', 'mean_id _cnt', 'burg
	pd.read				iset.csv',name				
srcip	srcport	dstip	dstport	proto	total fpackets	total fvolume	total bpackets	total byolume	min foktl
0 147.32.84.18	0 1027	74.125.232.195	80			1082			
1 147.32.84.18	0 1040	94.63.149.152				1476		26238	
3 147.32.84.18	0 1044	60.190.223.75	888			1006		1882	
						2906			

Figure 4: Main page

3 Data Importation and Extraction

3.1 Library importation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

3.1.1 Numpy

Numpy is an array processing package for specific purposes. This offers a multi-dimensional structure with maximum performance and resources for interacting with the arrays. This is the key package for Python's scientific computing.

3.1.2 Pandas

Pandas is for collecting and analyzing data. Pandas is a BSD-licensed library with an open source library that offers greater-performance, simple-to-use data structures for Python-language data analytics tools.

3.1.3 Matplotlib

Matplotlib is a Python 2D graphics library that manufactures performance figures in a number of physical copy representations and on-platform interactive environments.

3.1.4 Seaborn

Seaborn is a matplotlib-based framework for Python data visualisation. It offers an interface of high standard to draw appealing and insightful stats.

3.2 Dataset importation

data_frame = pd.read_csv('ISCX2014_Botnet_Dataset.csv')

Figure 5: Dataset importation

3.3 Additional Attributes Generation

The dataset requires some calculations to be done in order to generate certain general characteristics. The additional features are described below: The data_frame is looped and the summed for every operation. The total bytes sent in both directions

total_bytes = data['total_fvolume'] + data['total_bvolume']

Sum of the packets sent in both directions

packets_sum = data['total_fpackets'] + data['total_fpackets']

Total number multiplied by 8 bytes (1 byte = 8 bits)

total_bits = data['total_bytes'] * 8

Ratio of total Bytes and packages

bpp = data['total_bytes'] / data['total_packets']

Total bits divided by the length of the flow

bps = (data['total_bytes'] * 8)/(data['duration'] * 0.000006)

Total packages divided by the duration of the flow

pps = data['total_packets']/data['duration'] * 0.000006

Average standard deviation squared IAT

f_iat = data['std_fiat']
b_iat = data['std_biat']
f_iat = f_iat * f_iat
b_iat = b_iat * b_iat

 $var_iat = (f_iat + b_iat)/2$

Average sum of the average IAT values

f_iat = data['mean_fiat']
b_iat = data['mean_biat']

 $avg_iat = (f_iat + b_iat)/2$

Ratio between the number of packets sent in the forward direction and the total number of packets in the stream

```
pct_packets_pushed =data['total_fpackets']/data['total_packets']
```

Ratio between packet quantity in backward direction over quantity of forward direction

iopr = data['total_fpackets']/ data['total_bpackets']

Total number of bytes in the stream minus the sum of bytes of headers in both directions, then divided by the number of bundles

```
header_f = data['total_fhlen']
header_b = data['total_bhlen']
total_b = data['total_bytes']
packets = data['total_packets']
payload_length = total_b - (header_b + header_f)
avg_payload_length = payload_length / packets
```

Feature	Representation	Description				
Total Bytes	total_bytes	Total bytes sent in both directions				
Total packages	total_packets	Sum of the packets sent in both directions				
Total Bits	total bits	Total number multiplied by 8 bytes				
Total Dits		(1 byte = 8 bits)				
Bytes per packet	bpp	Ratio of total Bytes and packages				
Bytes per second	bps	Total bits divided by the length of the flow				
Deelrota por second	ppg	Total packages divided by the duration				
rackets per second	pps	of the flow				
Average IAT	avg_iat	Average sum of the average IAT values				
Average variance IAT	var_iat	Average standard deviation squared IAT				
		Ratio between the number of packets sent				
Percentage of packets sent	pct_packets_pushed	in the forward direction and the total number				
		of packets in the stream.				
IODD	ionr	Ratio between packet quantity in backward				
IOFR		direction over quantity of forward direction				
		Total number of bytes in the stream minus				
Assessed Developed Size	arm narria d langth	the sum of bytes of headers in both				
Average Fayload Size	avg_payload_length	directions, then divided by the number of				
		bundles				

 Table 1: Additional Features

3.4 Null Data

In terms of managing the data effectively, the idea of missing values is necessary to consider. Unless the missing values are not properly handled, the results may be incorrect.

```
data_frame.columns[(data_frame == 0).all()]
```

3.5 Data Balancing

```
_underscore = 6379
# Getting the number of items to be deleted
_total = len(data_frame[data_frame['label'] == 0]) - _underscore
# Getting sub-dataset to be dropped
_data_frame_underscore_index =
data_frame[data_frame['label'] == 0].head(_total).index
# deleting sub-dataset to main dataset
data_frame.drop(_data_frame_underscore_index, inplace=True)
# resetting dataset index
data_frame.reset_index(drop=True,inplace=True)
```

3.6 Data Cleaning and Exportation

data_frame.drop(['Unnamed: 0','srcip', 'srcport', 'dstip',

```
'dstport', 'proto', 'std_active', 'min_idle','mean_idle',
'max_idle',
'std_idle','furg_cnt','burg_cnt','sflow_bpackets',
'sflow_bbytes','sflow_fpackets','sflow_fbytes','dscp'],
axis=1,inplace=True)
data_frame.to_csv('traindata.csv')
```

4 Machine Learning Algorithms

This section contains the machine learning algorithms implemented in the ICT solution and their importation into the notebook.

4.1 Algorithms Importation

```
from sklearn.svm import SVC,LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
```

4.2 Metrics Importation

```
from time import time
from sklearn.metrics import precision_recall_fscore_support,
confusion_matrix,accuracy_score
```

4.3 Training Set Importation

data_frame = pd.read_csv('traindata.csv')

4.4 Dataset Division

```
X = data_frame.drop(['label'],axis=1)
y = data_frame['label']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

References