

Configuration Manual

MSc Research Project Cybersecurity

Idehen Jimmy Irivbogbe 19144814

School of Computing National College of Ireland

Supervisor; Michael Pantridge

National College of Ireland



MSc Project Submission Sheet

School of Computing

Student Name:	Irivbogbe Idehen Jimmy
Student ID:	
Programme	:Cybersecurity
Module:	Msc Internship
Lecturer:	MichaelPantridge
Due Date:	
Project Title:	Securing Internet of things (IoT) using SDN- enabled Deep learning Architecture
Word Count:	

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:

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

Idehen Iirivbogbe 19144814

1 Introduction

The configuration manual plays an important role in the reproduction of the proposed work using the algorithms of deep learning for intrusion detection in internet of things. This manual comprises the overall setup and related tools for the implementation of this work.

2 Specifications of the System

The system specifications are as follow:

- Processor: Intel Core i7, 7th generation @2.24 GHz
- GPU: Nvidia Geforce 6GB
- RAM: 16 GB
- SSD: 1 TB
- Operating System : Windows 10 professional

3 Software's/Tools

The software and tools that are used in the implementation of this work are as follow:

- Anaconda Navigator
- Python
- Spyder
- Origin

3.1 Software Installation

The step by step installation of the software are given below: Anaconda navigator can be downloaded from the given link (https://www.anaconda.com/products/individual)

As this work has used the windows version of the anaconda. So Anaconda for windows has been downloaded.

O ANACONDA.

Documentation Blog Contact Q. What is Anaconda? Products Support Resources About Downloads

	Version 5.3 Release Date: September 28, 2018	
	Download For: 📑 💣 👌	
High-Performance Distribution	Package Management	Portal to Data Science
Easily install 1,400+ <u>data science</u> <u>packages</u>	Manage packages, dependencies and environments with <u>conda</u>	Uncover insights in your data and create interactive visualizations
	Windows d macOS & Linux	

Figure 1: Downloading Anaconda

-		License Agreeme	nt		
)	ANACONDA	Please review the lie 5.3.0 (64-bit).	cense terms be	fore installing Anaco	nda3
Pres	s Page Down to see t	he rest of the agreeme	nt.		
===					
Ana ===	conda End User Licen	se Agreement ============	====		H
Copy	yright 2015, Anacond	a, Inc.			
All r	ights reserved under	the 3-clause BSD Licen:	se:		
Redi perr	istribution and use in nitted provided that t	source and binary form ne following conditions	s, with or witho are met:	out modification, are	
	- 1				Ψ.
If yo agre	u accept the terms of ement to install Anac	the agreement, click I and a 5.3.0 (64-bit).	Agree to contin	ue. You must accept	the
NEW R					

Figure 2: Anaconda Installation

Anaconda3 5.3.0 (64-b	t) Setup	
	Select Installation Type Please select the type of installati for Anaconda3 5.3.0 (64-bit).	on you would like to perform
Install for:		
Just Me (recommende)	d)	
O All Users (requires add	nin privileges)	
naconda, Inc		
	< Back	Next > Cancel

Figure 3: Anaconda Installation After selection of the installation type(Just Me) click on the next button.

Anacondas 2021.05 (04-	bit) Setup			\times
O ANACONDA.	Choose Install Location Choose the folder in which to install /	Anaconda3 3	2021.05 (64	⊢bit).
Setup will install Anaconda: folder, click Browse and sel	3 2021.05 (64-bit) in the following folder ect another folder. Click Next to continu	r. To install i ue.	n a differen	t
Destination Folder	2			
C: Users (np \anaconda	5	Br	owse	

Figure 4; Path Selection

O Anaconda3 2021.05 (64-1	bit) Setup	<u>elandaanidaan</u>		-	X
O ANACONDA.	Choose Insta Choose the fo	all Location older in which to ins	tall Anaconda	3 2021.0	5 (64-bit).
Setup will install Anaconda: folder, click Browse and sel	3 2021.05 (64-bi ect another fold	it) in the following fo er. Click Next to cor	older. To insta ntinue.	all in a diff	erent
Destination Folder C:\Users\hp\anaconda	3\			Browse	
Space required: 2.9GB					
Space available: 3.4GB					
Anaconda, Inc. ————					
		< Back	Next >		Cancel

Figure 5: Setting default python

O Anaconda3 5.3.1 (64-bit) Setup	
	Installation Complete Setup was completed successfully.	
Completed		
Show details		
Anaconda, Inc		
	< Back	Next > (Cancel

Figure 6: Setup completed

Further, you can download the python 3.9.0 from the below link (https://www.python.org/downloads/)



Figure 7: Downloading Python 3.9.0





Close

Figure 9: Python 3.9.0 Installation Competed



Figure 10: Python 3.9.0 Installation Confirmation

4 Implementation

The libraries used for the purpose of implementation are as follow:

- Pandas
- Numpy
- Sklearn
- Keras

For a proper experimentation. All of the libraries has been imported using spyder IDE.

```
1
      import pandas as pd
 2
      import numpy as np
 3
      from sklearn import preprocessing
 4
      from sklearn.preprocessing import MinMaxScaler
 5
      from sklearn.preprocessing import StandardScaler
      from sklearn.model_selection import train_test_split
 6
 7
      from keras.utils import to_categorical
 8
9
      from keras.models import Sequential
      from keras.layers import LSTM
10
      from keras.layers import Dense , Dropout
11
      from sklearn.metrics import accuracy_score
12
      from sklearn.metrics import confusion_matrix
13
      from sklearn.metrics import classification_report
14
     from keras.layers import CuDNNLSTM, Activation
15
```

Figure 11: Importing Libraries and packages

1. Dataset Loading

```
15 from keras.layers import CuDNNLSTM, Activation
16
17 #importing dataset
18 mydata=pd.read_csv("dataset.csv")
19
20 mydata = mydata.sample(frac = 1).reset_index(drop = True)
21
22 #mydata.fillna(0, inplace=True)
23 #feature extraction
```



2. Preprocessing

```
22
      #mydata.fillna(0, inplace=True)
23
      #feature extraction
24
      X=mydata.iloc[:, 0:79]
25
      Y=mydata.iloc[:,79:80]
26
      #
27
      sc=StandardScaler()
      X = sc.fit_transform(X)
28
29
      #
30
     #min_max_scaler = preprocessing.MinMaxScaler()
31
     #X = min max scaler.fit transform(X)
32
33
      x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.20)
34
35
      #min max scaler = preprocessing.MinMaxScaler()
36
      #X = min_max_scaler.fit_transform(X)
37
38
      #min_max_scaler = preprocessing.MinMaxScaler()
39
      #x_train = min_max_scaler.fit_transform(x_train)
40
41
      x_train = np.array(x_train)
      x_train = np.reshape (x_train,(x_train.shape[0], 1, x_train.shape[1]))
42
43
```

Figure 13: Dataset Preprocessing

3. One hot Encoding



4. Building Model

```
55
        model = Sequential()
56
        model.add(CuDNNLSTM(300, input_shape= (x_train.shape[1],x_train.shape[2]), return_sequences=True))
#model.add(CuDNNLSTM(650, return_sequences=True))
57
58
59
        #model.add(CuDNNLSTM(600, return_sequences=True))
60
        model.add(CuDNNLSTM(200, return_sequences=False))
61
        model.add(Dropout(0.4))
62
63
        #model.add(CuDNNLSTM(300, return_sequences=False))
64
65
        model.add(Activation('relu'))
        #model.add(LSTM(200, activation= 'relu', return_sequences=True))
#model.add(LSTM(150, activation= 'relu', return_sequences=True))
#model.add(LSTM(100, activation= 'relu', return_sequences=False))
66
67
68
69
70
71
72
73
74
75
76
77
        model.add(Dense(200, activation='relu'))
#model.add(Dense(150, activation='relu'))
        model.add(Dropout(0.2))
        model.add(Dense(100, activation='relu'))
model.add(Dense(50, activation='relu'))
78
        model.add(Dense(Num_classes, activation='softmax'))
79
80
```



5. Optimizer Adding

79			
80			
81	#from keras import optimizers		
82	#import keras		
83	#import keras.utils		
84	#from keras import utils as np_utils		
85	#keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, decay=	Ð.Ø,	amsgrad=False)
86			
87			
88	<pre>model.compile(optimizer='adamax', loss='binary_crossentropy', metrics=['accurac</pre>	y'])	
89	<pre>#model.summary()</pre>		
90			

Figure 16: Adding optimizer

6. Model Training

```
89
       #model.summary()
90
91
       import time
92
       start = time.clock()
93
94
       #train the model
95
96
       model.fit(x_train, y_train_ohe, epochs=5, batch_size=64)
97
98
       end = time.clock()
99
       print ("%.2gs" % (end-start))
       #predections
100
101
```

Figure 17: Training the model

7. Accuracy and Confusion Matrix

```
109
110
111 print("Accuracy:" + str(accuracy_score(y_test, prediction)* 100))
112 print(classification_report(y_test, prediction))
113 cm = confusion_matrix(y_test, prediction)
114 print (cm)
115
```

Figure 18: printing accuracy and confusion matrix

8. Calculating TP,FP,FN,TN

```
TTD
116
       #Pred = np.round(Pred)
117
       FP = cm.sum(axis=0) - np.diag(cm)
118
       FN = cm.sum(axis=1) - np.diag(cm)
       TP = np.diag(cm)
119
120
       TN = cm.sum() - (FP + FN + TP)
121
122
123
       FP = FP.astype(float)
124
       FN = FN.astype(float)
125
       TP = TP.astype(float)
126
       TN = TN.astype(float)
127
128
129
       # Sensitivity, hit rate, recall, or true positive rate
130
       TPR = TP/(TP+FN)
       # Specificity or true negative rate
131
     TNR = TN/(TN+FP)
132
                          Figure 19: Calculating FP, FN, TP, TN
```

9. Calculating overall accuracy, Precision

```
129
       # Sensitivity, hit rate, recall, or true positive rate
130
      TPR = TP/(TP+FN)
      # Specificity or true negative rate
131
132
      TNR = TN/(TN+FP)
      # Precision or positive predictive value
133
134
      PPV = TP/(TP+FP)
      # Negative predictive value
135
136
     NPV = TN/(TN+FN)
137
      # Fall out or false positive rate
138
     FPR = FP/(FP+TN)
139
     # False negative rate
140 FNR = FN/(TP+FN)
141 # False discovery rate
142
     FDR = FP/(TP+FP)
143
144
     # Overall accuracy
145
     ACC = (TP+TN)/(TP+FP+FN+TN)
      # Precision
146
147
      Precision = TP / TP + FP
1/0
                  Figure 20: Calculating Accuracy, Precision
```

5 Results

Table 1. Results and Evaluation								
Algorithm	Accuracy	Precision	Tpr	Fpr	F1-score	Recall		
DNNLSTM	99.92%	99.30%	99.87%	0.0070%	99.50%	99.50%		
GRU	99.79%	99.33%	99.77%	0.0090%	99.50%	99.83%		
LSTMGRU	99.11%	99.16%	99.77%	0.012%	90.10%	88.19%		