Configuration Manual

MSc Research Project
Cloud Computing

Phani Kumar Kalyanadurgam ChandraShekhar
Student ID: 21175098

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Phani Kumar Kalyanadurgam Chandrashekhar…..……………………………
Student ID:	21175098……………………………………………………………………….……
Programme:	MSc in Cloud Computing……………………… **Year:** 2023………………
Module:	Research Project………………………………………………………………….………
Lecturer:	21/12/2023…………………………………………………………………….………
Submission Due Date:	………………………………………………………………………………….………
Project Title:	A Hybrid approach in detecting DDOS attacks in Software-Defined-Networks
Word Count:	1211…………………………… **Page Count:** 11…………………………….…….………

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.
ALL 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: Phani Kumar Kalyanadurgam Chandrashekhar.……………………………

Date: 21/12/2023…………………………………………………………………………………

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

Phani Kumar Kalyanadurgam Chandrashekhar
Student ID: 21175098

1 Introduction

SDN (Software-Defined Networking): SDN is a network architecture that divides the control plane from the data plane, allowing for centralized control and programmability. Software-Defined Networking (SDN) involves the abstraction and relocation of network information to a controller based on software, resulting in a network management strategy that is more adaptable and responsive (Badotra, sumit (2017)).

Ryu Controller: Ryu is a Python-based SDN controller that is available as open-source. The controller adheres to the OpenFlow protocol and functions as the central intelligence of a software-defined network (SDN), overseeing flow control to provide a network architecture that is both customizable and scalable. Ryu is very adaptable, enabling developers to construct tailored SDN applications and network services.

The OpenFlow SDN Protocol: It is a standardized communication protocol that facilitates communication between the SDN controller and network devices, such as switches and routers. The purpose of this is to specify the manner in which the controller may alter the functioning of network devices by making adjustments to flow tables. OpenFlow facilitates the real-time adjustment and management of network data flow, establishing it as a crucial protocol in software-defined networking (SDN) settings.

Mininet: It is a network emulator that is open source that enables the building of a virtual network on a single workstation. It offers a nimble, expandable setting for evaluating SDN applications without requiring tangible equipment. Mininet enables the construction of a network structure including of hosts, switches, and controllers, making it a perfect instrument for the development and evaluation of Software-Defined Networking (SDN).

Jupyter: It is a freely available online program that allows users to create and share live code, equations, visualizations, and narrative prose. It provides support for many programming languages, including Python, which is often used in SDN development. Jupyter Notebooks provide a dynamic and collaborative setting, which makes them highly suitable for executing and documenting SDN initiatives.

2 Simulation setup

This section briefs about the steps to be taken to install all the packages and software required to run the project.

The platform must be setup on Ubuntu 20.04.6 LTS operating system and the following tools must be installed to run the simulation.

```
Ubuntu 20.04.6 LTS phani tty1

phani login: phani
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-169-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information disabled due to load higher than 2.0


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

14 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Sun Dec 17 23:52:51 UTC 2023 from 192.168.101.1 on pts/1
phani@phani:~$
```

Figure1: Ubuntu operating system

- Before going ahead with the installation of tools make sure you have updated the Linux OS and libraries with python 3.8.
- Open terminal and run the below commands.

-sudo apt-get update
-sudo apt-get upgrade
-sudo apt install python3

Open flow protocol for SDN: OpenVswitch must be installed in the OS as it is the standard protocol used for communication in SDN.
-sudo apt-get install openvswitch-switch

- Give Y wherever it asks and and check for the version to be sure if its installed.

-ovs-vsctl –version

```
phani@phani:~$ ovs-vsctl --version
ovs-vsctl (Open vSwitch) 2.13.8
DB Schema 8.2.0
phani@phani:~$ _
```

Figure2: OpenVswitch check

Ryu Controller: In order to install the ryu controller, it is necessary to install the PIP for installing Python packages. Since ryu is a controller based on Python, it must be installed via PIP. To install PIP and Ryu controller, do the following instructions in the terminal.

-sudo apt install python3-pip
-pip3-version

```
phani@phani:~$ pip3 --version
pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
phani@phani:~$ _
```

Figure3: Pip check

-sudo pip3 install ryu
-ryu-manager –version

Figure4: Ryu Check

Mininet: It is a network simulator and it creates virtual network topology for SDN. To install mininet follow the below instructions.

-sudo apt-get install mininet
-mn –version

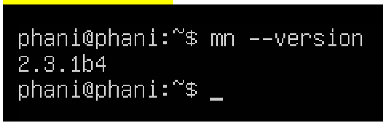

Figure5: Mininet check

3 Traffic data collection.

- Navigate to the folder structure where all the code files are present.

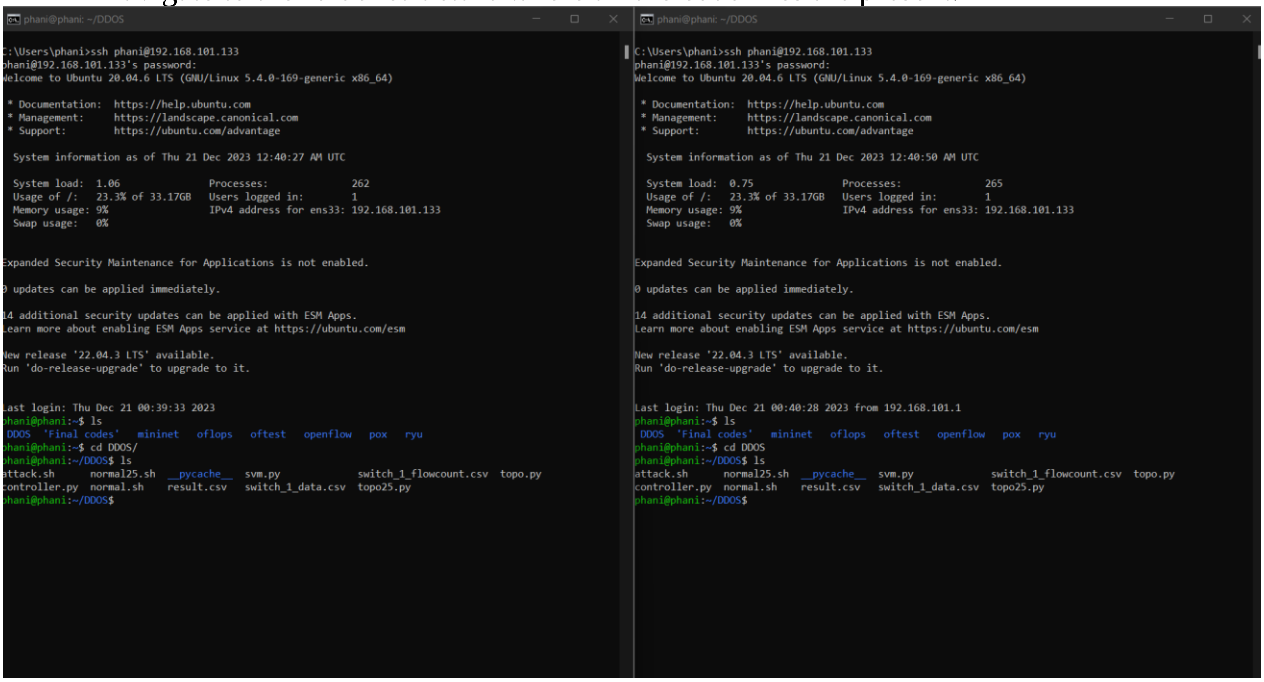

Figure6: Folder structure check.

- Open controller.py in one terminal and check if the APP_TYPE and TEST_TYPE is set to 0 to generate the normal traffic, also make sure that in topo.py file the TEST_TYPE is normal.

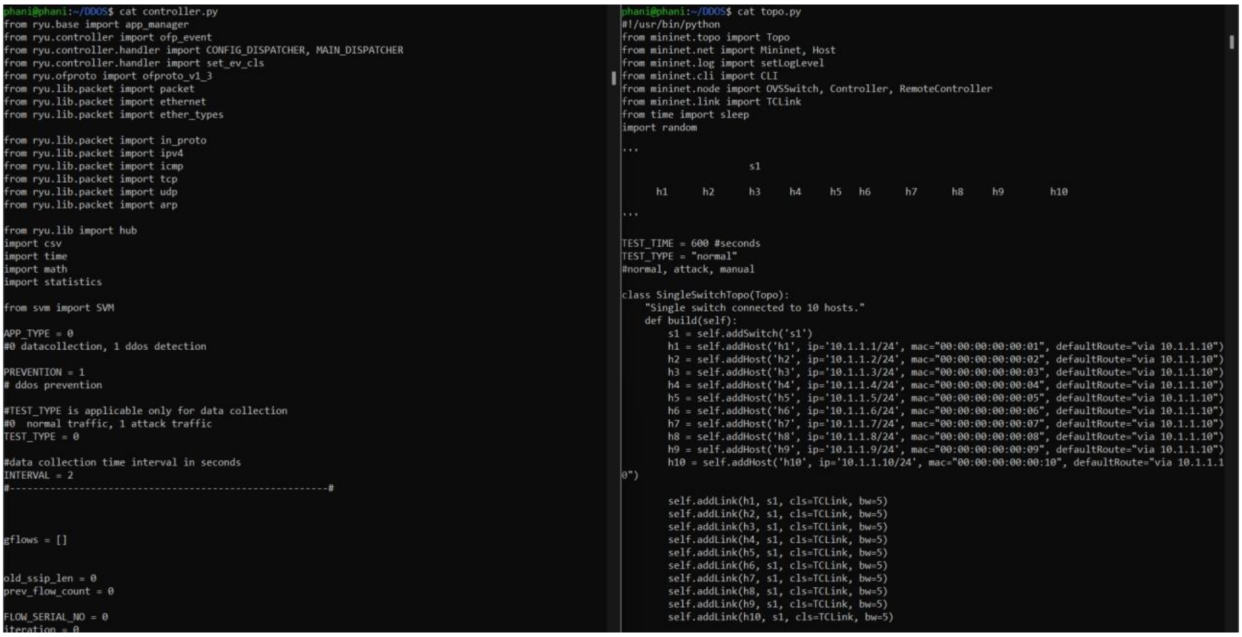

Figure7: Check for controller.py and topo.py file normal traffic data generation.

- Run the controller.py file in one terminal and topo.py file in another terminal for the normal traffic generation.

-ryu-manager controller.py

-sudo python3 topo.py

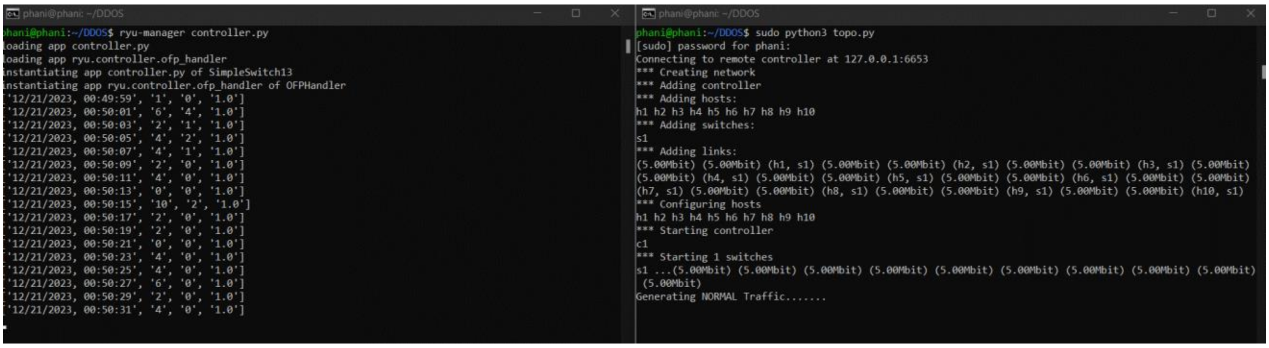

Figure8: Normal traffic generation.

- Now lets generate the Traffic data, to do this now we have to change the APP_TYPE=0 and TEST_TYPE=1 in controller.py file and TEST_TYPE=’attack’ in topo.py file.

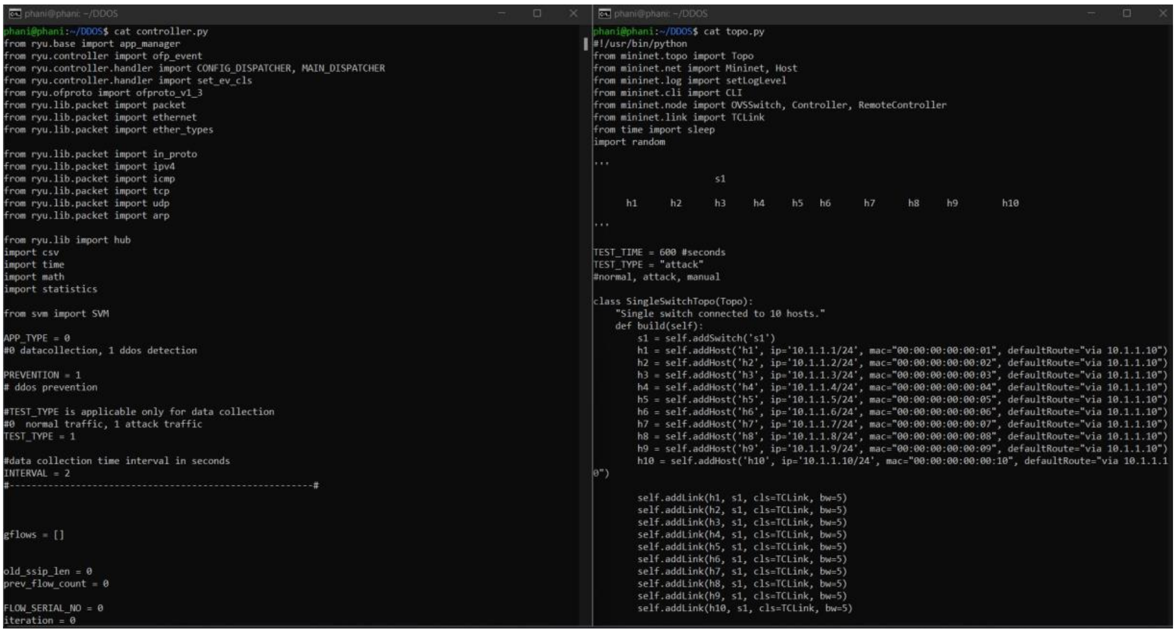

Figure9: Check for controller.py and topo.py file for attack data generation.

- Follow the same steps to run the files this time to generate the traffic data.

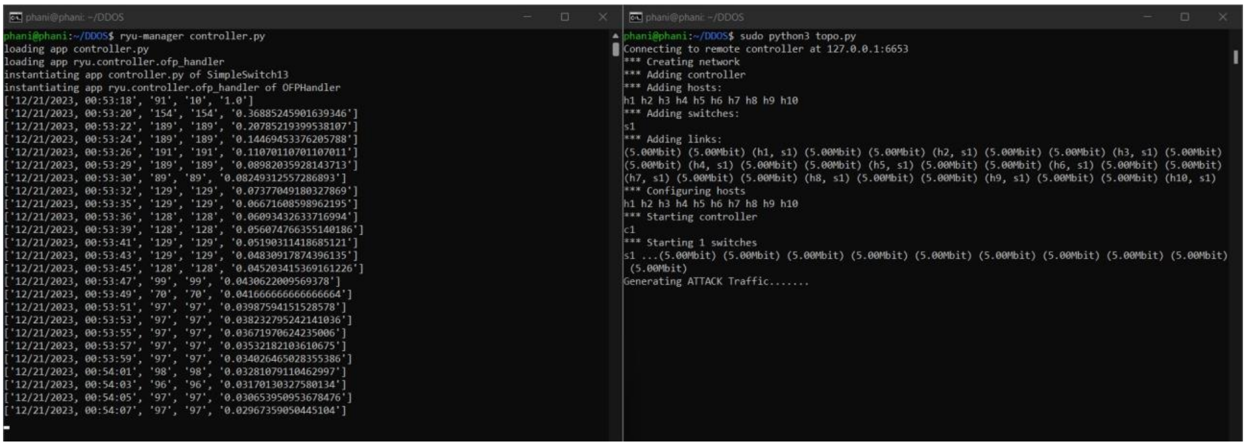

Figure10: Attack traffic generation.

4. Data processing and Machine learning model development.

- To start of with open the jupyter notebook, either in your local or in AWS. You can either run the python notebooks in google colab and AWS sage maker.

- Create a folder in the jupyter and then upload the CSV files and the notebook files which is zipped and shared.

Figure11: Files upload to the jupyter.

- Open the time-series-of-ddos-ml.ipynb file and click on run all option this will run the whole code which is present in the file, in this file basically we are trying to find the starting point of the DDOS attack and once we get that starting trigger we then classify it using the traditional machine learning algorithms to conclude if it's a DDOS attack or not.

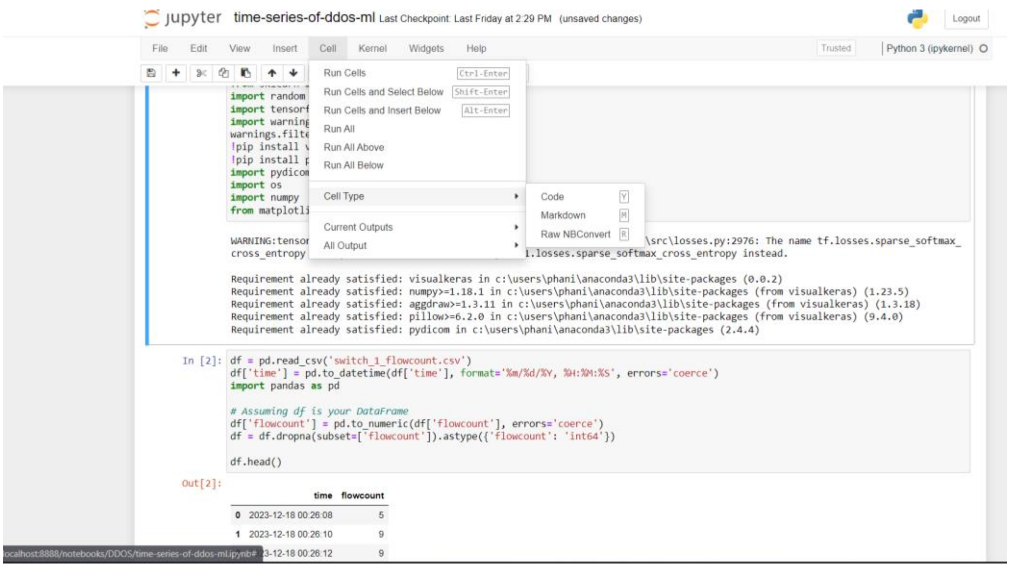

Figure12: Notebook file to run all in a single click.

- Once all the cells are executed, the results will be generated in the form of graphs and tables. Graphs will showcase the confusion matrix and the accuracy alongside the tables will showcase the evaluation metrics and performance.
- Open the other notebook file which is ddos-malware-ml.ipynb file, this file is used for classifying the attack as malicious or a general traffic to a server.

5. RESULTS

Model	TN	FP	FN	TP	Accuracy	Precision	Recall or Sensitivity	F1 Score	Specificity
DecisionTreeClassifier	57.8	0.0	8.5	37.7	0.918269	1.0	0.814956	0.897016	1.0
GaussianNB	57.8	0.0	0.0	46.2	1.000000	1.0	1.000000	1.000000	1.0
GradientBoostingClassifier	57.8	0.0	31.8	14.4	0.694231	1.0	0.309991	0.471160	1.0
LogisticRegression	57.8	0.0	46.2	0.0	0.555769	NaN	0.000000	NaN	1.0
RandomForestClassifier	57.8	0.0	31.8	14.4	0.694231	1.0	0.309991	0.471160	1.0

Figure13: Evaluation metrics

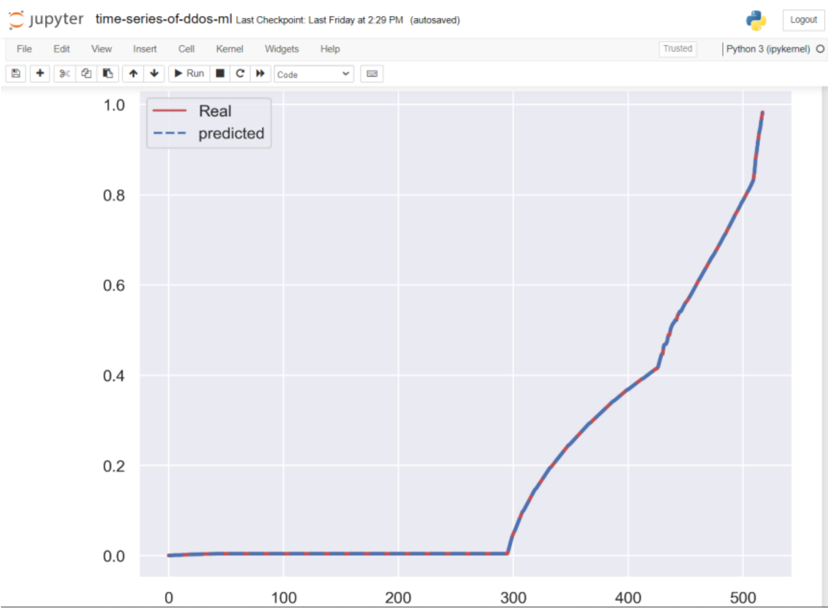

Figure14: Decision tree regressor

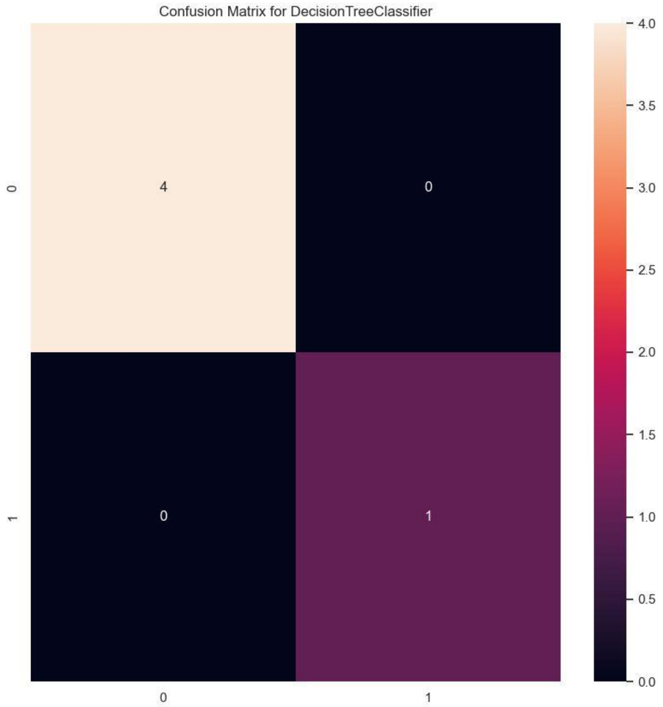

Figure15: Confusion matrix of random forest classifier

Out[22]:

Model	SPlit	TN	FP	FN	TP	Accuracy	Precision	Recall or Sensitivity	F1 Score	Specificity
DecisionTreeClassifier	Test	297.0	0.0	0.0	223.0	1.000000	1.0	1.000000	1.000000	1.0
	Train	4.0	0.0	0.0	1.0	1.000000	1.0	1.000000	1.000000	1.0
GaussianNB	Test	297.0	0.0	0.0	223.0	1.000000	1.0	1.000000	1.000000	1.0
	Train	4.0	0.0	0.0	1.0	1.000000	1.0	1.000000	1.000000	1.0
GradientBoostingClassifier	Test	297.0	0.0	149.0	74.0	0.713462	1.0	0.331839	0.498316	1.0
	Train	4.0	0.0	0.0	1.0	1.000000	1.0	1.000000	1.000000	1.0
LogisticRegression	Test	297.0	0.0	223.0	0.0	0.571154	NaN	0.000000	NaN	1.0
	Train	4.0	0.0	1.0	0.0	0.800000	NaN	0.000000	NaN	1.0
RandomForestClassifier	Test	297.0	0.0	205.0	18.0	0.605769	1.0	0.080717	0.149378	1.0
	Train	4.0	0.0	0.0	1.0	1.000000	1.0	1.000000	1.000000	1.0

Figure16: Evaluation metrics

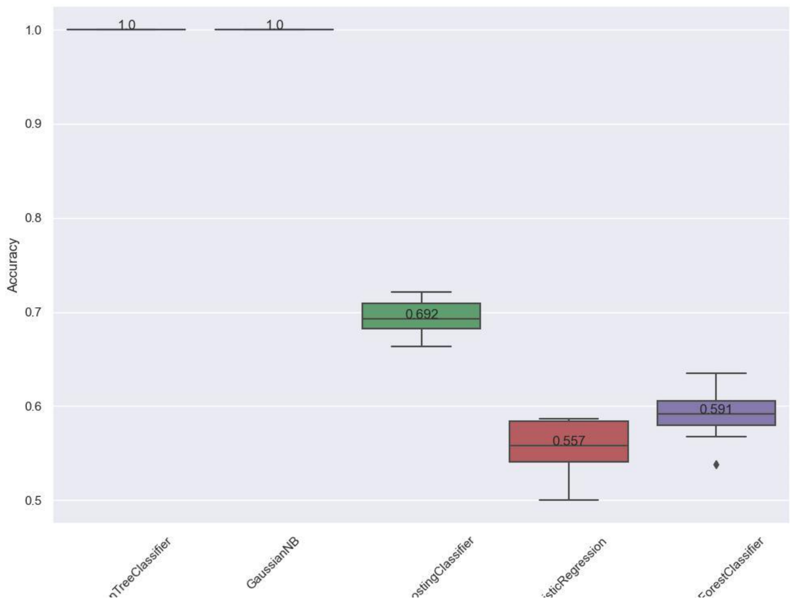

Figure17: Accuracy graph of machine learning algorithms

References

Badotra, Sumit. (2017). A Review Paper on Software Defined Networking. International Journal of Advanced Computer Research. 8.