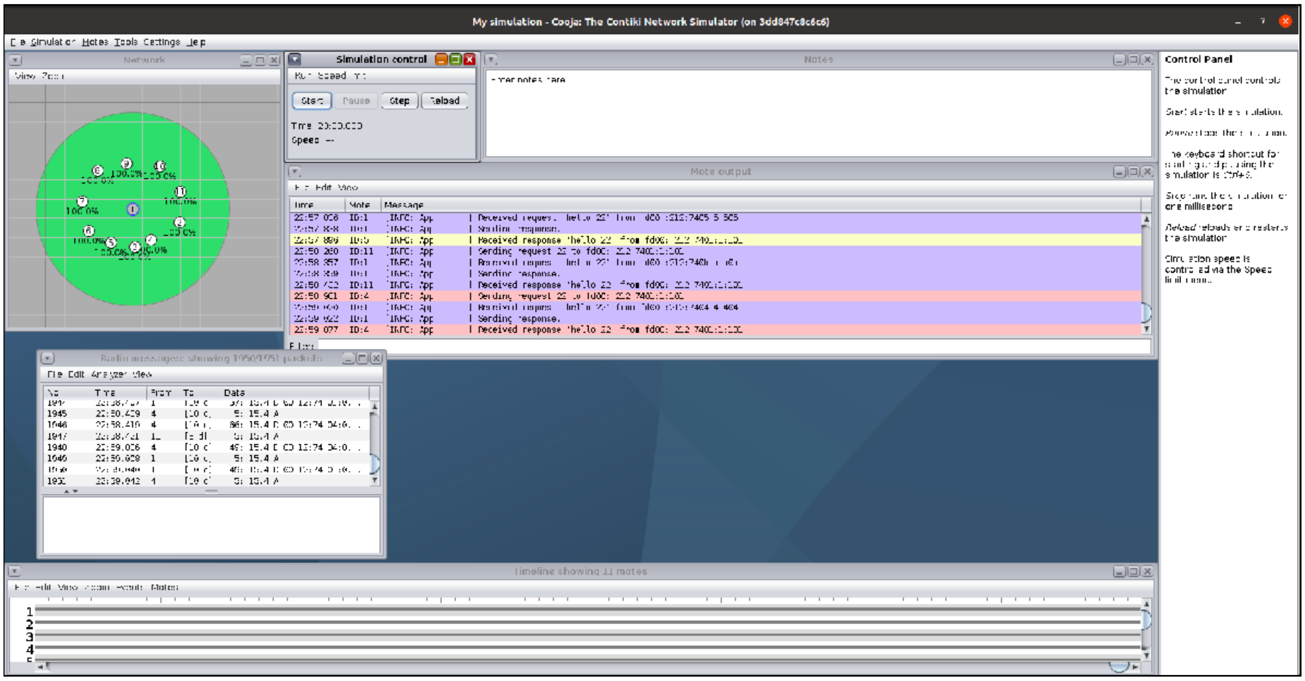
Assignment Four
Contiki-NG Wireless Sensor Network - Network Attacks
Group 3

AHMED SHEHATA MAHMOUD ABOMOUSTAFA
SARAH HOSSAM ABDELHAMEED ELMOWAFY
MOHAMED SAYED ABDELWAHAB HUSSIEN
ABDELMEGEED AHMED ABDELMEGEED
[UNIVERSITY OF OTTAWA]

1. How do you implement the two network attacks?

a. Normal Network

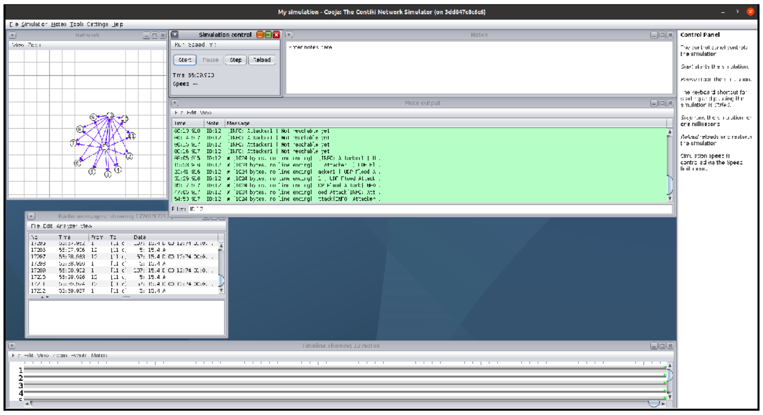
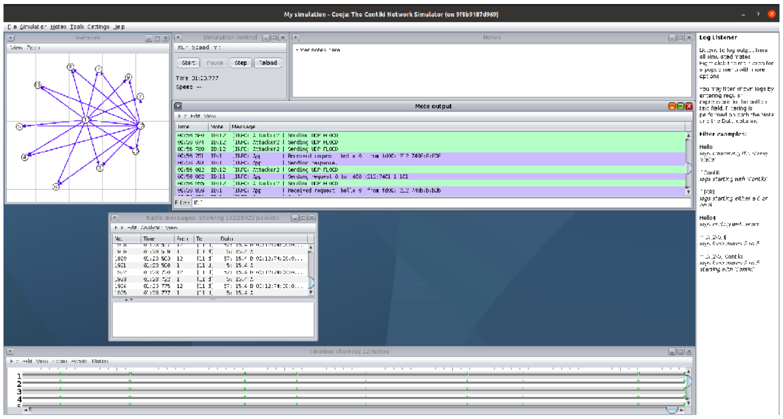
i. First, We insert a node to represent the server.
ii. Second, We insert 10 nodes to represent the clients.
iii. Then, We Start the network to send and receive packets and generate data with .pcap files.


My simulation - Cooja: The Contiki Network Simulator (on 3dd847c8c6c6)
Network
Simulation control
Start
Pause
Step
Reload
Notes
Control Panel
Mote output
File Edit View
Radio messages: showing 1950/1951 packets
File Edit Analyzer View
Timeline showing 11 motes

```
Time       Mote   Message
22:57 006  ID:1   [INFO: App      ] Received request 'hello 22' from fd00::212:7405:5:505
22:57 8?8  ID:1   [INFO: App      ] Sending response.
22:57 889  ID:5   [INFO: App      ] Received response 'hello 22' from fd00::212:7401:1:101
22:50 260  ID:11  [INFO: App      ] Sending request 22 to fd00::212:7401:1:101
22:58 357  ID:1   [INFO: App      ] Received request 'hello 22' from fd00::212:740b:b:b0b
22:58 3?9  ID:1   [INFO: App      ] Sending response.
22:50 ?22  ID:11  [INFO: App      ] Received response 'hello 22' from fd00::212:7401:1:101
22:50 901  ID:4   [INFO: App      ] Sending request 22 to fd00::212:7401:1:101
22:59 ???  ID:1   [INFO: App      ] Received request 'hello 22' from fd00::212:7404:4:404
22:59 622  ID:1   [INFO: App      ] Sending response.
22:59 077  ID:4   [INFO: App      ] Received response 'hello 22' from fd00::212:7401:1:101
```

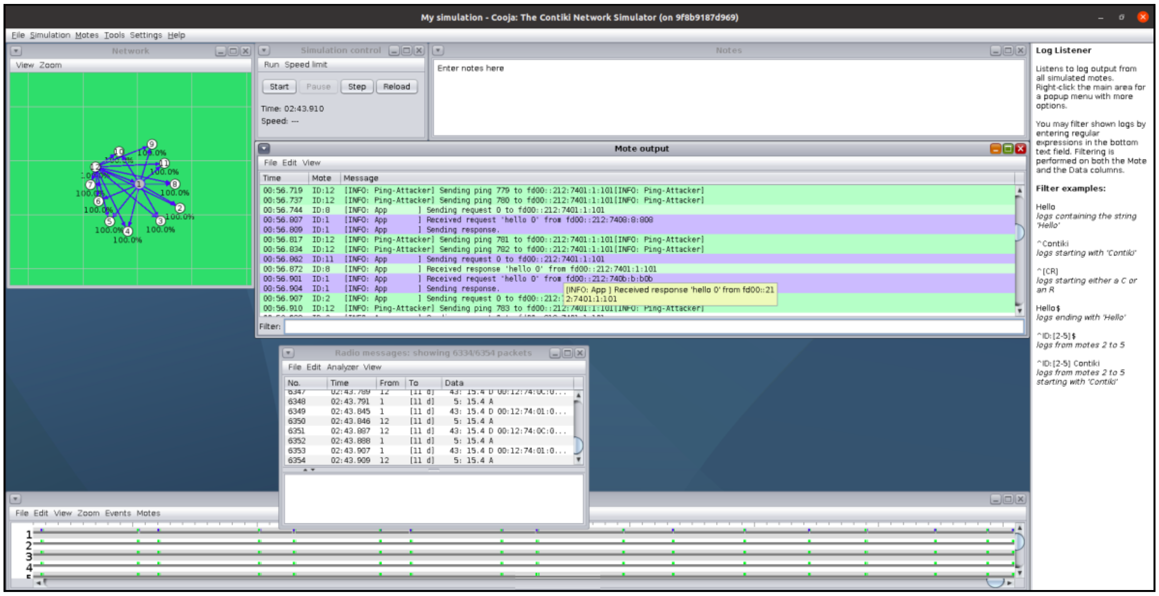
b. Attack 2 Network

i. First, we insert a node to represent the server.
ii. Second, we insert 10 nodes to represent the clients.
iii. Third, we insert 1 node to represent the UDP flood attacker, Attacker1 to take 6LowPAN packet from it.
iv. Finally, we take the packet and using it in attacker2 and run the simulation again and provide .pcap file.

Attacker1	Attacker2
	

c. Ping Attack Network

i. First, We insert a node to represent the server.
ii. Second, We insert 10 nodes to represent the clients.
iii. Third, We insert 1 node to represent the ping attacker.


My simulation - Cooja: The Contiki Network Simulator (on 9f8b9187d969)
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start
Pause
Step
Reload
Notes
Enter notes here
Log Listener
Mote output
File Edit View
Radio messages: showing 6334/6354 packets
File Edit Analyzer View
File Edit View Zoom Events Motes

```
Time: 02:43.910
Speed: ---
```

```
Time      Mote   Message
00:56.719 ID:12  [INFO: Ping-Attacker] Sending ping 779 to fd00::212:7401:1:101[INFO: Ping-Attacker]
00:56.737 ID:12  [INFO: Ping-Attacker] Sending ping 780 to fd00::212:7401:1:101[INFO: Ping-Attacker]
00:56.744 ID:8   [INFO: App        ] Sending request 0 to fd00::212:7401:1:101
00:56.807 ID:1   [INFO: App        ] Received request 'hello 0' from fd00::212:7408:8:808
00:56.809 ID:1   [INFO: App        ] Sending response.
00:56.817 ID:12  [INFO: Ping-Attacker] Sending ping 781 to fd00::212:7401:1:101[INFO: Ping-Attacker]
00:56.834 ID:12  [INFO: Ping-Attacker] Sending ping 782 to fd00::212:7401:1:101[INFO: Ping-Attacker]
00:56.862 ID:11  [INFO: App        ] Sending request 0 to fd00::212:7401:1:101
00:56.872 ID:8   [INFO: App        ] Received response 'hello 0' from fd00::212:7401:1:101
00:56.901 ID:1   [INFO: App        ] Received request 'hello 0' from fd00::212:740b:b:b0b
00:56.904 ID:1   [INFO: App        ] Sending response.
00:56.907 ID:2   [INFO: App        ] Sending request 0 to fd00::212:
00:56.910 ID:12  [INFO: Ping-Attacker] Sending ping 783 to fd00::212:7401:1:101[INFO: Ping-Attacker]
Filter:
```

```
[INFO: App ] Received response 'hello 0' from fd00::21
2:7401:1:101
```

```
No.   Time       From  To      Data
6347  02:43.789  12    [11 d]  43: 15.4 D 00:12:74:0C:0...
6348  02:43.791  1     [11 d]   5: 15.4 A
6349  02:43.845  1     [11 d]  43: 15.4 D 00:12:74:01:0...
6350  02:43.846  12    [11 d]   5: 15.4 A
6351  02:43.887  12    [11 d]  43: 15.4 D 00:12:74:0C:0...
6352  02:43.888  1     [11 d]   5: 15.4 A
6353  02:43.907  1     [11 d]  43: 15.4 D 00:12:74:01:0...
6354  02:43.909  12    [11 d]   5: 15.4 A
```

2. How do you generate the dataset?

- We have .pcap files, and we used feature extraction for converting it to .csv files by NetworkFlowMeter package.
- Then, We give label to the three files generated after converting the three .pcap files as following:-
    - Normal network takes label **normal**
    - Attack2 takes two labels, normal and attack1, by filtering the dataset by server as,
        - Any packet with **Src IP = ::212:7401:1:101**, I label it as **attack1**
        - Else, We label it as **normal**
    - The same thing with ping attack,
        - Any packet with **Src IP = ::212:7401:1:101**, I label it as **ping attack**
        - Else, We label it as **normal**

- Then, I drop the network hardware features
    - Session Key
    - Src IP
    - Dst IP
    - Init Ts
    - Ts
    - Protocol
    - Src Port
    - Dst Port
    - Last Ts
    - Mac Addr
- Then, I split the data to train and test for modeling using sklearn **train_test_split** method
    - Make 80% for training
    - And 20% for testing

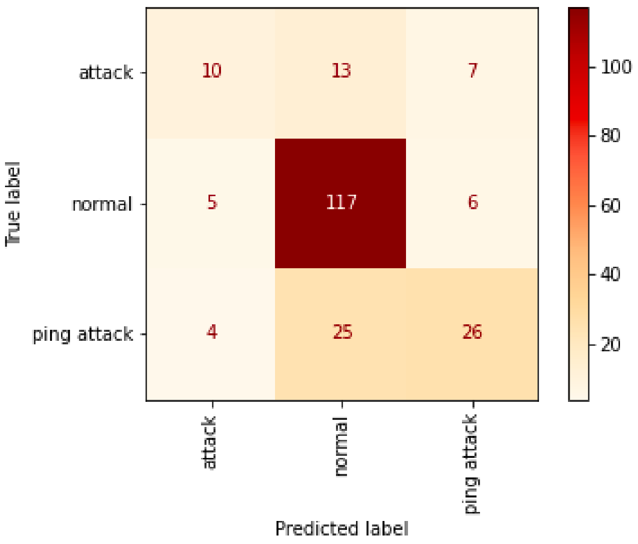
3. Compare the performances of the three machine learning algorithms.

Do not forget to Include the macro F1 score, precision, recall, accuracy and confusion matrices.

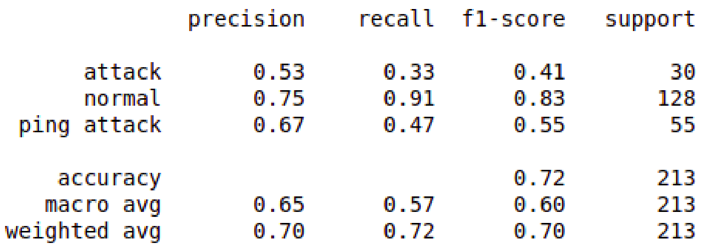
- After we applied the three machine learning models on our dataset:-

1. Random Forest Classifier

<table>
<tr><th>Confusion Matrix</th><th>Classification Report</th></tr>
<tr><td>



</td><td>



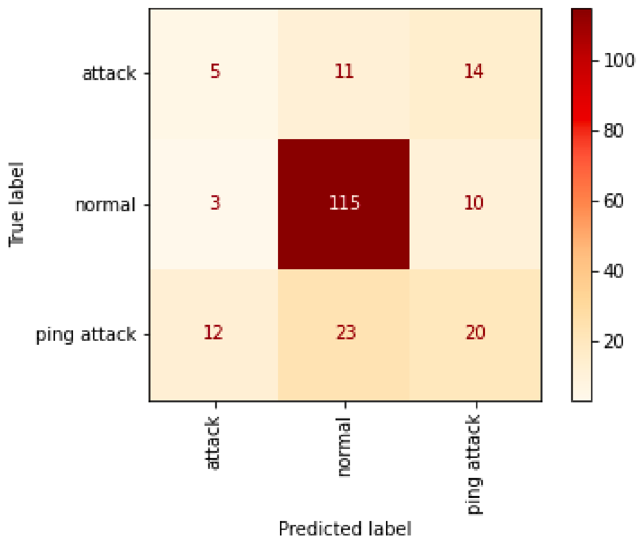
```
              precision    recall  f1-score   support

      attack       0.53      0.33      0.41        30
      normal       0.75      0.91      0.83       128
 ping attack       0.67      0.47      0.55        55

    accuracy                           0.72       213
   macro avg       0.65      0.57      0.60       213
weighted avg       0.70      0.72      0.70       213
```

- **Testing Accuracy :** 0.718
- **Precision Score:** 0.700
- **Recall Score:** 0.718
- **F1 Score:** 0.697
- **Accuracy:** 0.72

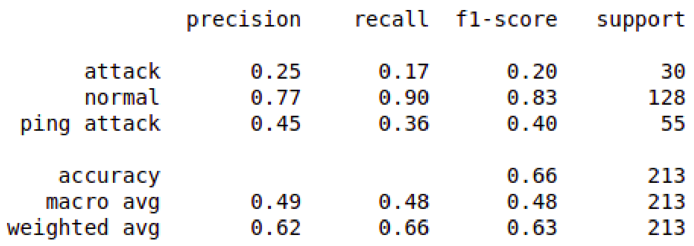
</td></tr>
</table>

2. Ada boosting

Confusion Matrix



Classification Report

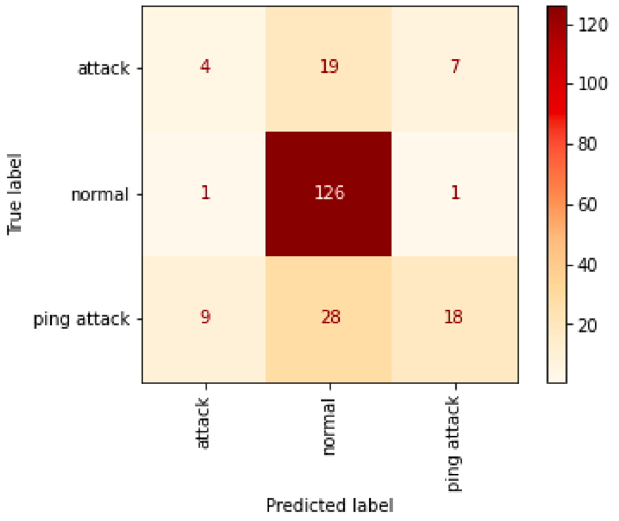


```
              precision    recall  f1-score   support

      attack       0.25      0.17      0.20        30
      normal       0.77      0.90      0.83       128
 ping attack       0.45      0.36      0.40        55

    accuracy                           0.66       213
   macro avg       0.49      0.48      0.48       213
weighted avg       0.62      0.66      0.63       213
```

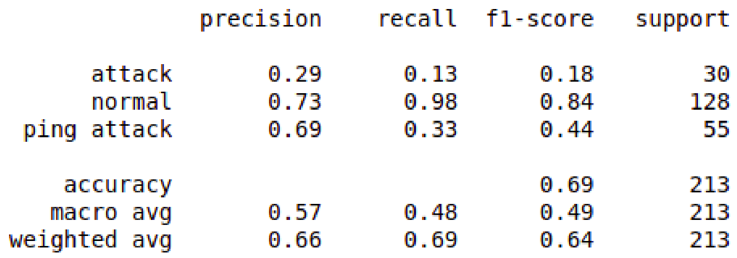
- **Testing Accuracy :** 0.657
- **Precision Score:** 0.616
- **Recall Score:** 0.657
- **F1 Score:** 0.631
- **Accuracy:** 0.66

3. Naive Bayes

Confusion Matrix



Classification Report



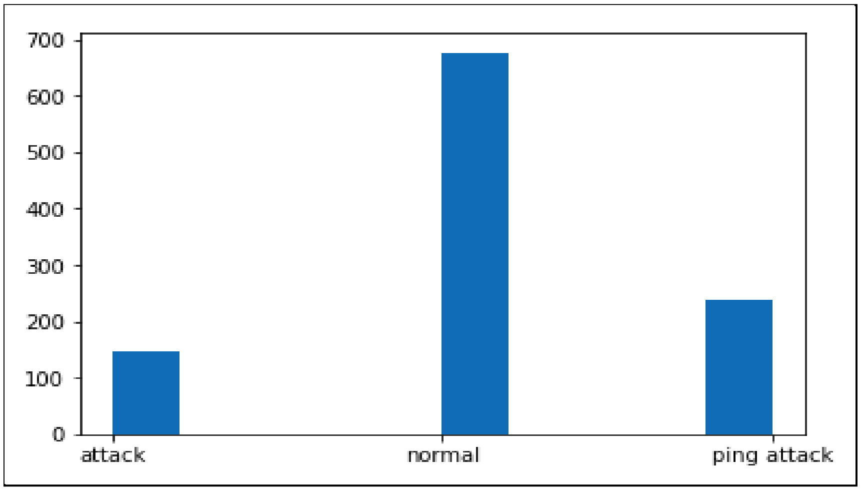
```
              precision    recall  f1-score   support

      attack       0.29      0.13      0.18        30
      normal       0.73      0.98      0.84       128
 ping attack       0.69      0.33      0.44        55

    accuracy                           0.69       213
   macro avg       0.57      0.48      0.49       213
weighted avg       0.66      0.69      0.64       213
```

- **Testing Accuracy :** 0.695
- **Precision Score:** 0.657
- **Recall Score:** 0.695
- **F1 Score:** 0.643
- **Accuracy:** 0.69

- From that, we see that **Random Forest** give the best accuracy on this data according to all metrics.

4. Conclusion

- As our final conclusion, we conclude that, in the situation of imbalanced data like the data we have, most of the machine learning algorithms do not do well with imbalanced data as we see in the graph.



- As we see there is an imbalance in the data, and the data biased toward the **normal** class, and this obvious in confusion matrix that the most predicted true class is the normal class.
- As Random forest is an extension of bagging that randomly selects subsets of features used in each data sample, so it gives us the best accuracy.
- Because we need a robust classifier to classify the attack with faster manner, we need to use robust algorithms as deep learning, and use more data to make the algorithm able to learn in a good way.
- The main goal of the **UDP flood attack** is to flood random ports on a remote host.
- **Ping Flood Attack** is another kind of Denial of Service (DoS) attack where the attacker tries using ICMP requests to drain the server's resources and cause blocking for normal network traffic.