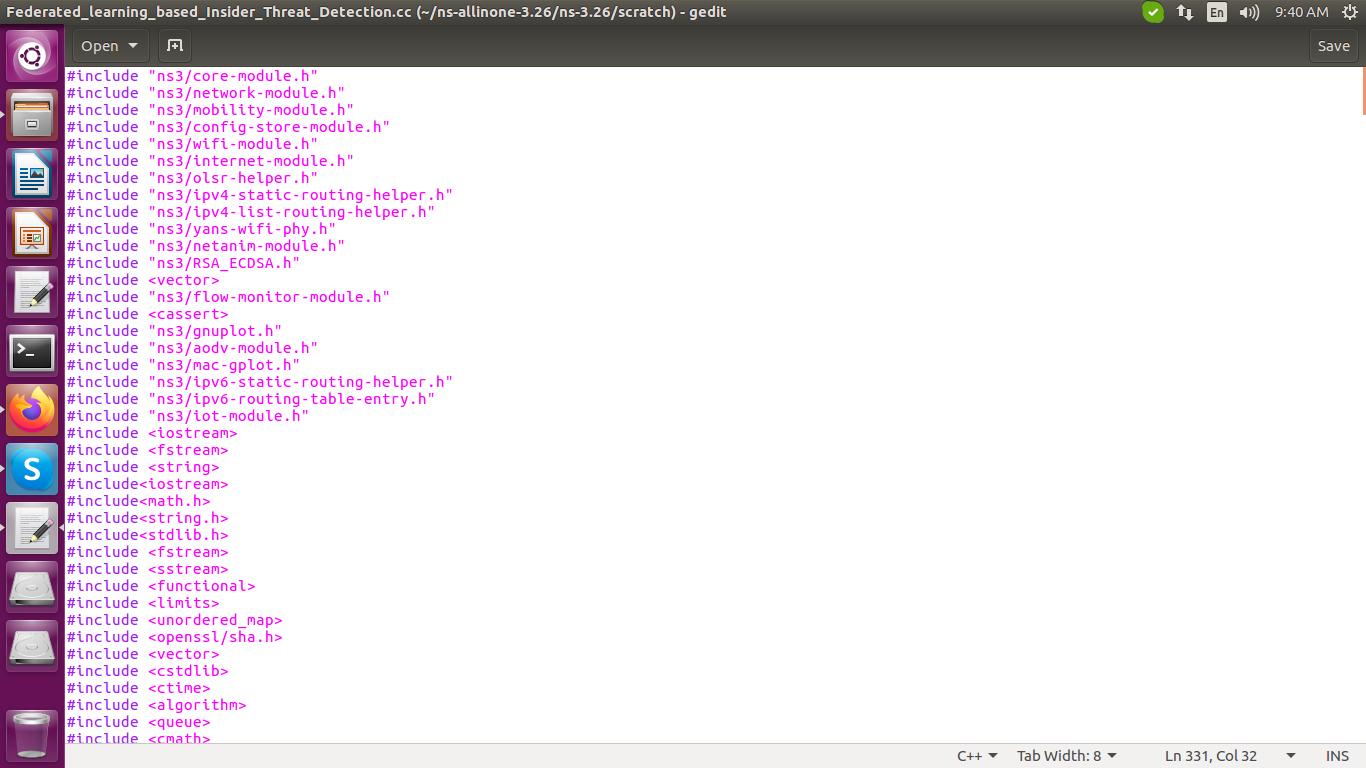
**Title:- Federated learning-based Insider Threat Detection**

In this Project, create a Network, it consists of 100- IoT devices, 10- Users 1- Trust Authority 1- Blockchain, 2- Edge Devices and 1- Cloud Server. Initially, we perform the Node Authentication process, In this process the users are registered by their credentials such as user name, user password, mail ID, one-time password, digital certificate, and then security questions and answers. Based on the password, the Trust Authority (TA) generates a random digital certificate using the Hybrid Rivest-Shamir-Adleman with an Elliptic Curve Digital Signature Algorithm (RSA-ECDSA). The data are stored in the blockchain using hashing (Stellar Consensus Protocol). Next, Clustering process, In this process nodes are clustering using the Ordering Points to Identify the Clustering Structure (OPTICS) technique. Next, Local model generation and Threat detection process, In this process the clustering heads are sent to the local model for data privacy. Next, perform the Global model generation process, In this process the Local model data are converted to the Global model using session token (Secure Hash Algorithm). The proposed approach is validated through several performance metrics such as, Accuracy vs number of IoT devices, Accuracy vs Communication rounds, Time period vs trust value, Throughput vs running time, Node number vs batch size, Performance vs tests.

**Steps:**

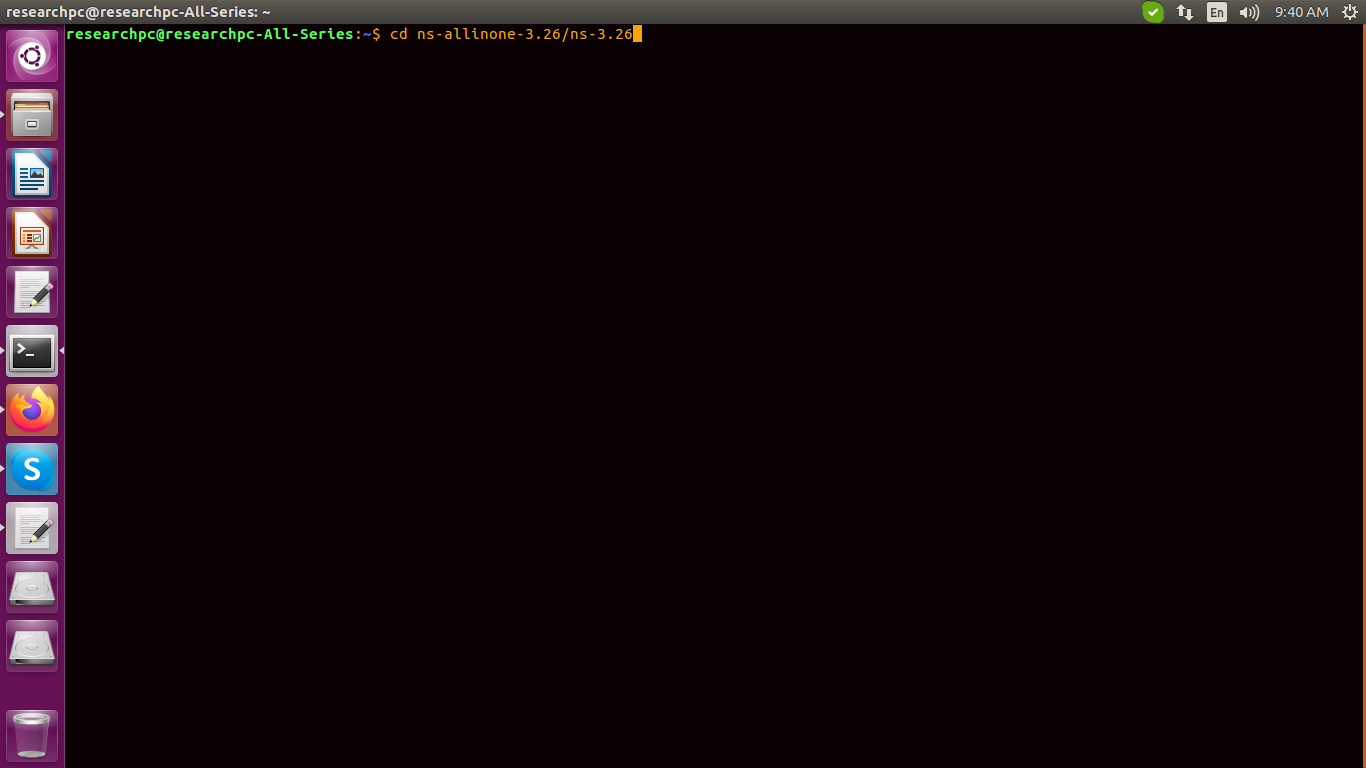
Proposed Main File



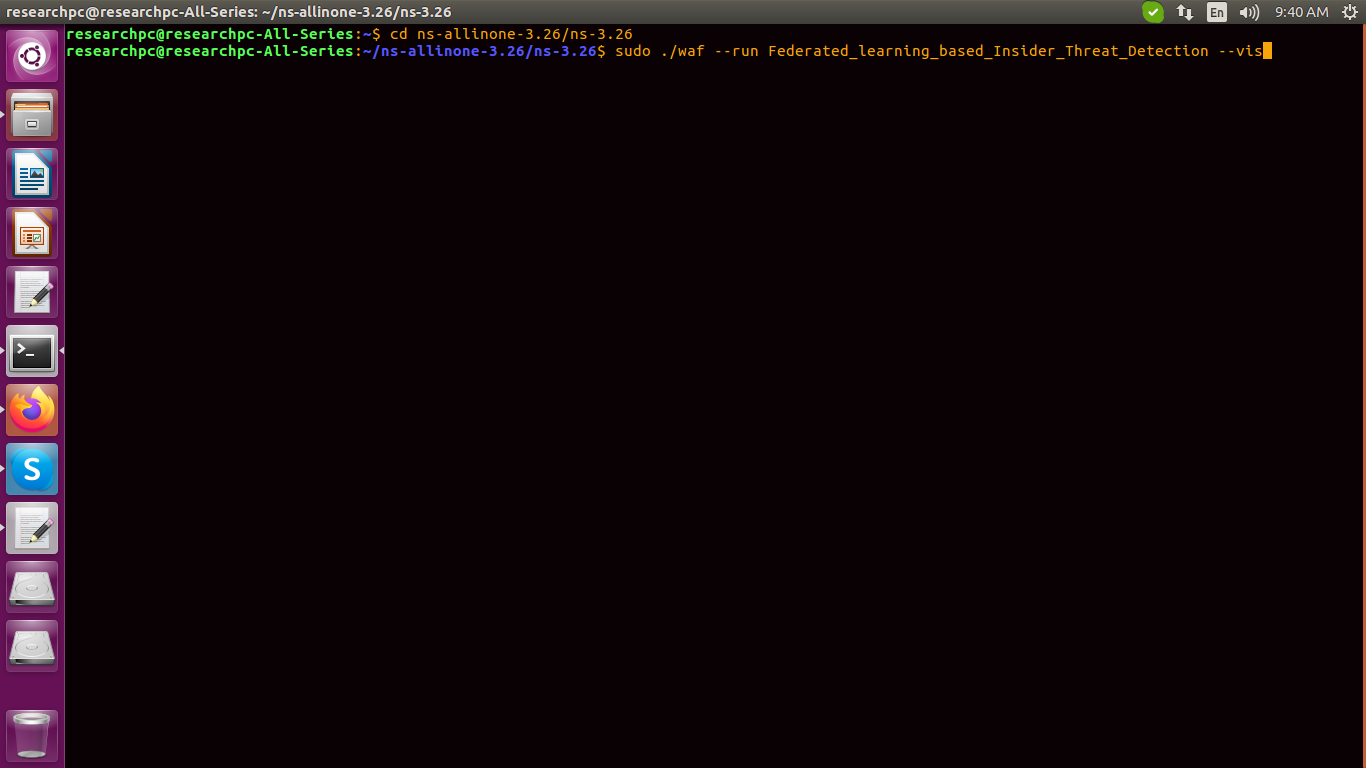
Open the terminal [ctrl+alt+t].



Change the project location by using the cd command.

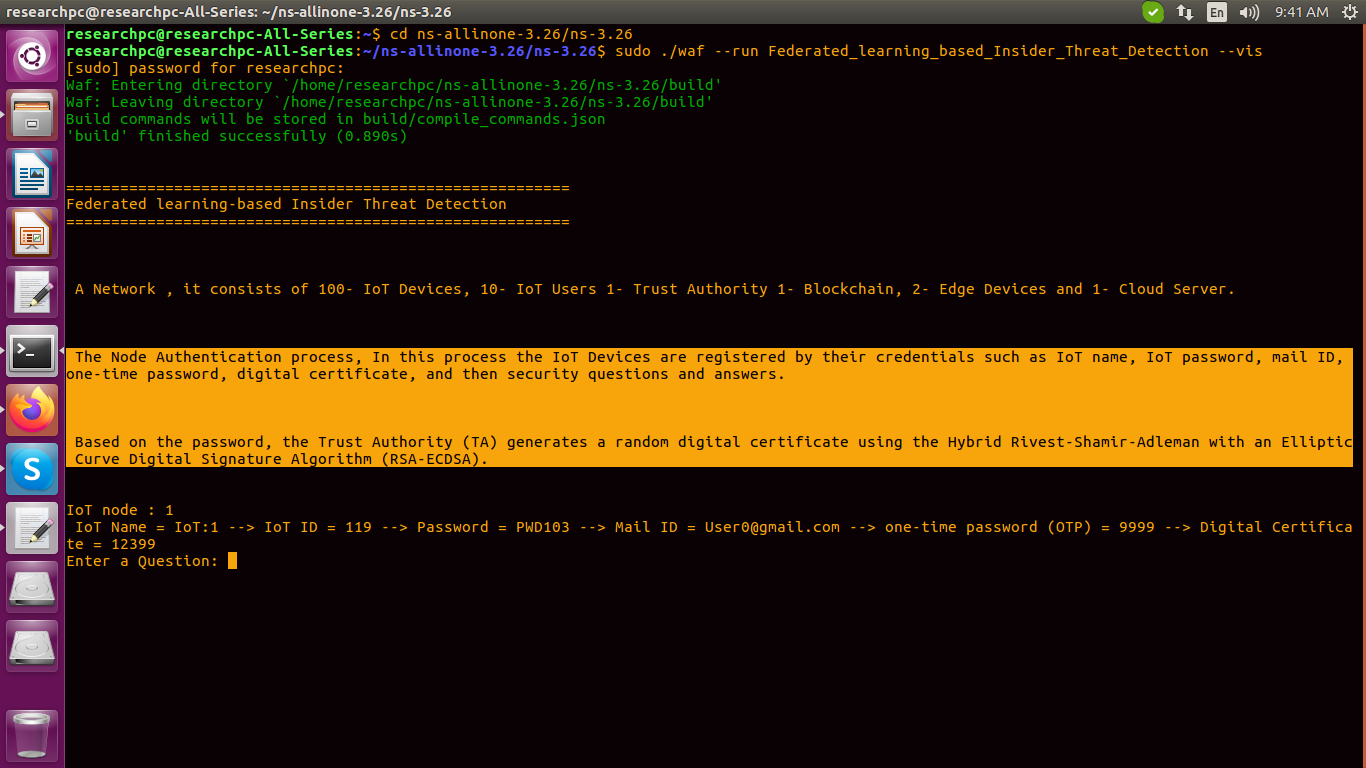


Execute the Proposed main file

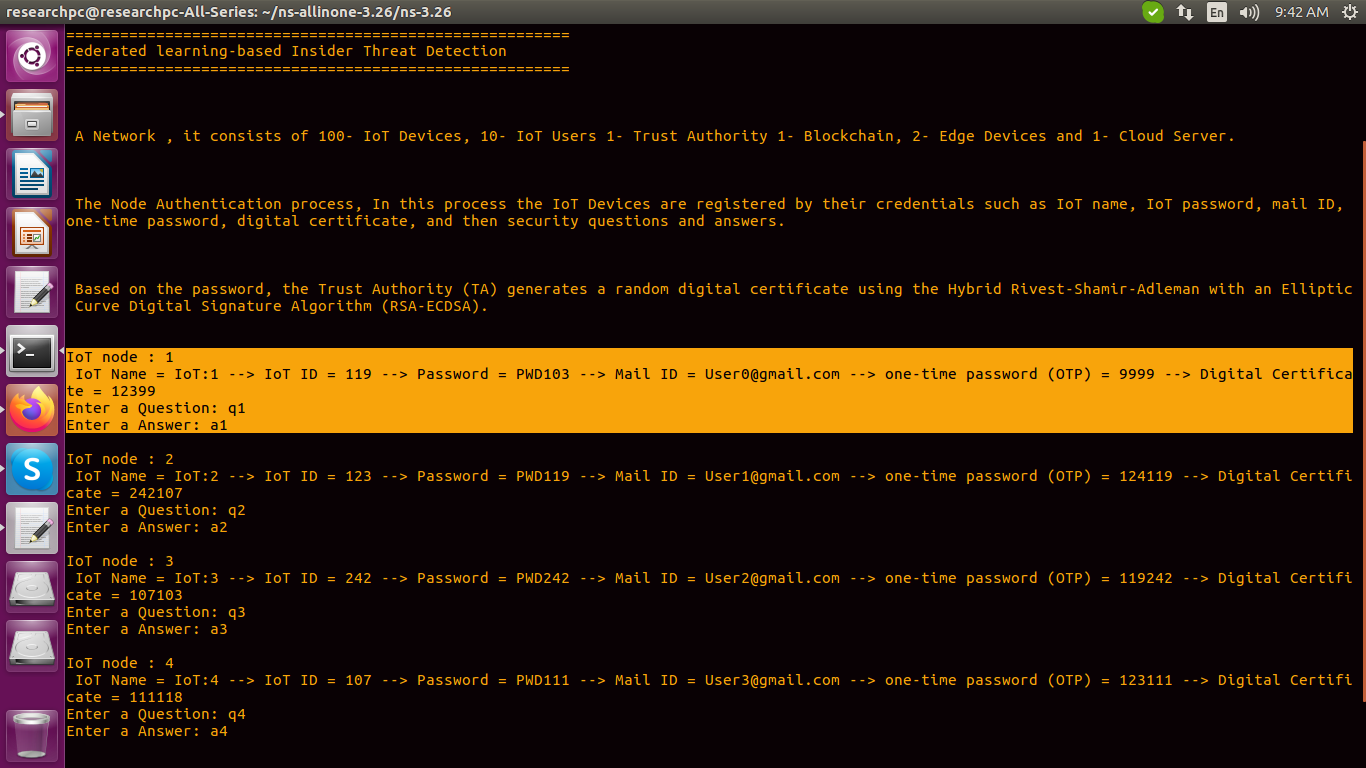


Initially, perform the Node Authentication process, In this process the users are registered by their credentials such as user name, user password, mail ID, one-time password and then security questions and answers.

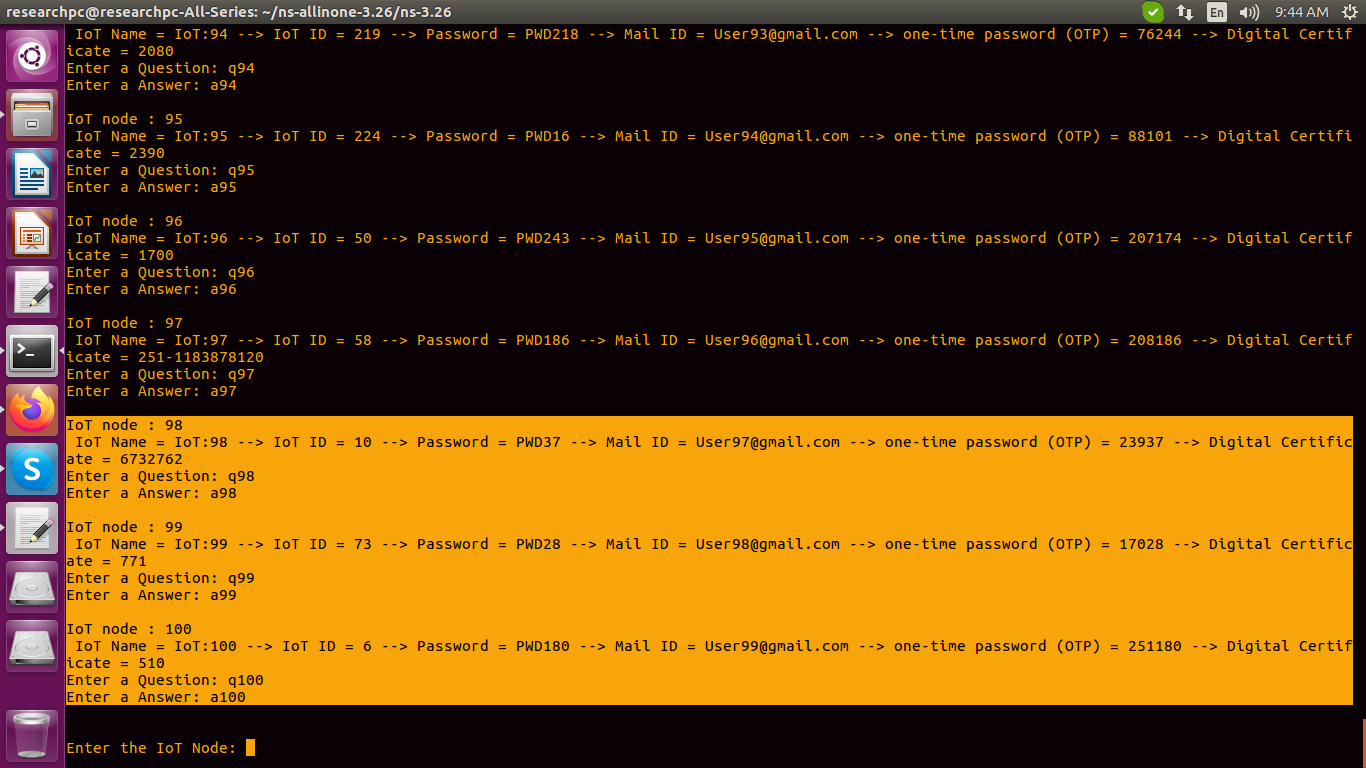
Based on the password, the Trust Authority (TA) generates a random digital certificate using the Hybrid Rivest-Shamir-Adleman with an Elliptic Curve Digital Signature Algorithm (RSA-ECDSA).



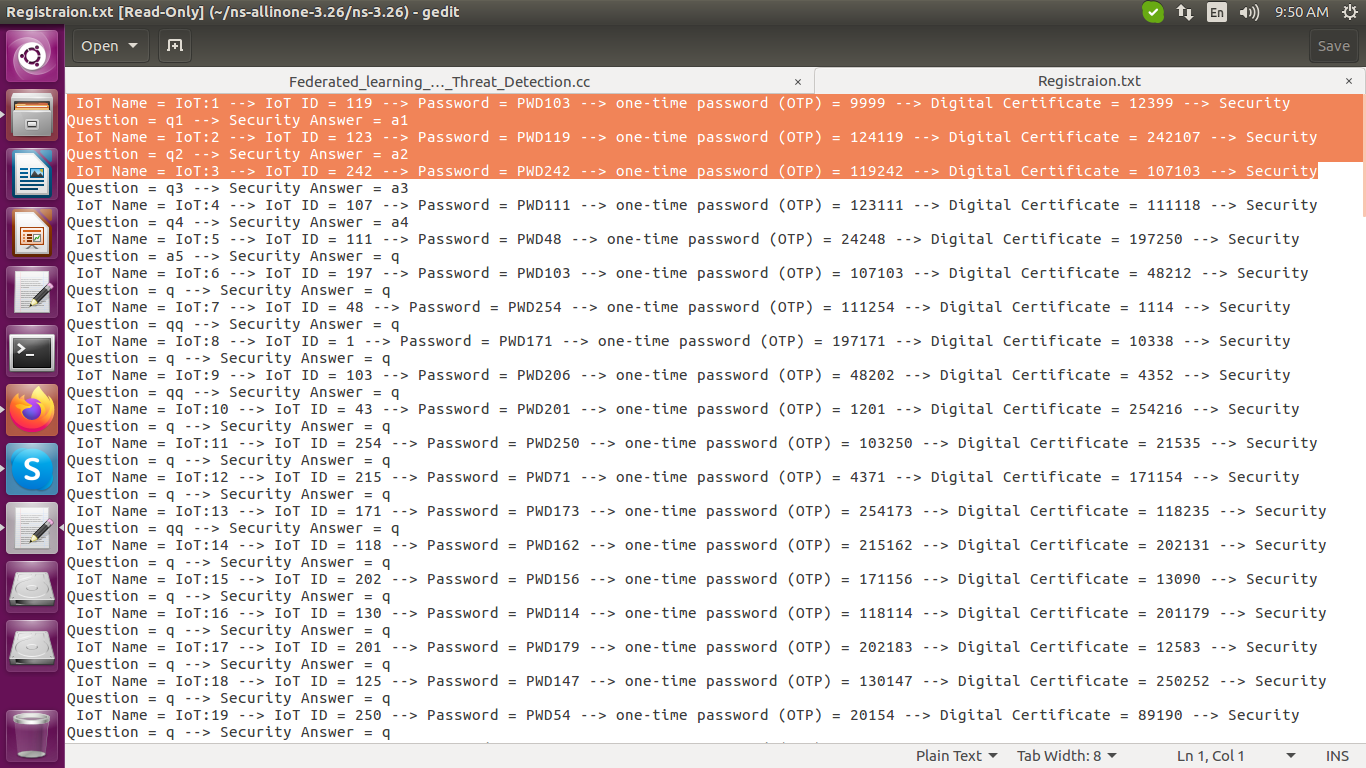
Enter the Question and Answer:



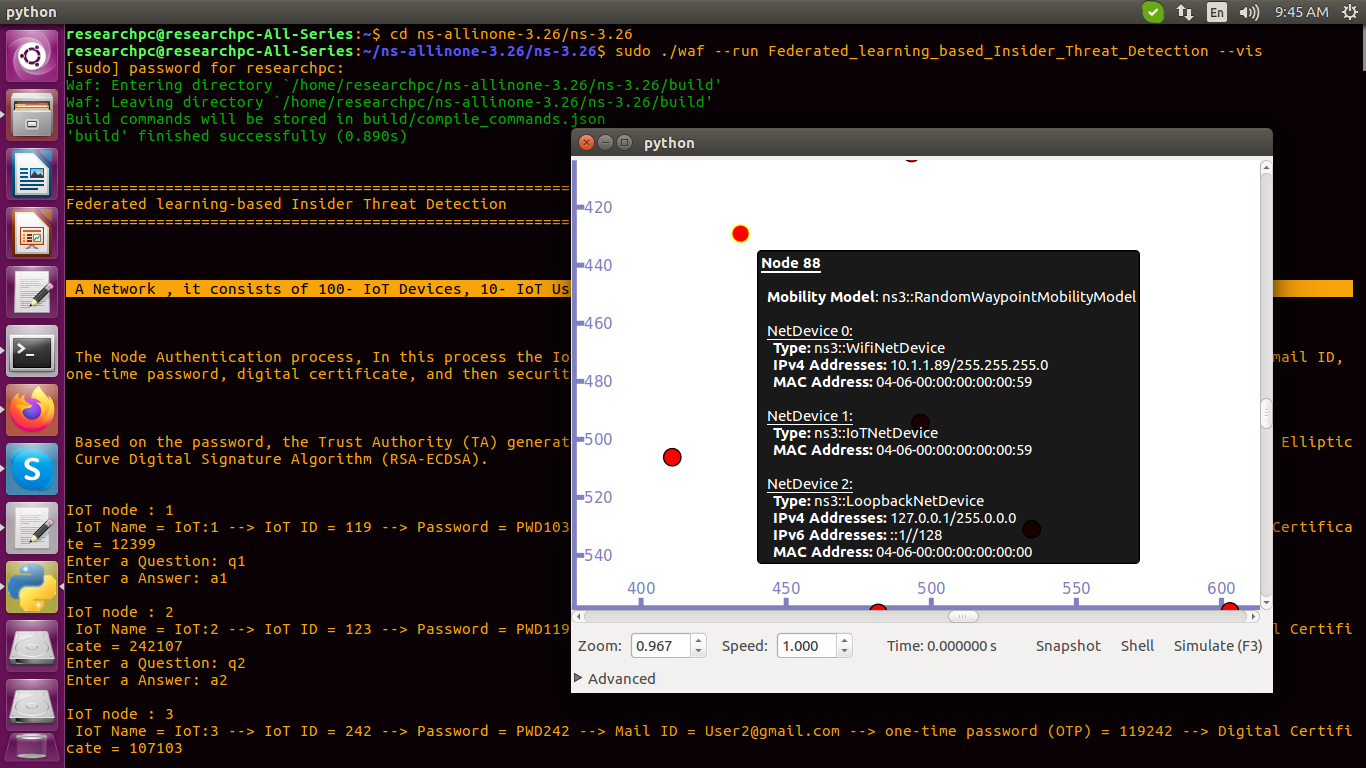
Enter the Login details:



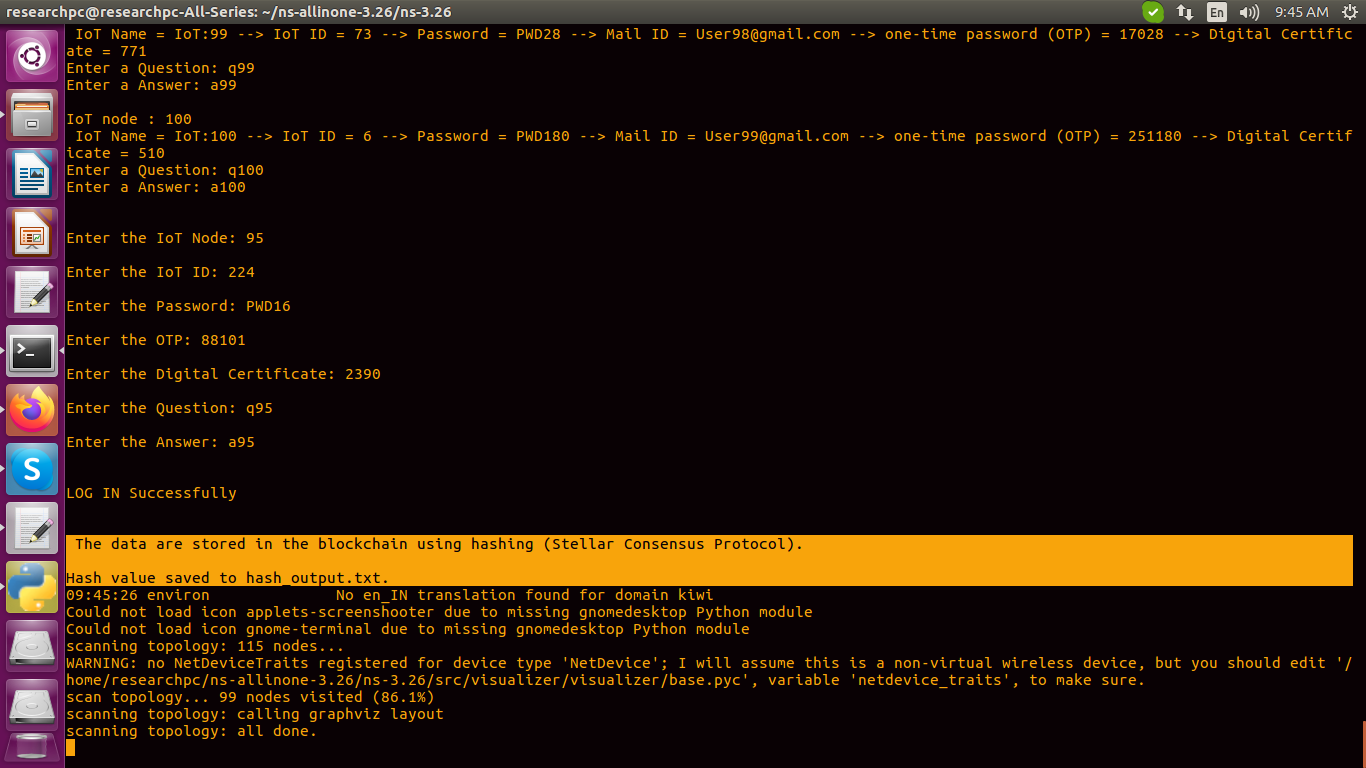
Here stored the Registration .txt file



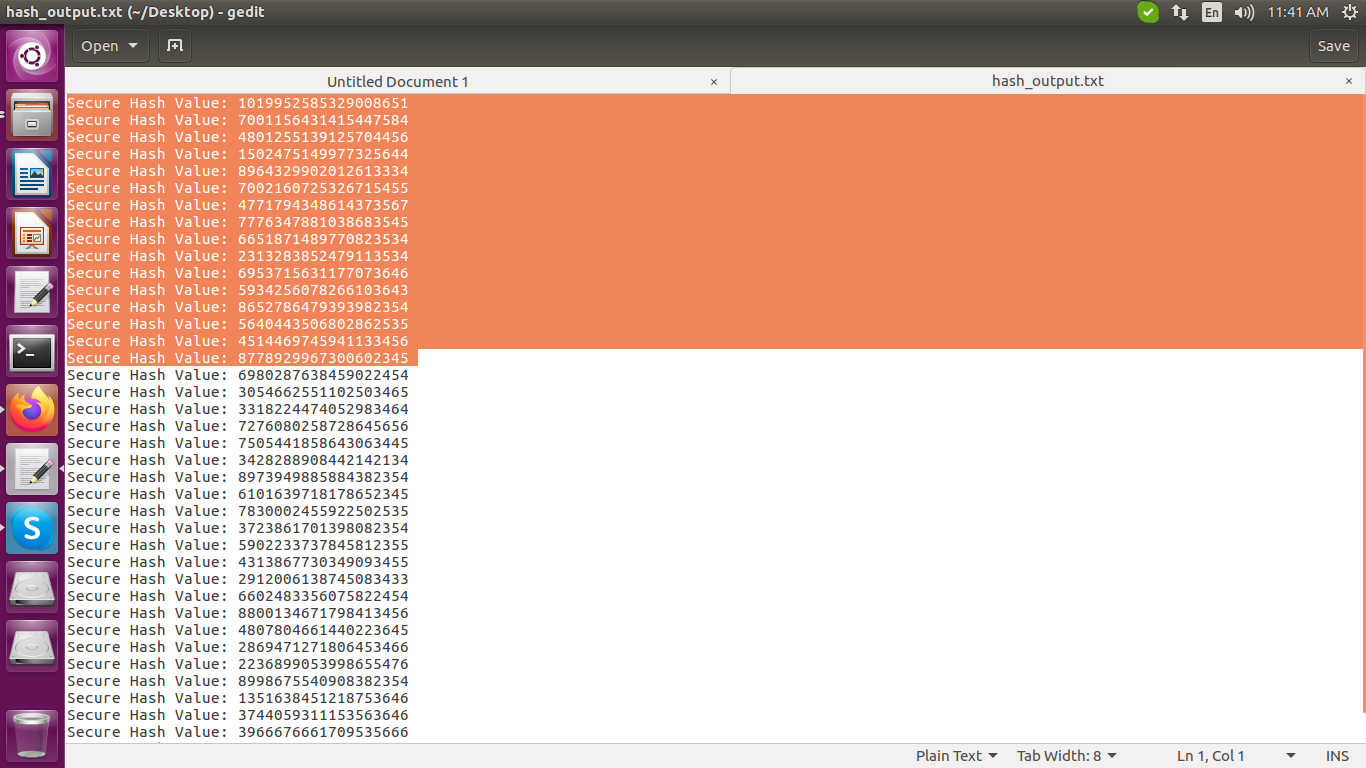
Create a Network , it consists of 100- IoT devices, 10- Users 1- Trust Authority 1- Blockchain, 2- Edge Devices and 1- Cloud Server.



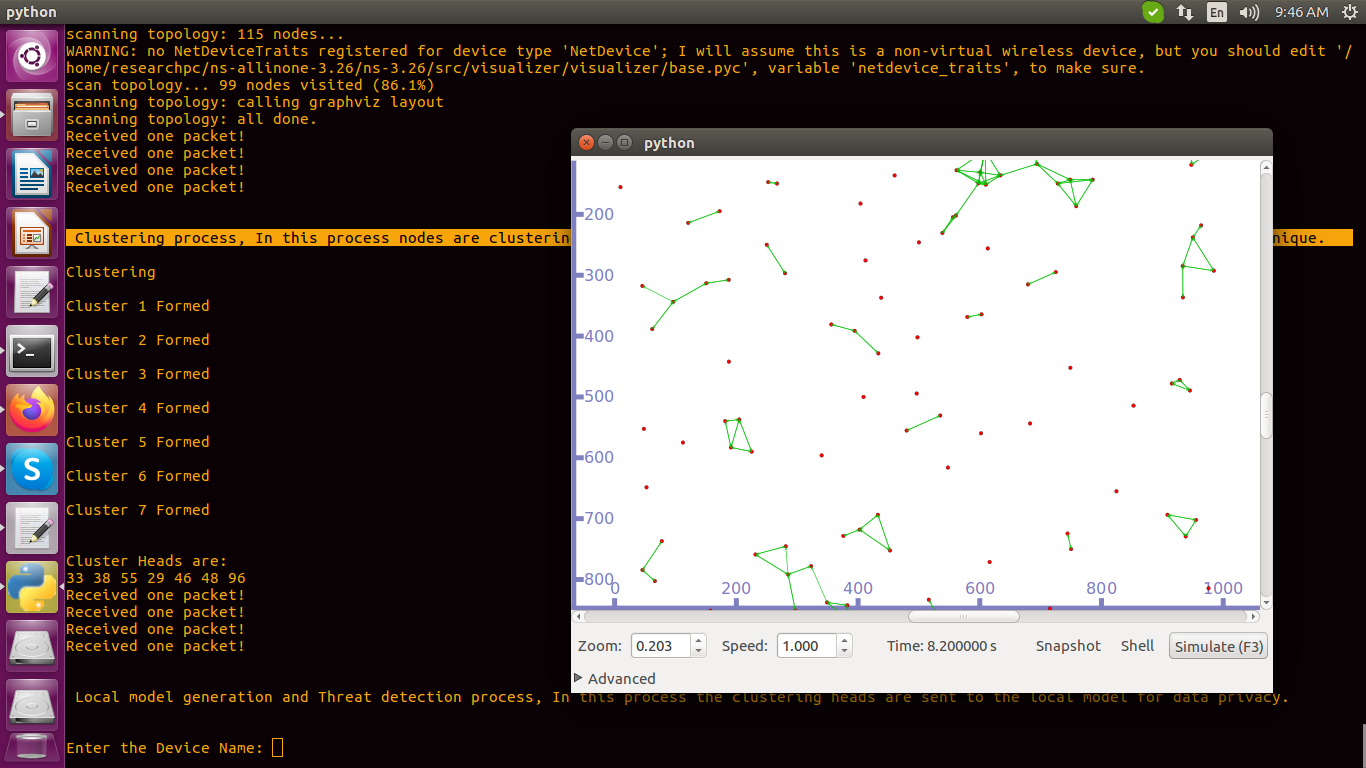
The data are stored in the blockchain using hashing (Stellar Consensus  
Protocol).



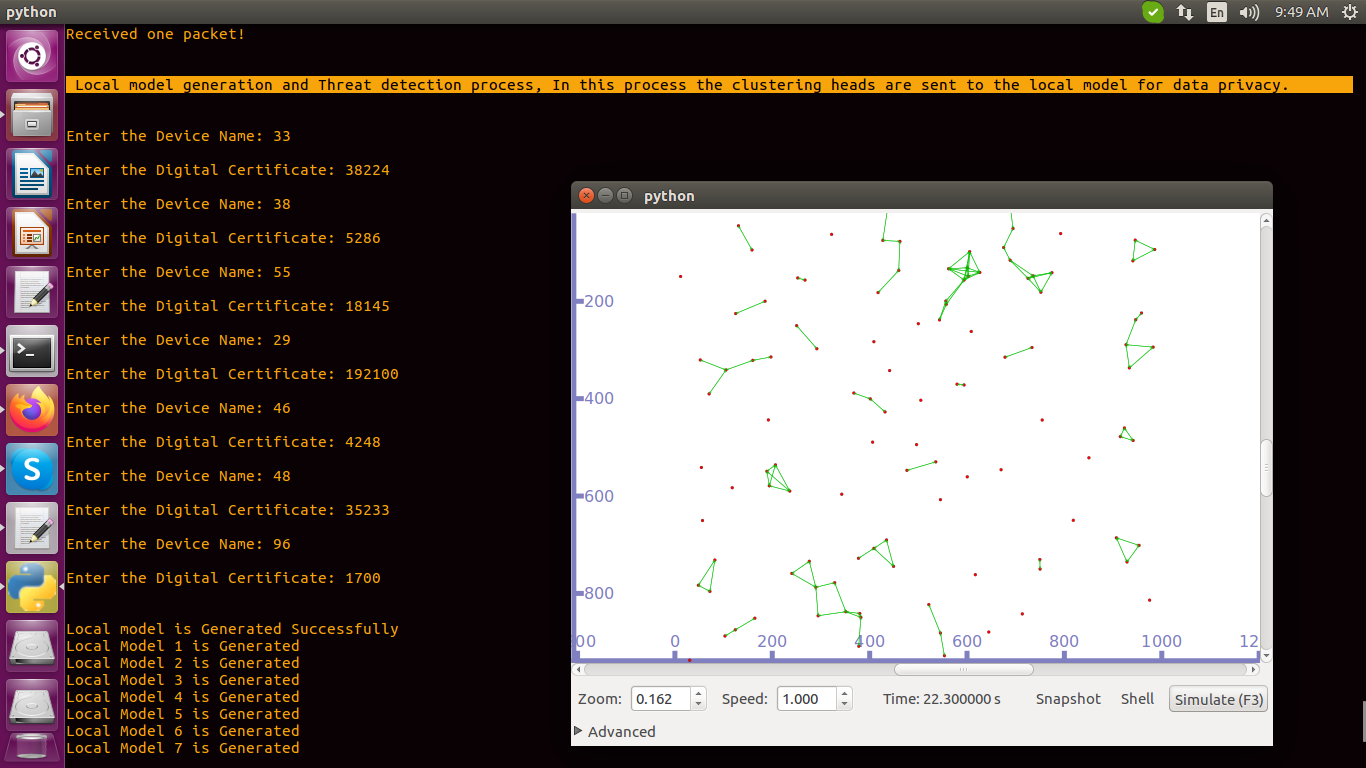
Hash value saved to hash\_output.txt file



Next, Clustering process, In this process nodes are clustering using the Ordering Points to Identify the Clustering Structure (OPTICS with Centroid refinement) technique.

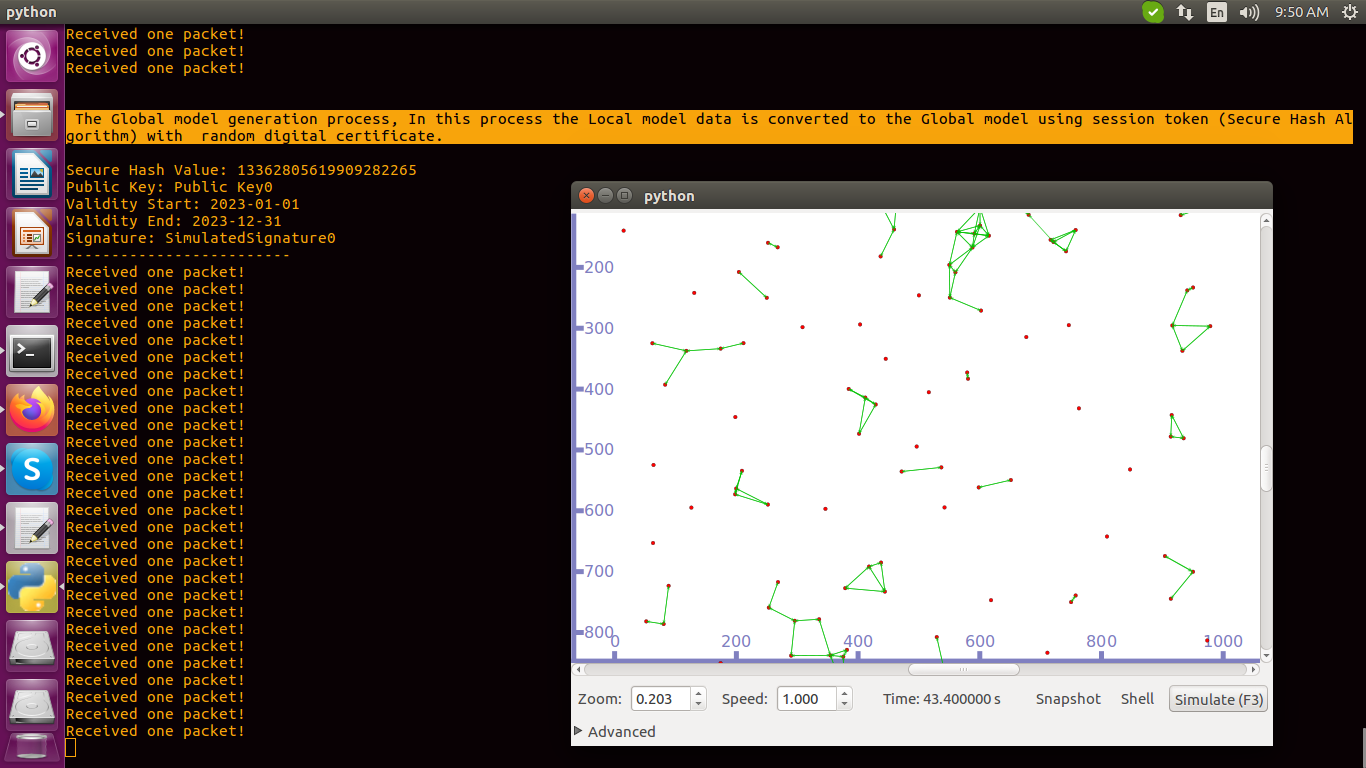


Next, Local model generation and Threat detection process, In this process the clustering heads are sent to the local model for data privacy.

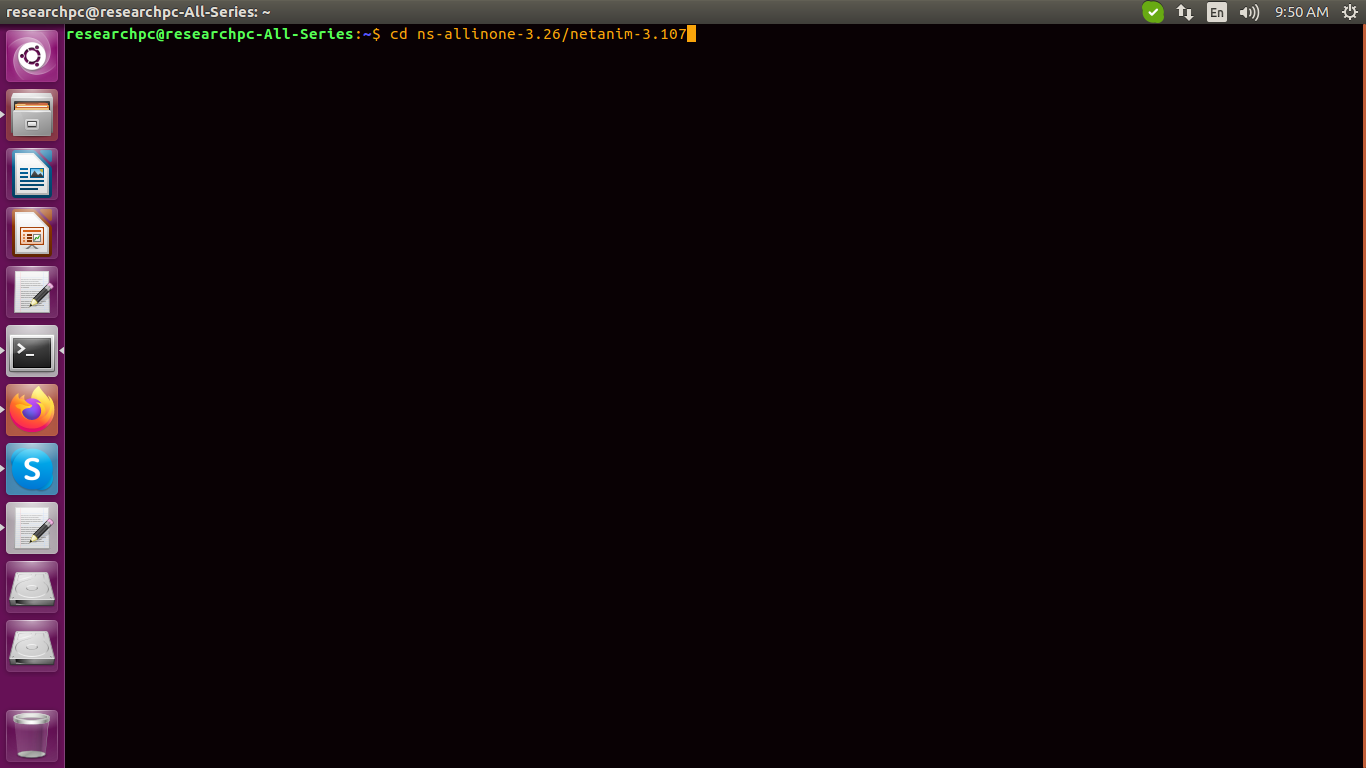




Next, we perform the Global model generation process, In this process the Local model data are converted to the Global model using session token (Secure Hash Algorithm).



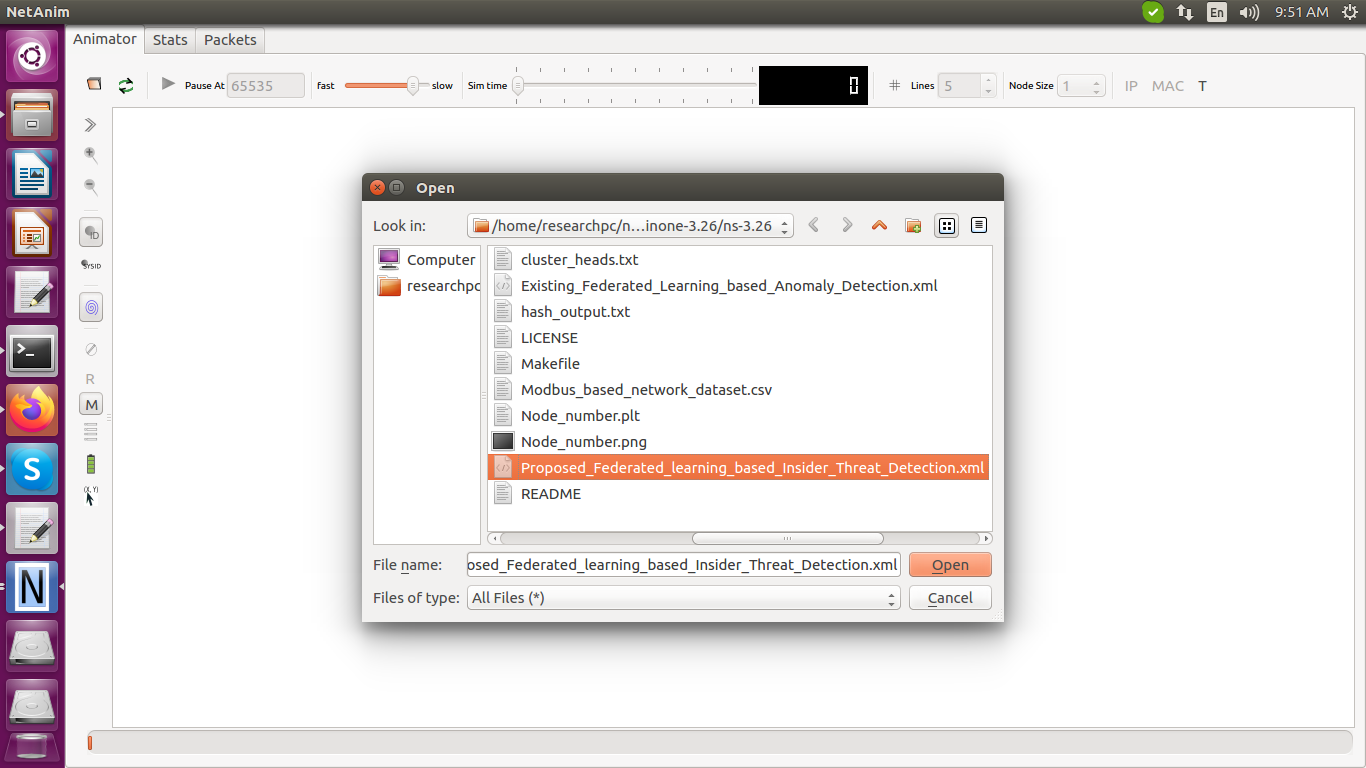
Change the location for get the netanimator result



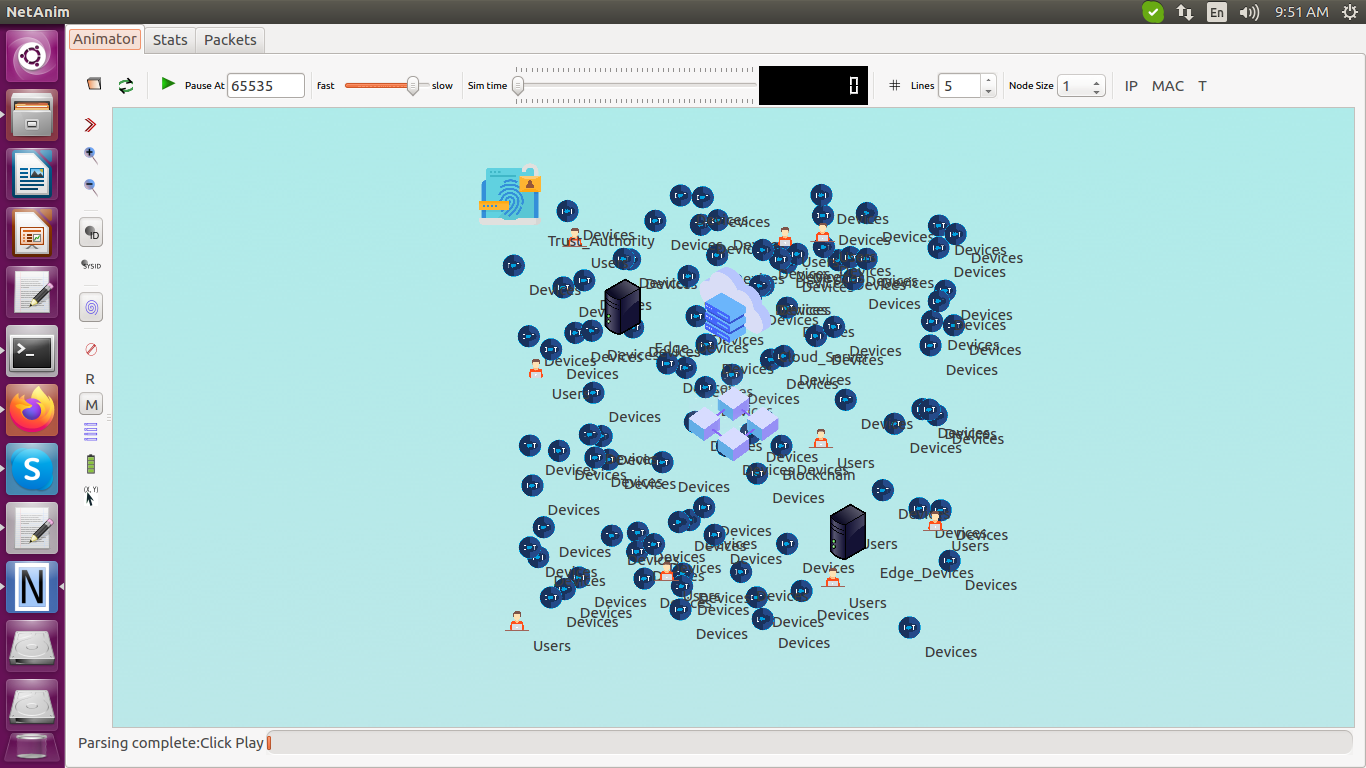
execute the command

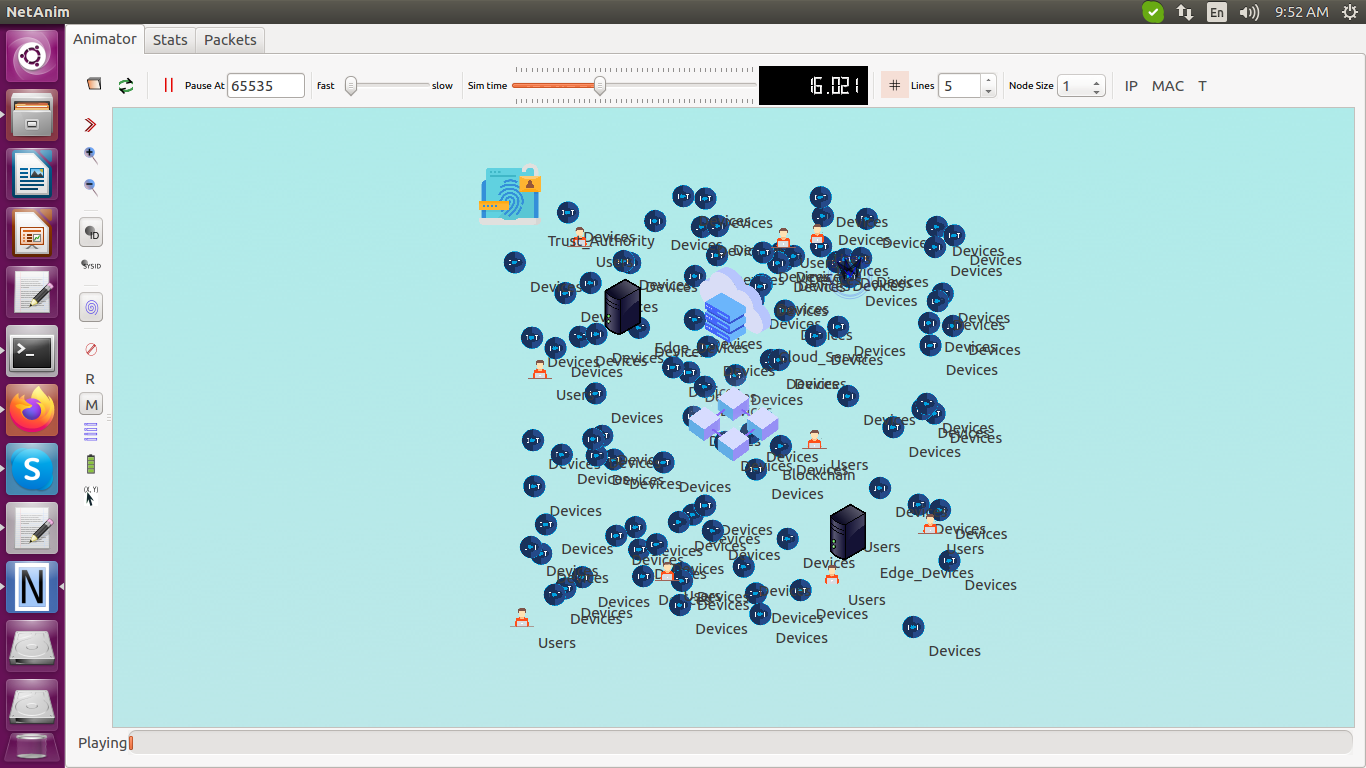


select the xml file

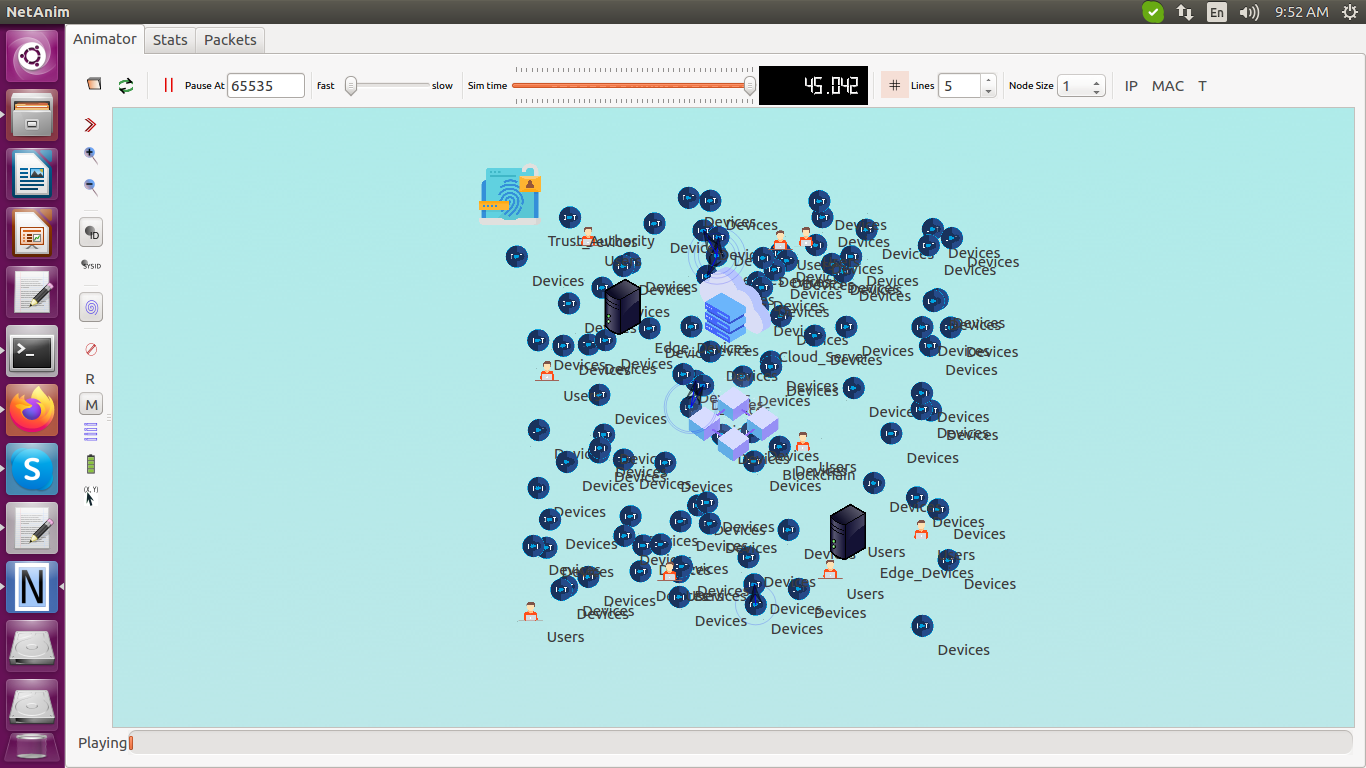


get the simulation



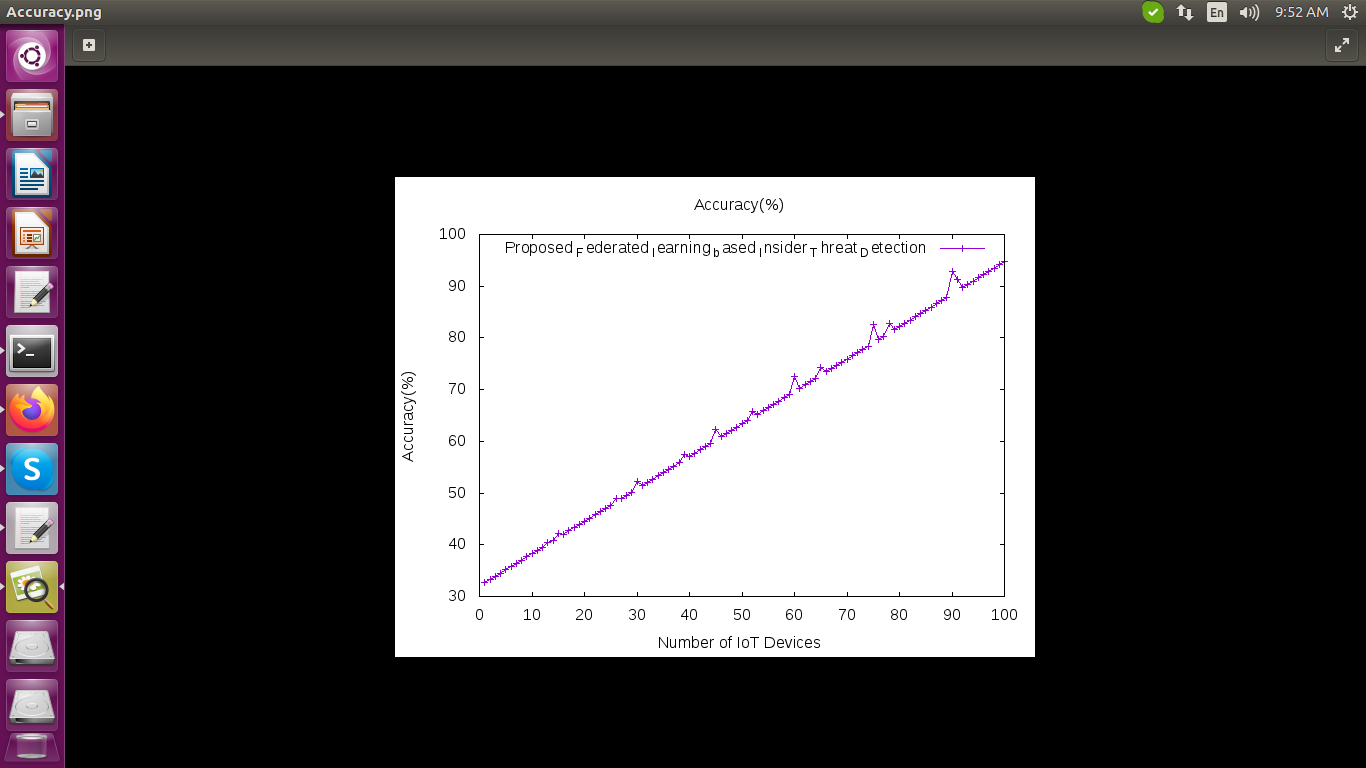




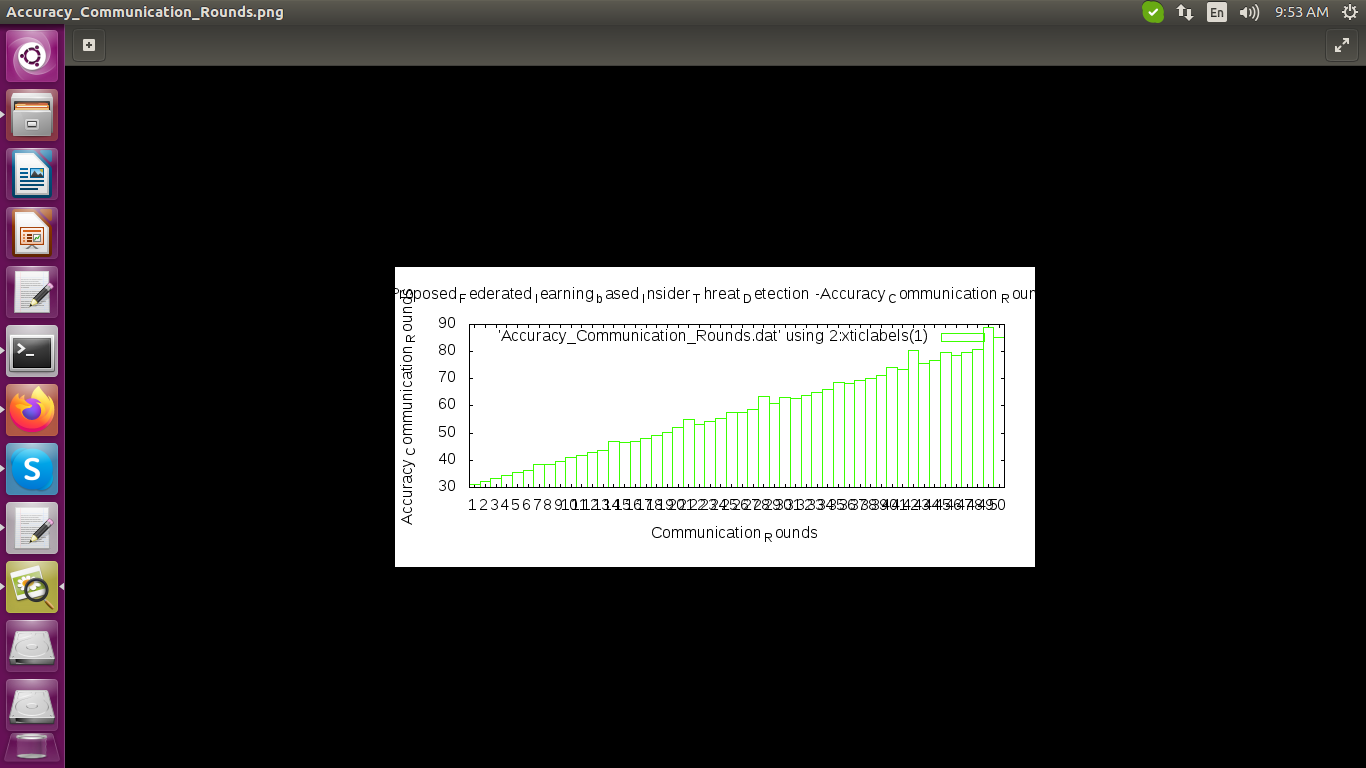


The proposed approach is validated through several performance metrics such as,

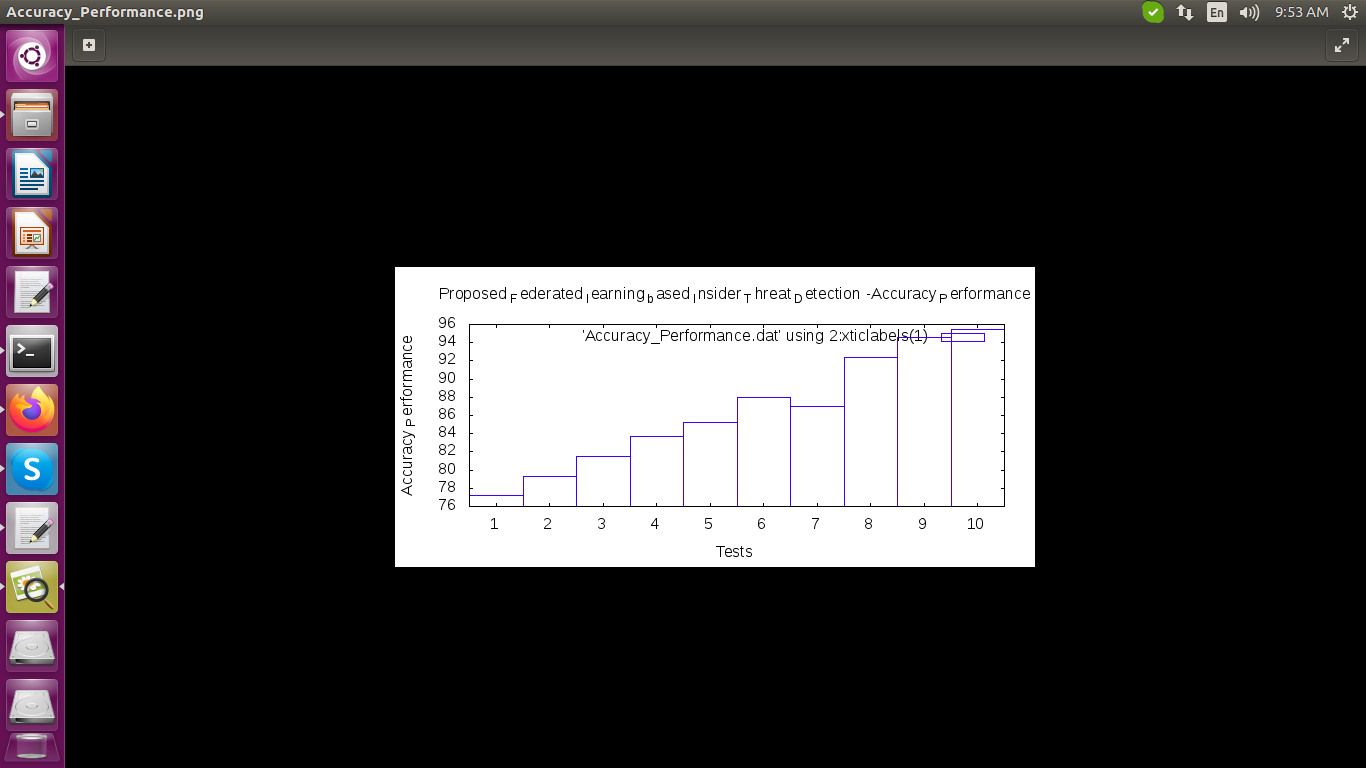
The results graph for Accuracy(%) vs. Number of IoT devices



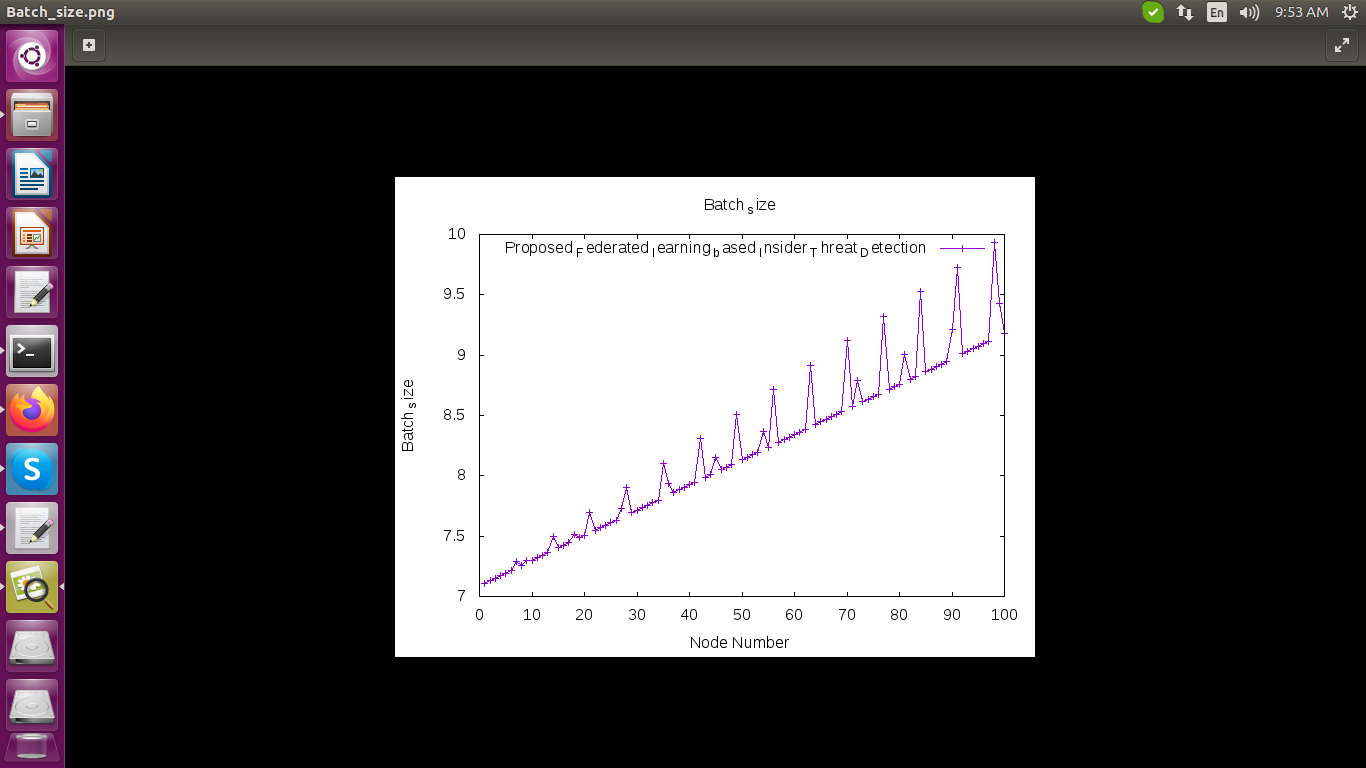
The results graph for Accuracy vs. Communication Rounds



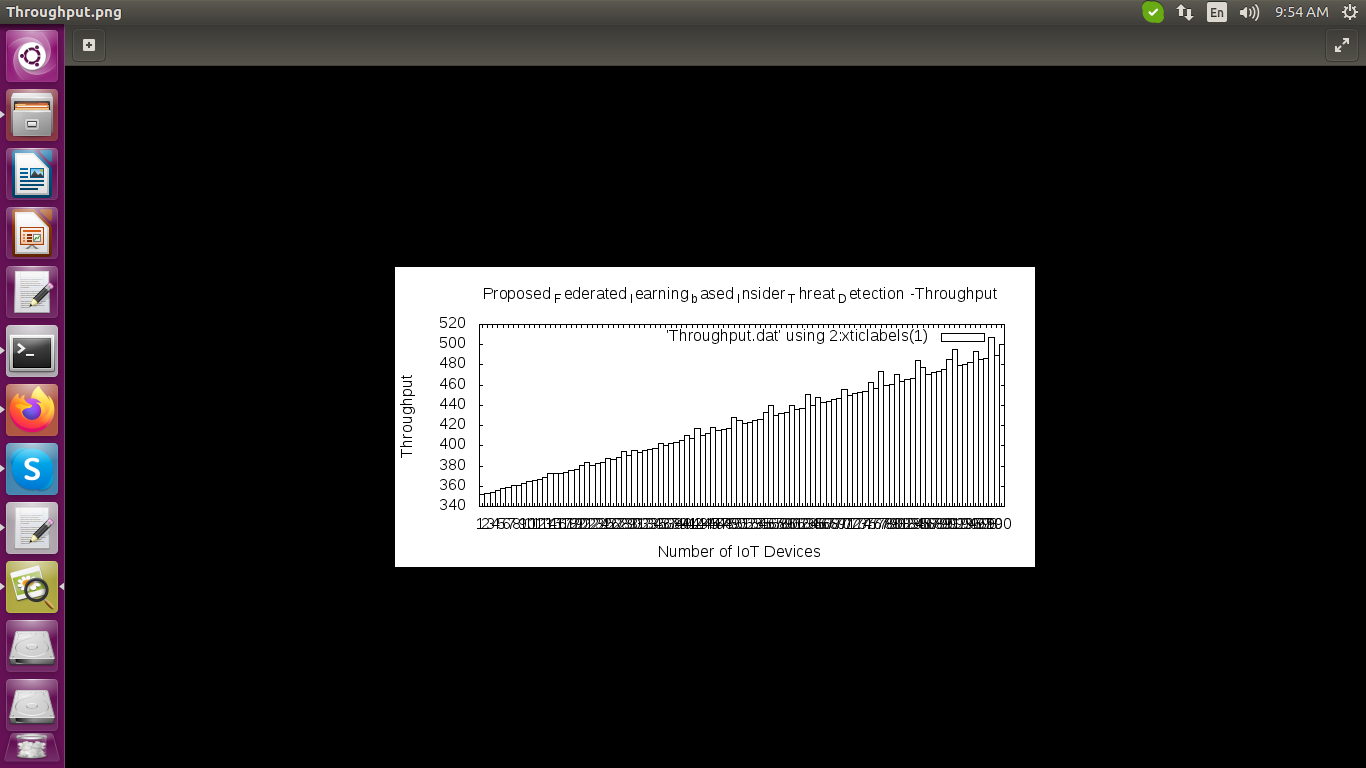
The results graph for Accuracy vs. Tests



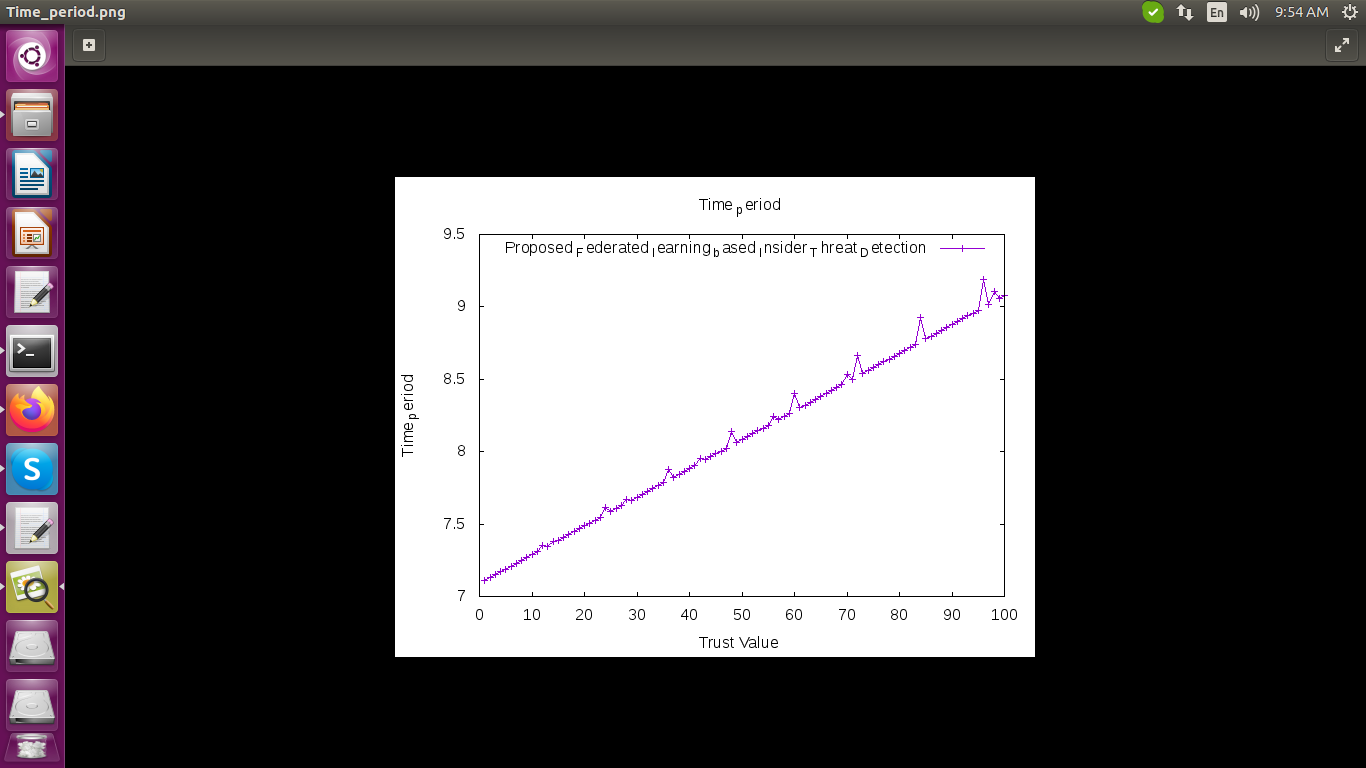
The results graph for Node number vs. Batch size



The results graph for Throughput vs. Number of IoT Nodes



The results graph for Time Period vs. Trust Value

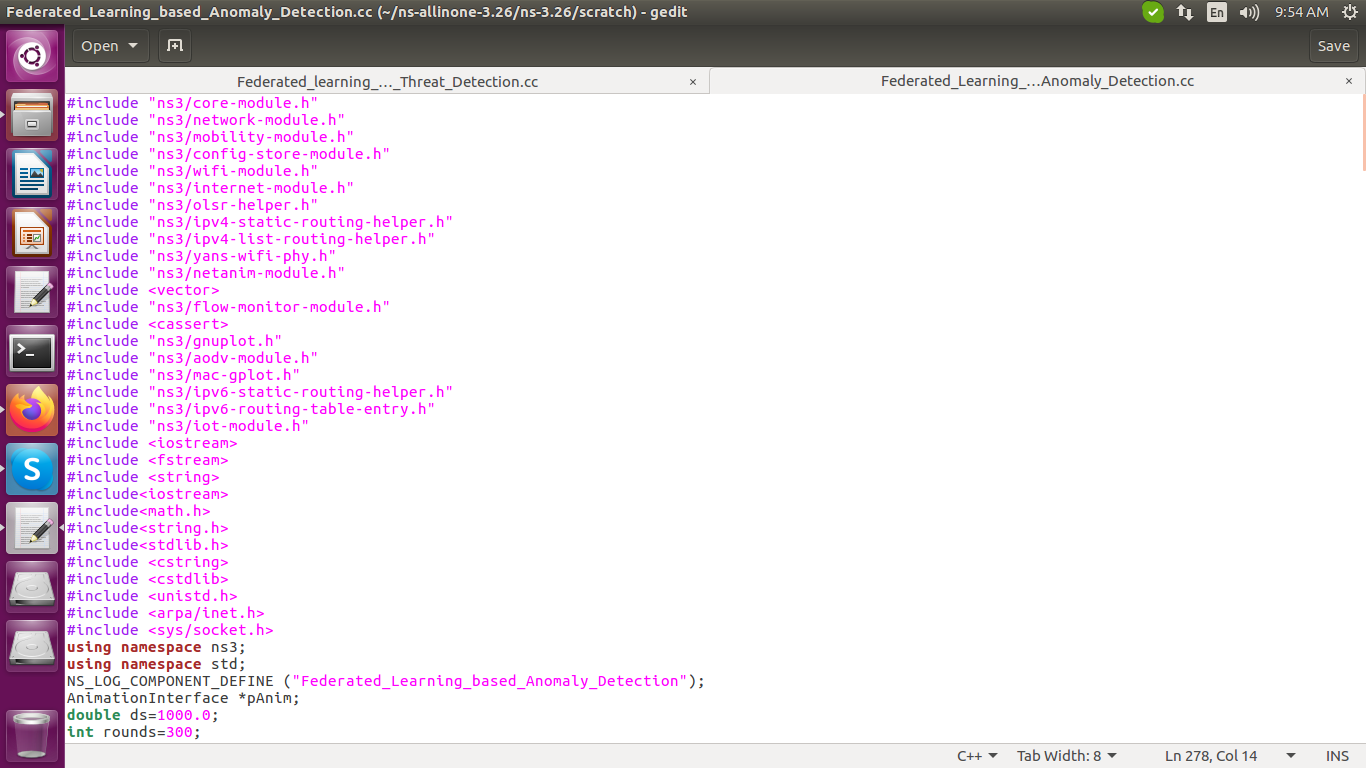


**Implement the Existing process based on the Reference 2 Title:- Federated Learning-based Anomaly Detection for IoT Security Attacks**

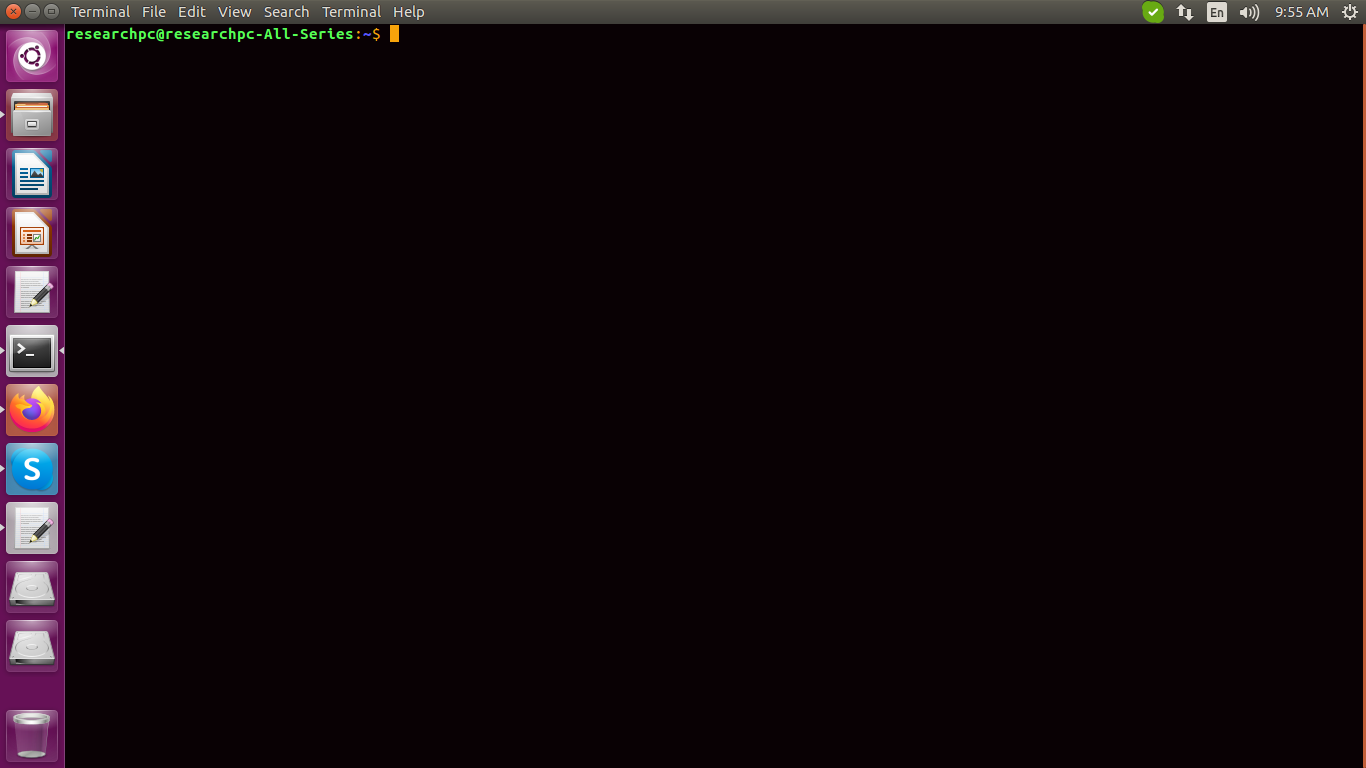
Initially, create an An IoT based Network, it consists of 100 - IoT Devices and 1-Cloud Server. Next load the Modbus\_based\_network\_dataset.csv. Next perform the attacker sends a flood of messages to overwhelm the end-device and make it unavailable to serve genuine message packets. Then next perform the non-FL version of intrusion detection process, in this process Classified the IoT nodes into normal or attacked. The Existing approach is validated through several performance metrics such as, Accuracy vs number of IoT devices, Accuracy vs Communication rounds, Time period vs trust value, Throughput vs running time, Node number vs batch size, Performance vs tests.

**Steps:**

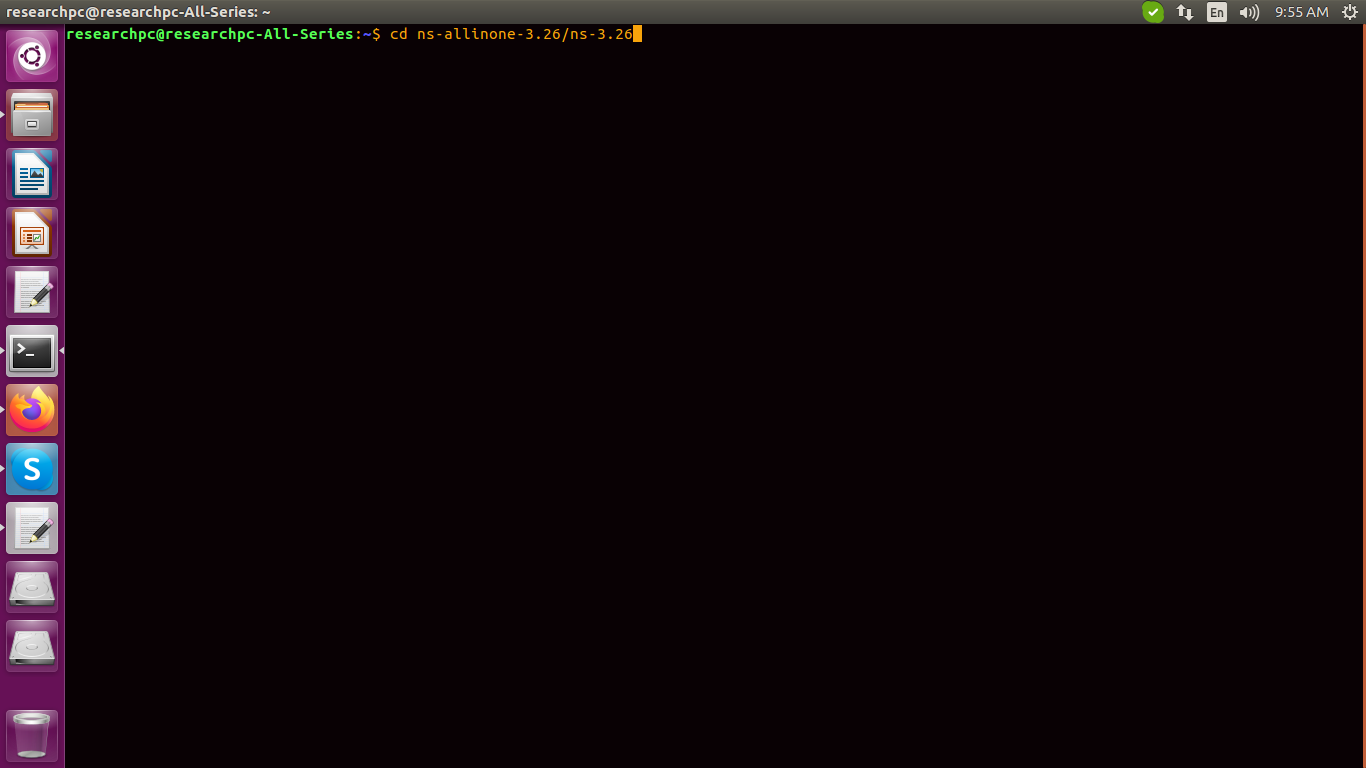
Existing Main File



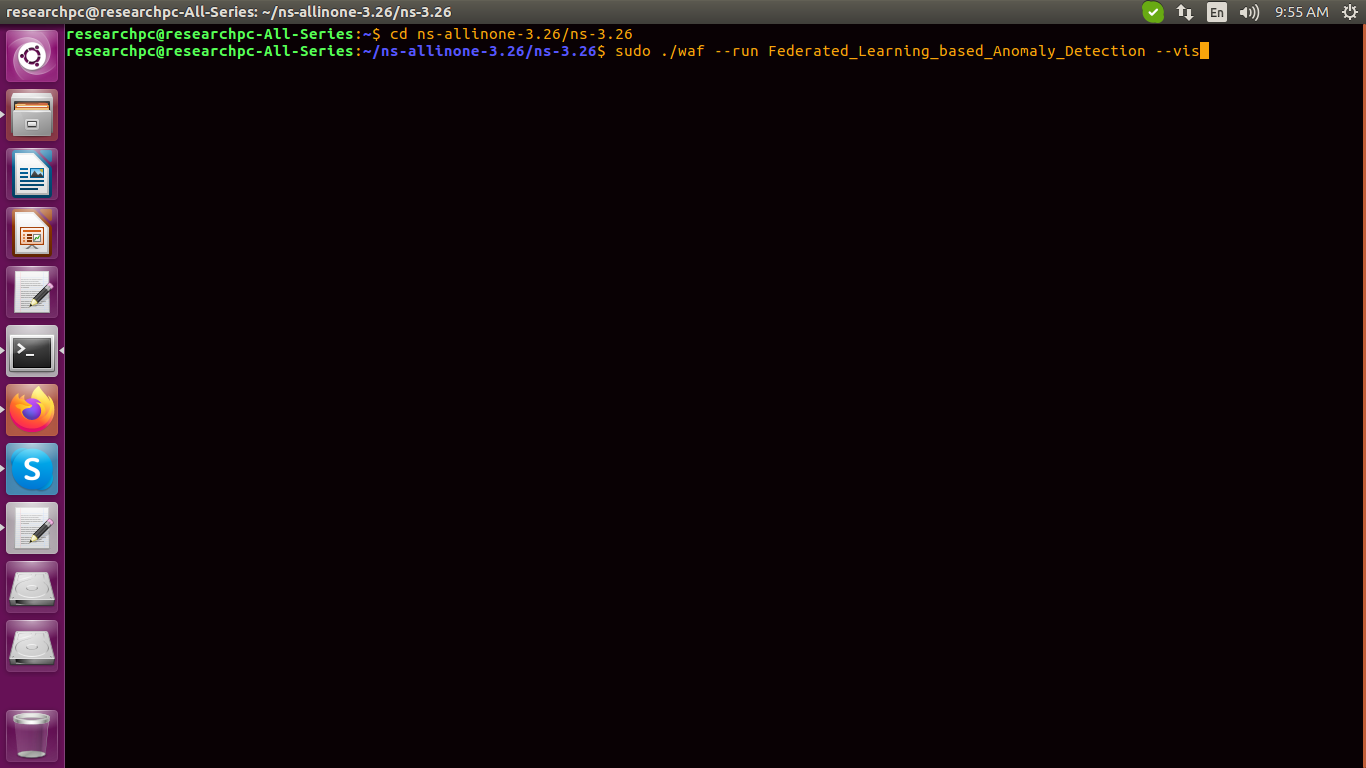
Open the terminal [ctrl+alt+t].



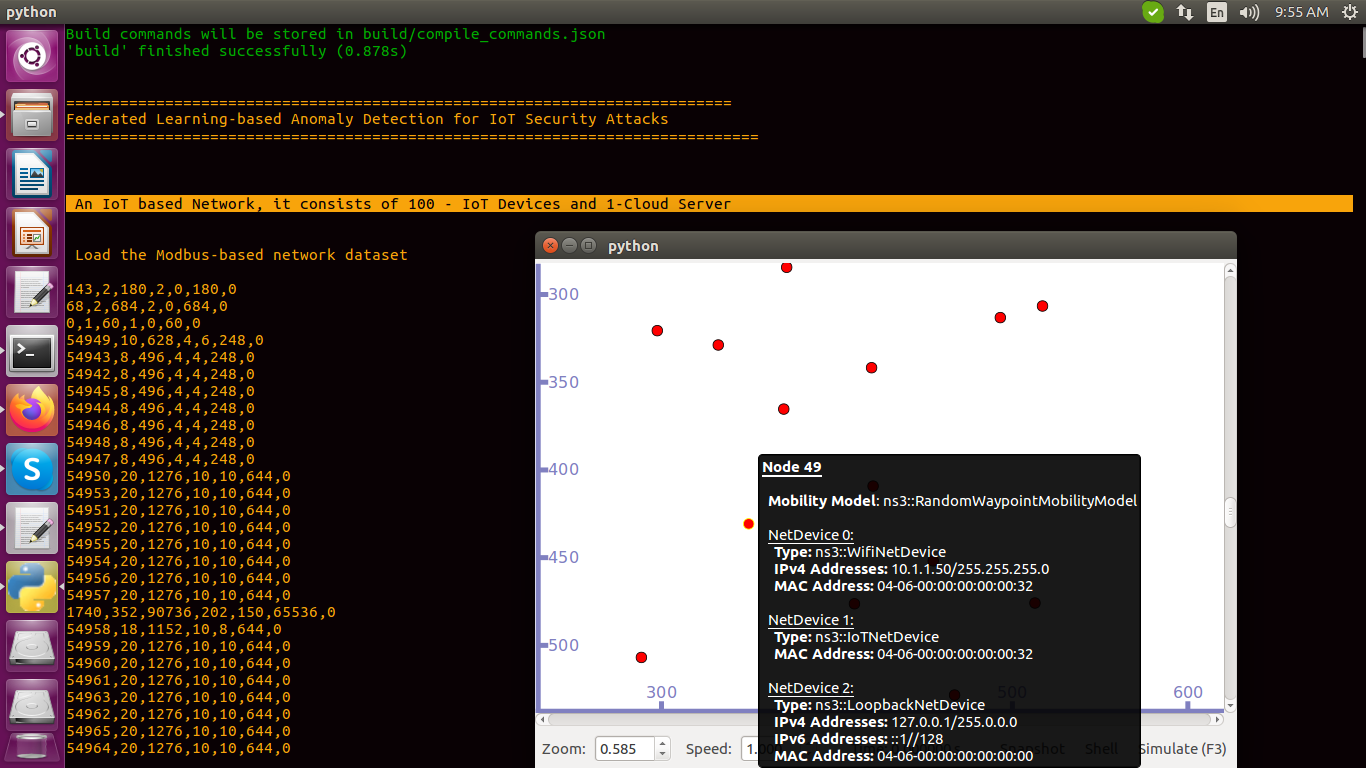
Change the project location by using the cd command.



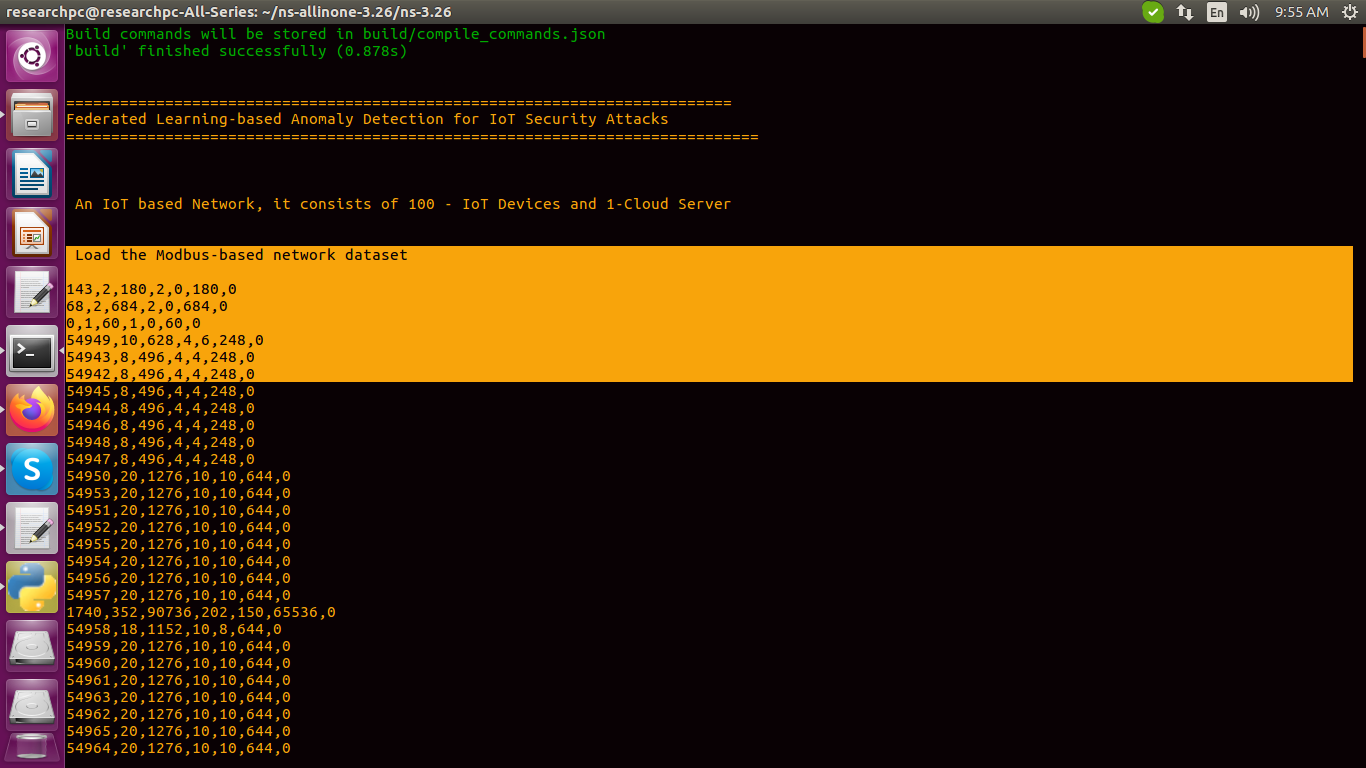
execute the Existing main file



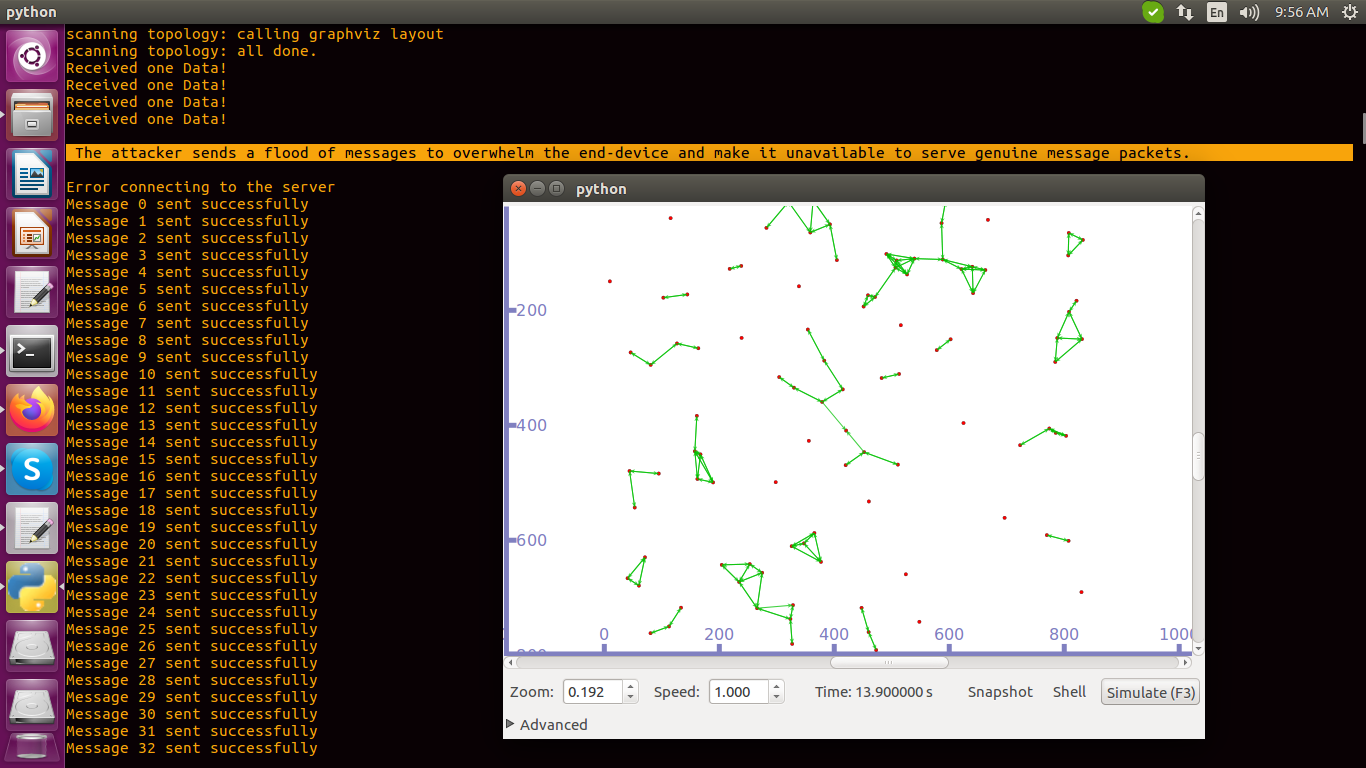
An IoT based Network, it consists of 100 - IoT Devices and 1-Cloud Server



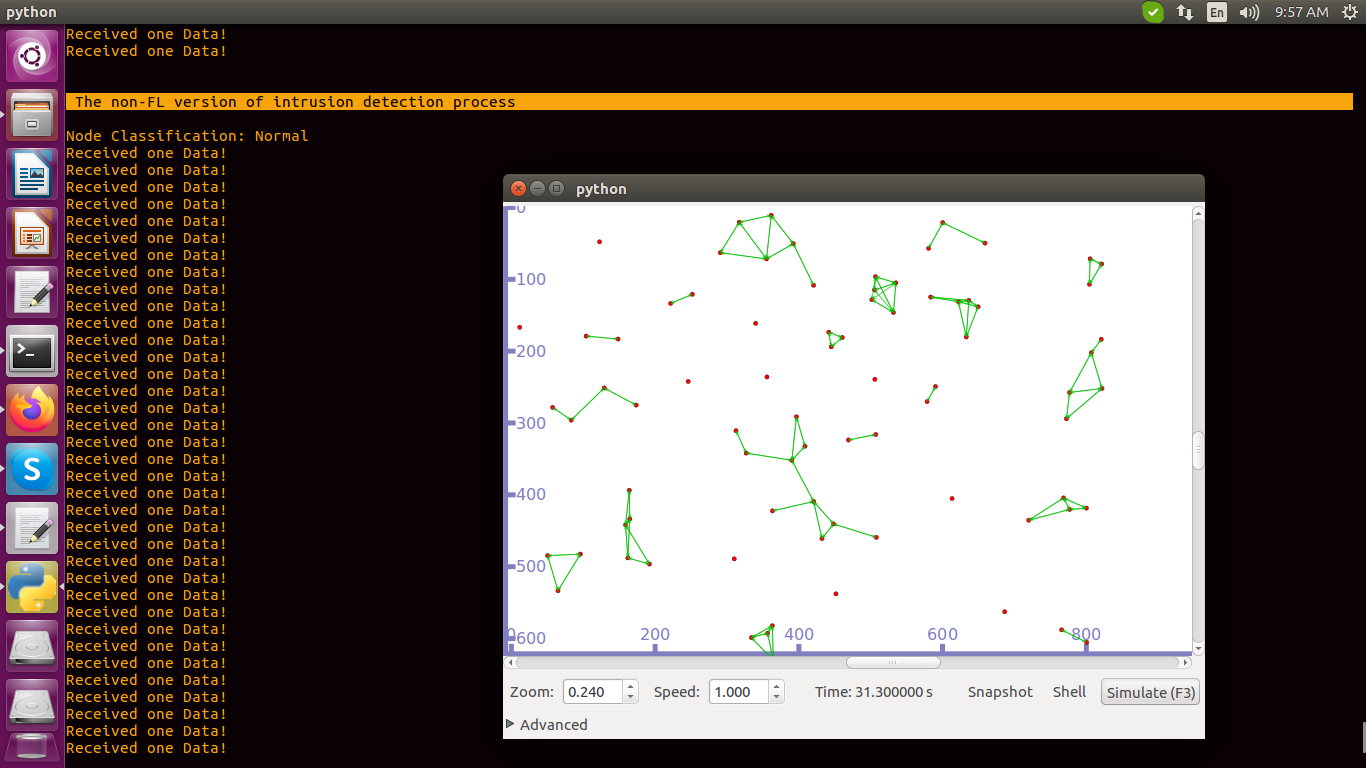
Next we load the Modbus\_based\_network\_dataset.csv.



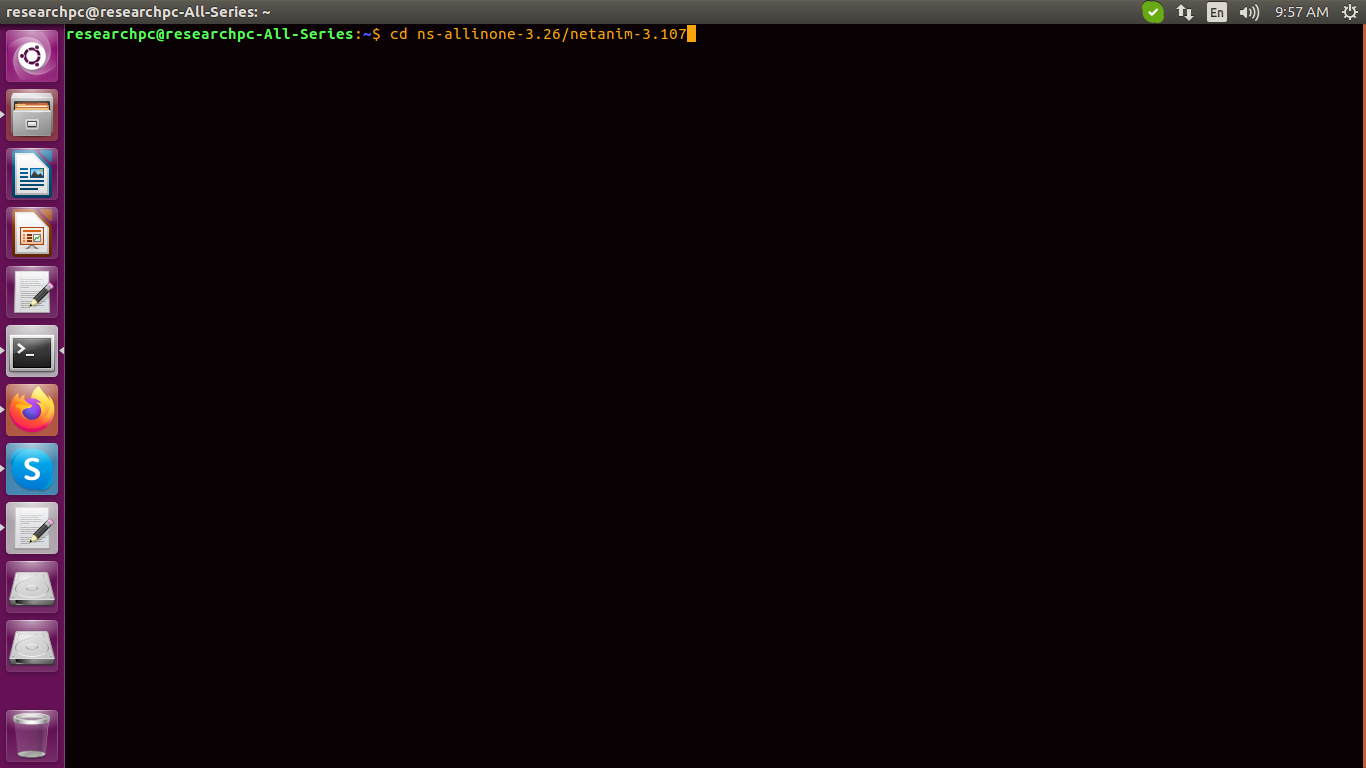
Next we perform the attacker sends a flood of messages to overwhelm the end-device and make it unavailable to serve genuine message packets.



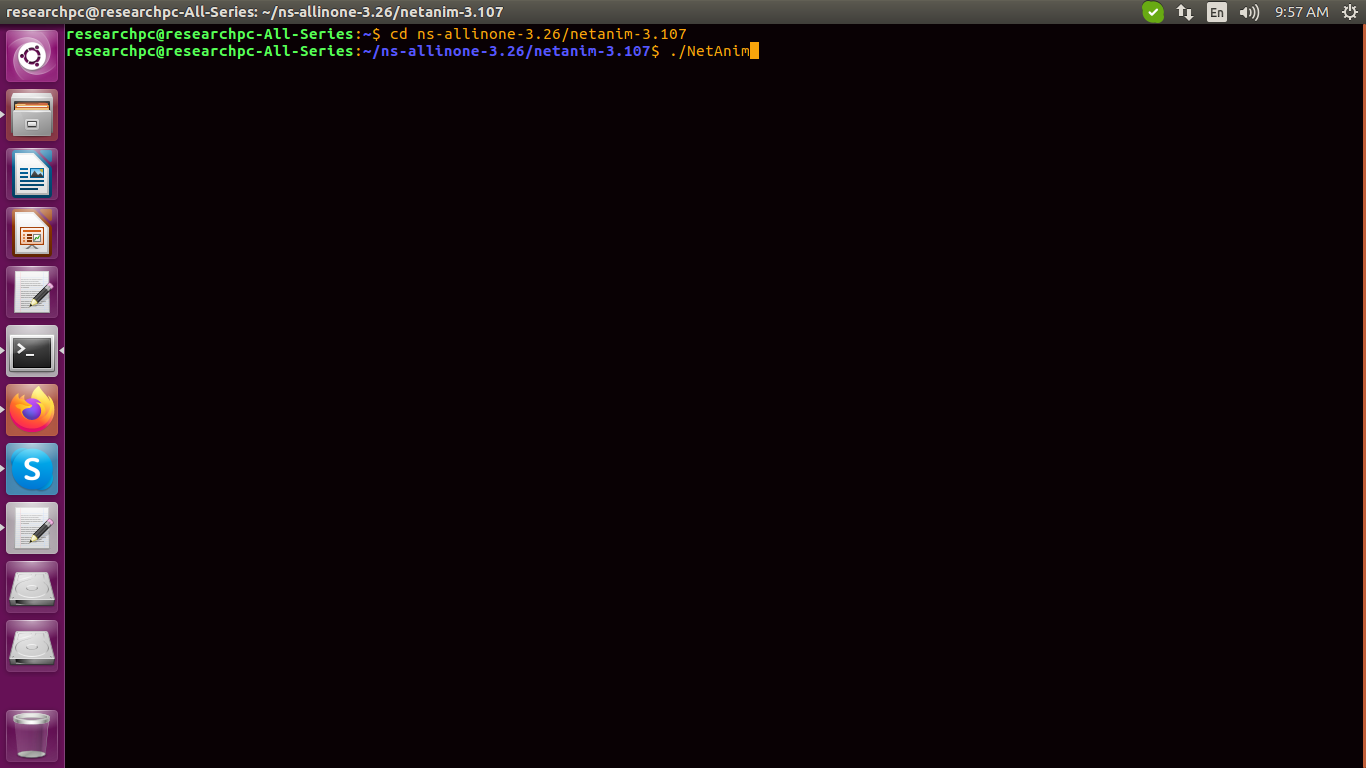
Then next we perform the non-FL version of intrusion detection process, in this process we Classified the IoT nodes into normal or attacked.



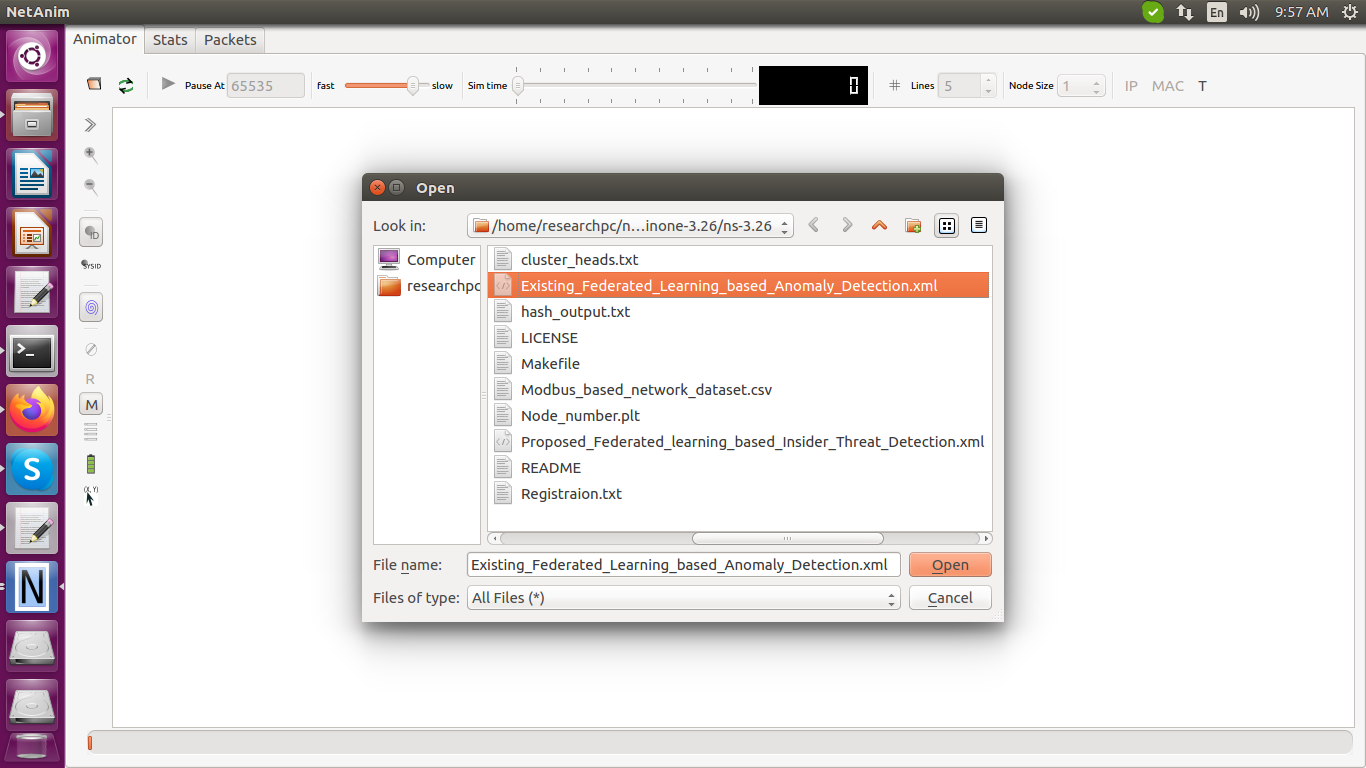
Change the location for get the netanimator result



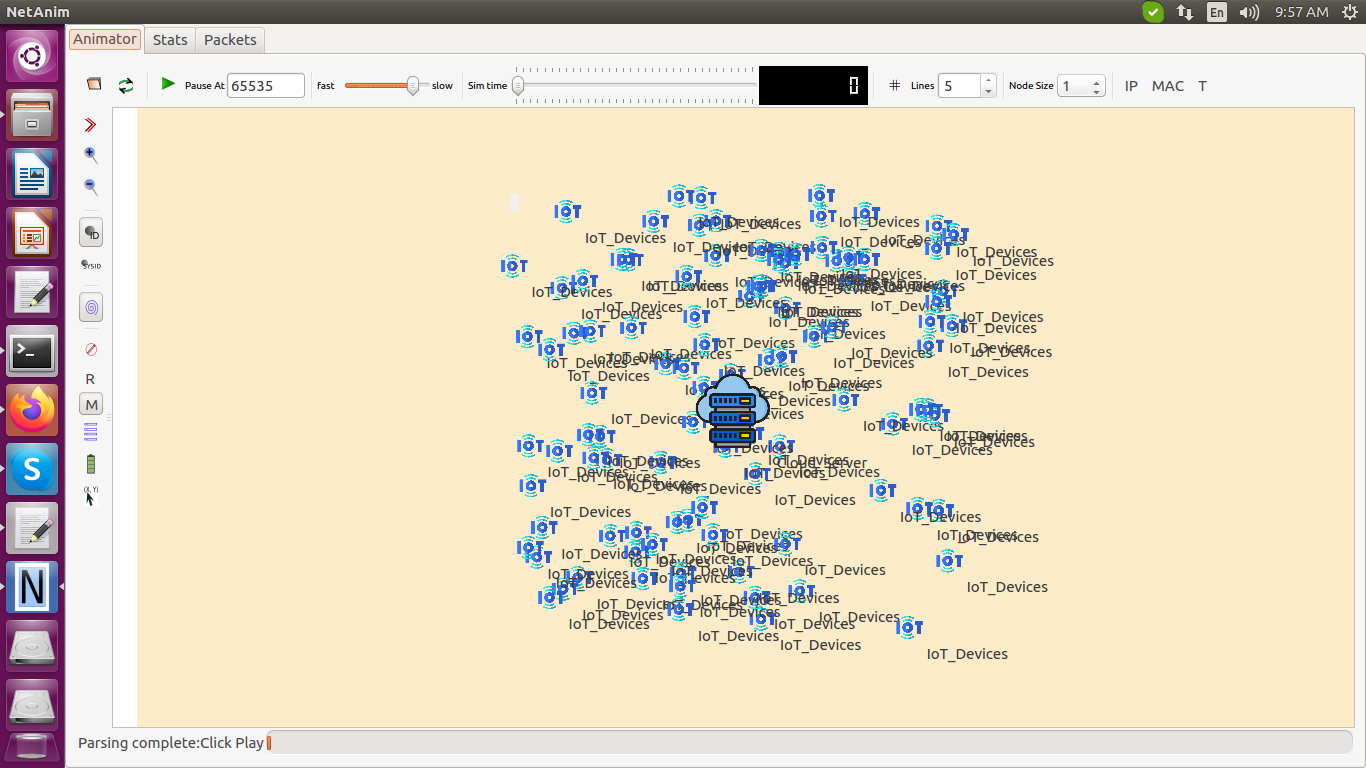
execute the command



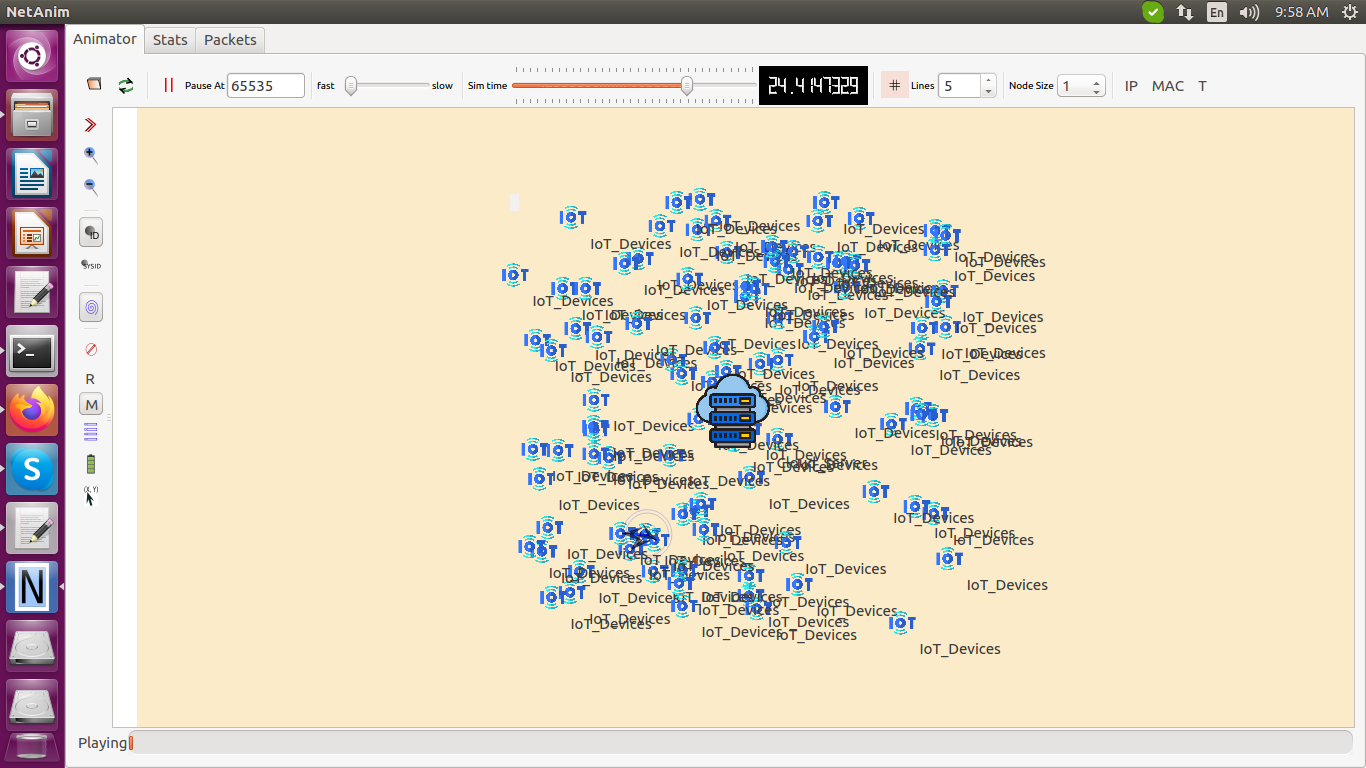
select the xml file

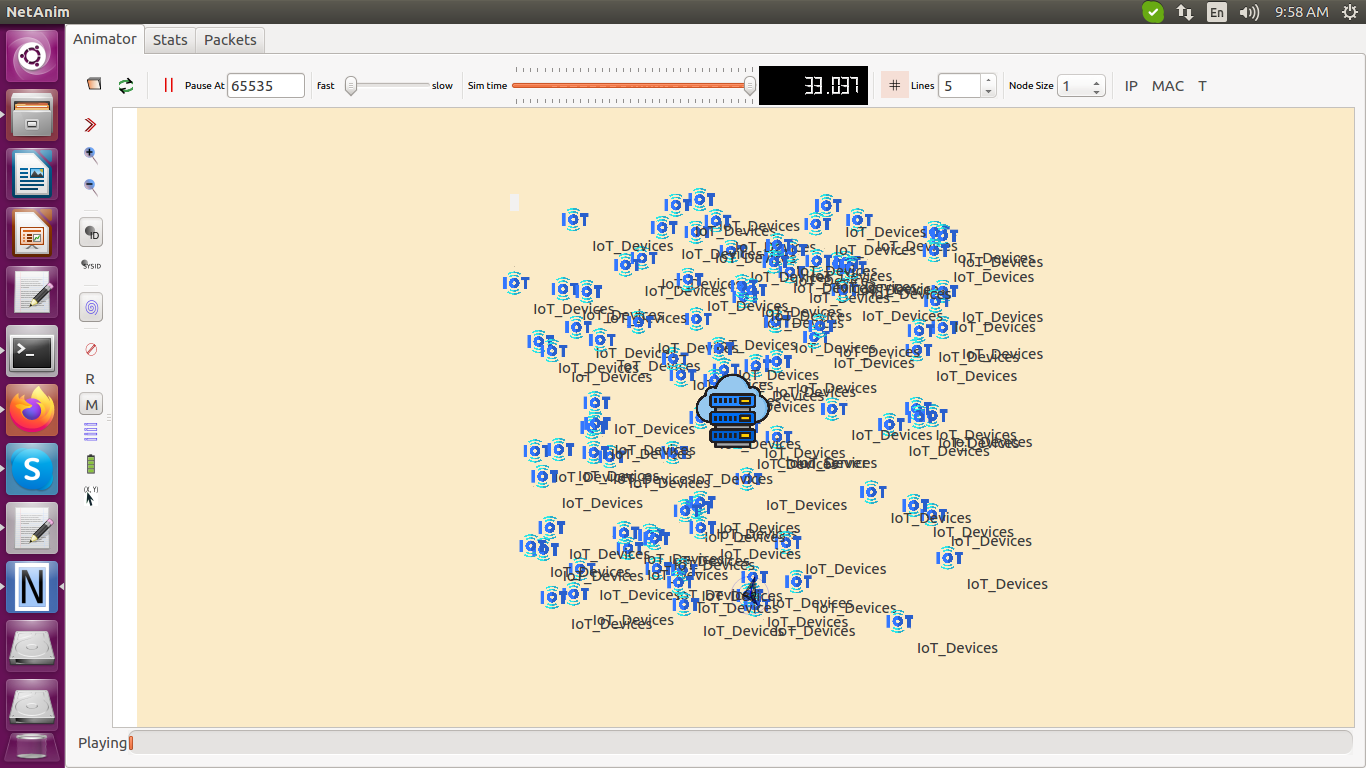


get the simulation



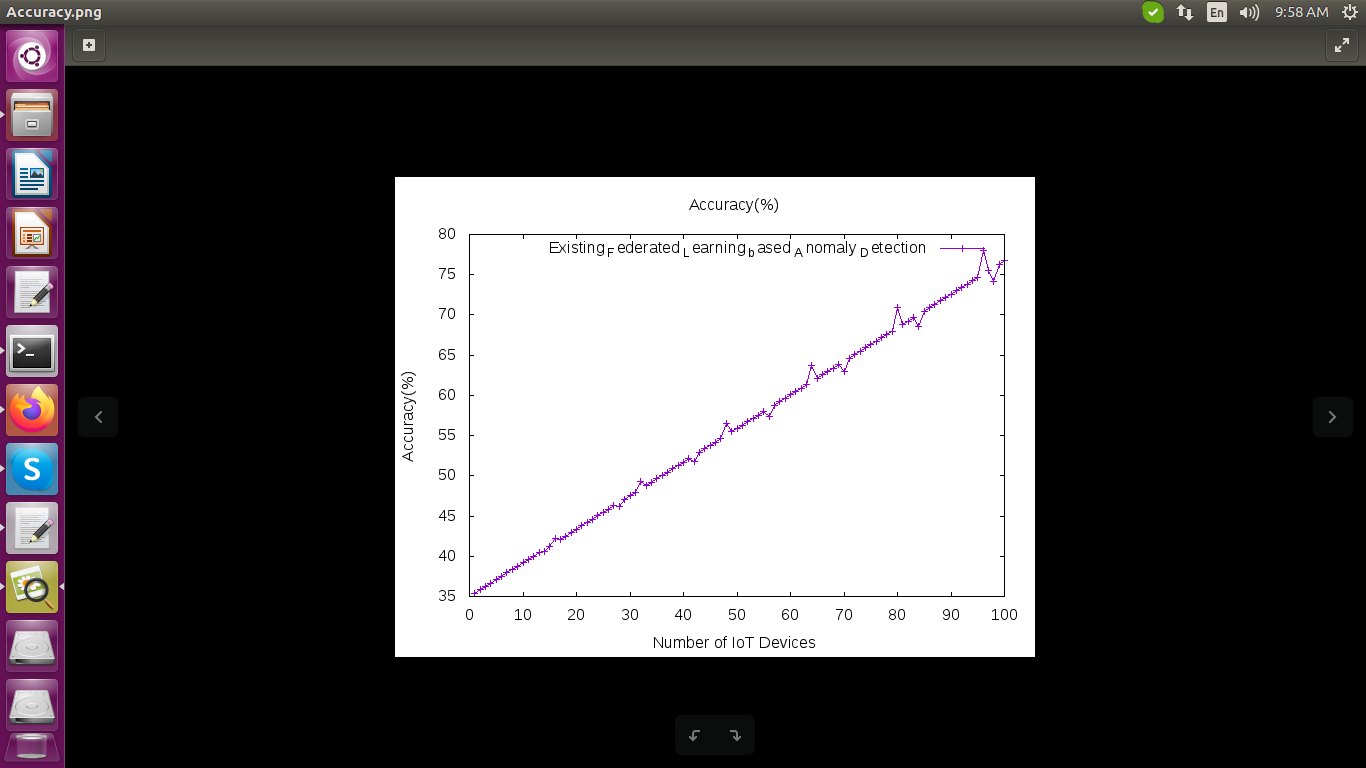




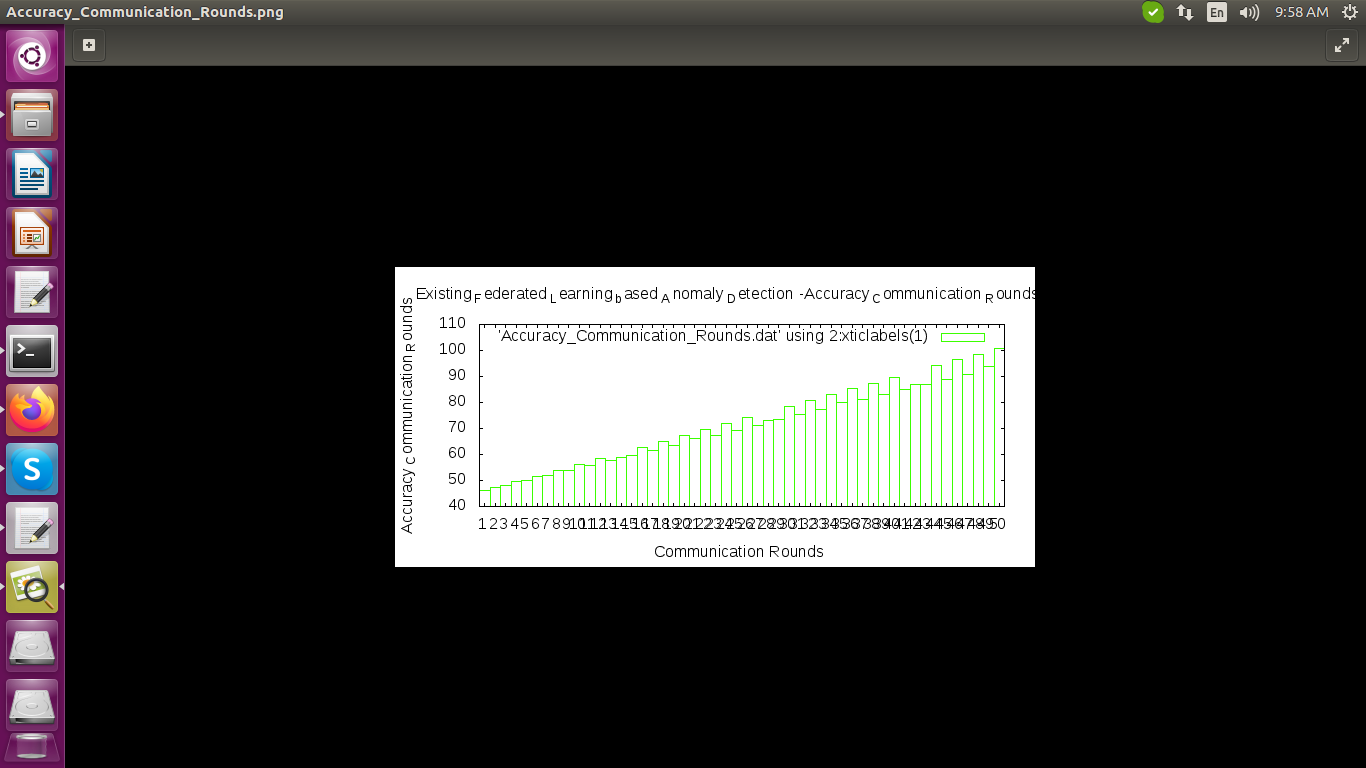


The Existing approach is validated through several performance metrics such as,

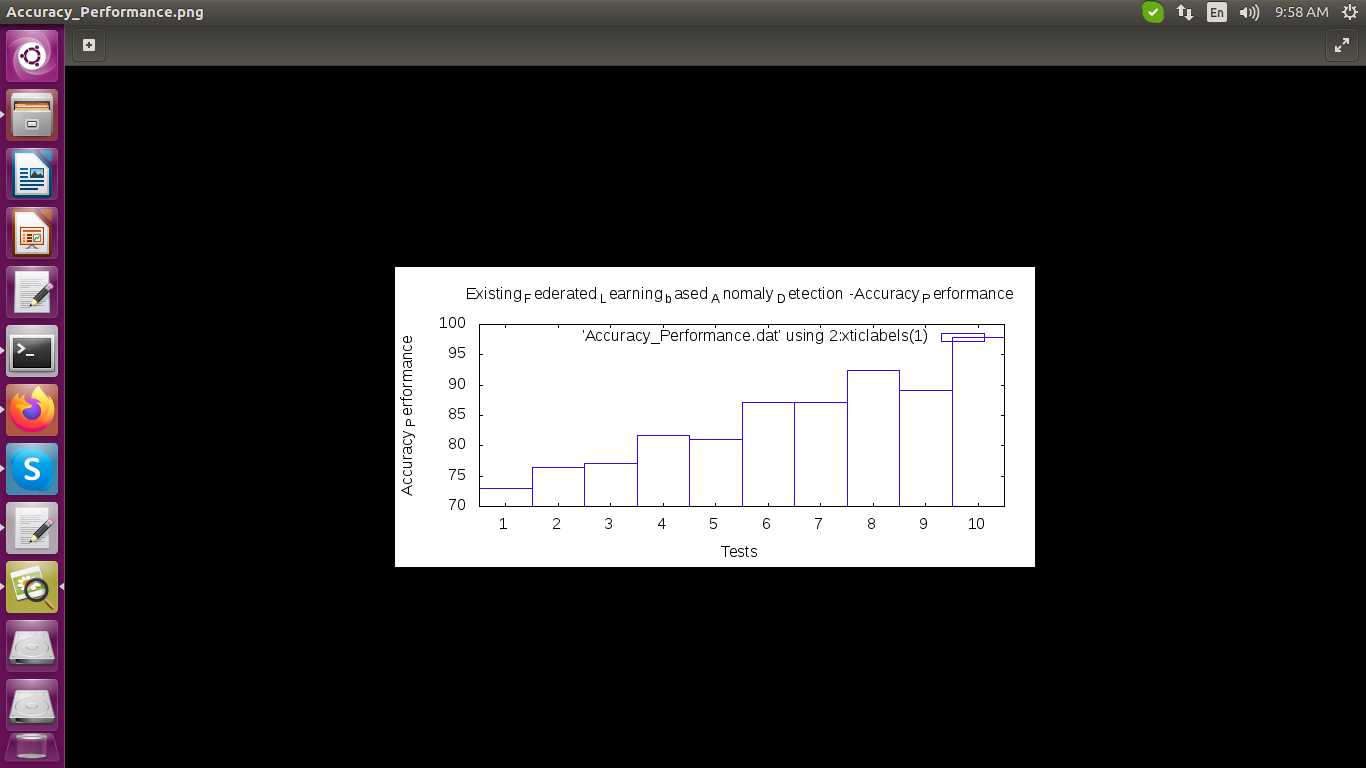
The results graph for Accuracy(%) vs. Number of IoT devices



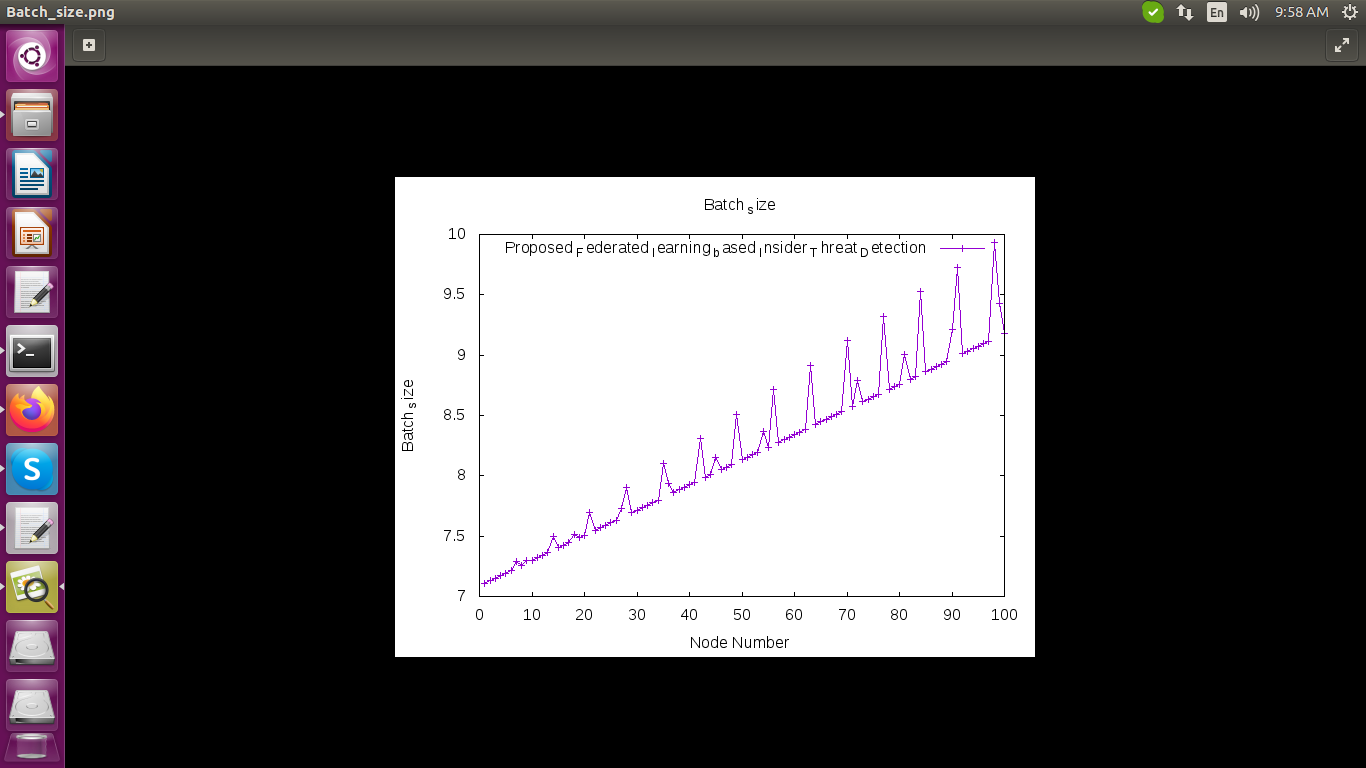
The results graph for Accuracy vs. Communication Rounds



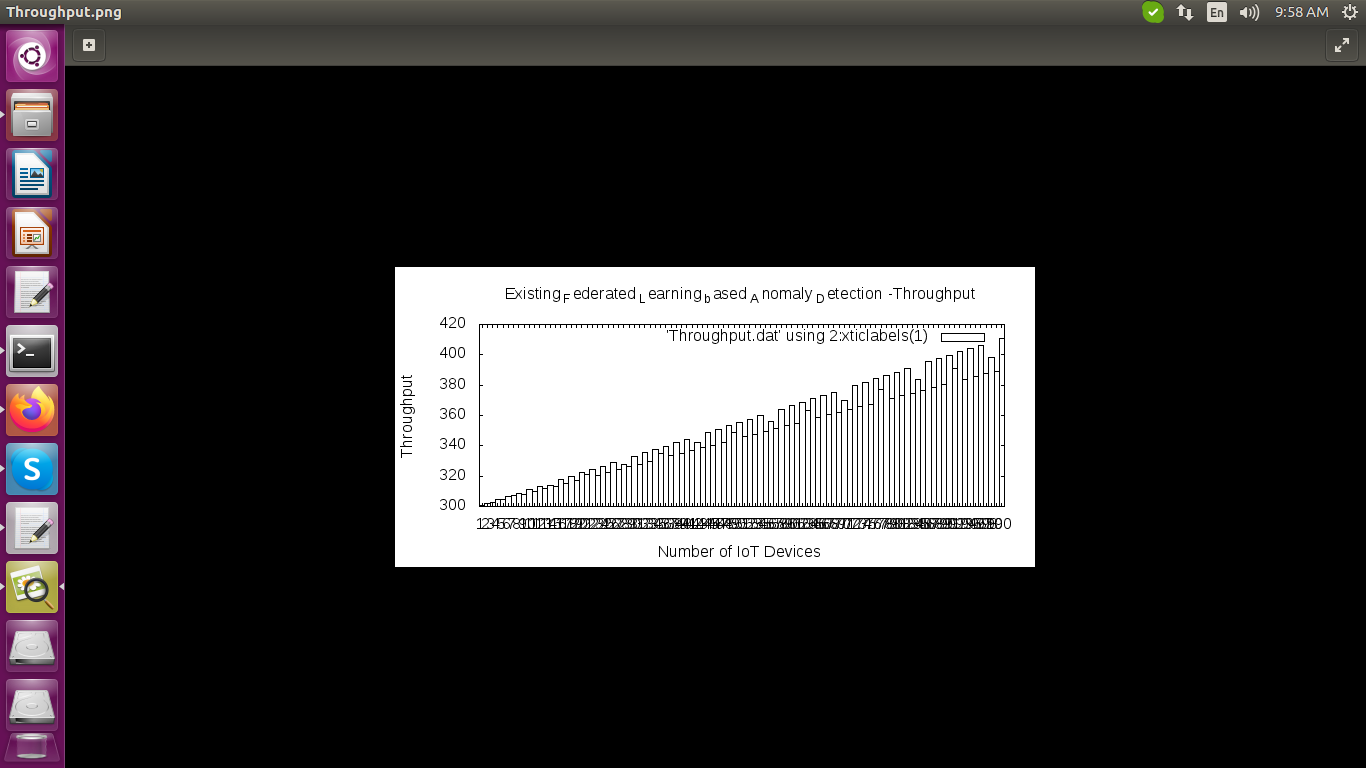
The results graph for Accuracy vs. Tests



The results graph for Batch size vs. Node number



The results graph for Troughput vs. Number of IoT Devices



The results graph for Trust Value vs. Time Period

