REPLAY ATTACK ON IOT DEVICES

Thesis submitted to the SASTRA Deemed to be University
in partial fulfillment of the requirements
for the award of the degree of

B.Tech. Electronics & Communication Engineering

Submitted by

NIVETHA S - 123004168

PUJITA V - 123004189

VARSHA R - 123004268

June 2023

SASTRA
ENGINEERING - MANAGEMENT - LAW - SCIENCES - HUMANITIES - EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

THANJAVUR, TAMILNADU, INDIA- 613401

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

THANJAVUR, TAMILNADU, INDIA- 613401

Bonafide Certificate

This is to certify that the thesis titled "**Replay Attack on IOT devices**" submitted in partial fulfilment of the requirements for the award of the degree of B. Tech. Electronics & Communication Engineering to the SASTRA Deemed to be University, is a bona-fide record of the work done by **Ms. NIVETHA S (123004168), Ms. PUJITA V (123004189), Ms. VARSHA R (123004268)** during the final semester of the academic year 2022-23, in the **School of Electrical & Electronics Engineering**, under my supervision. This thesis has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other similar title to any candidate of any University.

Signature of Project Supervisor:

Name with Affiliation: Dr. Sriranjani R, Sr. Assistant Professor, SEEE, SASTRA University

Date:

Project Viva-voce held on ________________________

Examiner 1 **Examiner 2**

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR- 613401

Declaration

We declare that the thesis titled **"Replay Attack on IoT Devices"** submitted by us is an original work done by us under the guidance of **Dr. Sriranjani R, Sr. Assistant Professor School of Electrical and Electronics Engineering, SASTRA Deemed to be University** during the final semester of the academic year 2022-23, in the **School of Electrical and Electronics Engineering**. The work is original and wherever We have used materials from other sources, we have given due credit and cited them in the text of the thesis. This thesis has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, or other similar title to any candidate of any University.

Signature of the candidate(s): S.Nivethe

Pujita.V

R. Varsha

Name of the candidate(s): NIVETHA S

PUJITA V

VARSHA.R

Date :

ACKNOWLEDGEMENTS

First of all, I would like to thank God Almighty for his endless blessings.

I would like to express my sincere gratitude to **Dr. S. Vaidhyasubramaniam, Vice Chancellor, SASTRA DEEMED UNIVERSITY** for his encouragement and for providing us a platform to improve our knowledge and build ourselves.

I would forever remind grateful and I would like to thank **R.Chandramouli, Registrar, SASTRA DEEMED TO BE UNIVERSITY, and Dr. Thenmozhi K, Dean, School of Electrical and Electronics Engineering** for their overwhelming support and encouragement.

I am extremely grateful to **Dr. Sridhar K, Associate Dean, ECE** for his constant support, motivation, and academic help extended over the past three years of my life in the School of Electrical and Electronics Engineering.

I would like to thank our project coordinators, **Dr. Ramya Vijay, (AP-III/ECE/SEEE)** and **Dr. Padmapriya Pravinkumar, (Associate Professor/ECE/SEEE)** for their constant guidance and for allotting sufficient time to complete the project in a smooth and peaceful manner.

I would specially thank and express my gratitude to my guide **Dr. Sriranjani R, Sr. Assistant Professor (EEE/SEEE)** for providing me an opportunity to do this project and for his guidance and support in successfully completing this project.

I would also like to thank all the Teaching and Non-teaching faculty, and all other people who have directly or indirectly helped us through their support, encouragement, and all other assistance extended for the completion of my project.

Finally, I thank my parents and all others who have helped me to acquire this interest in the project and aided us in completing it within the deadline without so many struggles.

ABSTRACT

Among all identified cyber security threats in IOT, replay attack (Playback attack) is one of the prominent threats that may cause confusions or wrong decision making in control operations of IoT devices. Although the protection mechanisms for IOT devices are being updated over time, the security measures cannot efficiently prevent Replay attacks. IoT communication setup like Zigbee network is established using XCTU software. Then the dataset that collected during normal transmission and during replay attack has been fed as input to the pool of Deep Learning models for classification. Deep Learning Model for intrusion detection is considered in this study. Further, the dataset is processed and analysed by defining Loss function, Optimizer and metrics. The intrusion detection for replay attack has been tested using deep learning LSTM model. The Obtained Experimental results illustrated better classification with hyperparameters. Different kinds of cyber-attack detection models are developed in further work.

Specific Contribution

Zigbee to Zigbee communication.

Sensor data sent to cloud using IoT analytics platform service Thing Speak.

Developed LSTM model for intrusion detection, further developed to bidirectional LSTM model.

Specific Learning

Setting the ZigBee network

Deep Learning

LSTM model

Signature of the Guide	Student Reg. No : 123004268

Name : Dr. Sriranjani.R	Name : Varsha R

ABSTRACT

Replay attack is one of the most significant cyber security risks to IoT that could lead to errors in judgement or misunderstanding when controlling IoT devices. Even though IOT device security safeguards are constantly being improved, replay attacks are still difficult to effectively stop. XCTU software is used to set up IoT communication networks like the Zigbee network. After that, a collection of Deep Learning models of categorization were given the dataset that was gathered during regular transmission and during the replay assault. In this study, a deep learning model enabling identifying intrusions is taken into account. Additionally, a Loss function, an Optimizer, and metrics are defined in order to process and examine the dataset. Deep learning LSTM model has been tested for recognising intrusions for replay attack. The obtained experimental findings showed that using hyperparameters improved classification. In additional research, many internet attack detection techniques are developed.

Specific Contribution

Zigbee to Zigbee communication

Dataset capture using a network protocol analyser Wireshark

Developed LSTM model for intrusion detection, Further developed to bidirectional LSTM Model.

Specific Learning

Setting the ZigBee network

Deep Learning

LSTM model

Signature of the Guide

Student Reg. No : 123004189

Name **:** Dr. Sriranjani.R

Name : Pujita V

ABSTRACT

One of the biggest cyber security threats to IoT is replay attack, which may cause users to make mistakes or misunderstand instructions when operating IoT devices. Replay attacks remain challenging to effectively thwart, despite ongoing advancements in IOT device security measures. IoT networks for communication such as the Zigbee network are set up using XCTU software. The dataset that was obtained during regular transmit and during the replay invasion was then supplied to a group of Deep Learning categorization models. A model based on deep learning that makes it possible to identify intrusions is included in this work. In order to handle and analyse the dataset, further definitions include an error function, an Optimizer, other metrics. Replay attack detection using a deep-learning LSTM model was recently worked. The results of the experiment demonstrated that classification was enhanced by the use of hyperparameters. Numerous methods for detecting internet attacks are created in further study.

Specific Contribution

Zigbee to Zigbee communication

Developed LSTM model for intrusion detection.

Developed to bidirectional LSTM model to detect replay attack

Specific Learning

Setting the ZigBee network

Deep Learning

LSTM model

Signature of the Guide	Student Reg. No : 123004168
Name: Dr. Sriranjani.R	Name : Nivetha S

List of Figures

List of Table

ABBREVIATIONS:

Bi-LSTM	– Bidirectional Long Short-Term Memory
CNN	- Convolutional Neural Networks
DNN	- Deep Neural Networks
IEEE	- Institute of Electrical and Electronics Engineers
LSTM	- Long Short-Term Memory
MitM	- Man in Middle attack
PHY	- Physical Layer
RNN	- Recurrent Neural Networks

Table of Contents

Title **Page No**

CHAPTER 1

INTRODUCTION

A cyber-attack is a hostile, unauthorised third-party intrusion into a system or network. A network of computers, database, or private device may have confidential data that is being stolen or attempted to be deleted. The perpetrator of this incident is a hacker. The importance of data to us has increased recently. The safety and security of data are crucial. We cannot take the chance of some private and delicate information getting into the hands of someone else. However, there are sporadic instances where a third-party can access our information. Any unauthorised intrusion or hack that jeopardises the data's veracity, integrity, or confidentiality is referred to as a security attack.

When outsiders tamper with a two-party transaction, attacks referred to as "man-in-the-middle" (MitM) or "eavesdropping" occur. Once the communication has been stopped, the intruders may filter and steal data.

For MitM assaults, two common entry points exist:

- On unsecure public Wi-Fi, attackers can put themselves in between an individual's device and the network. Without even noticing it, the visitor provides the attacker with all available information.
- Once malware has spread to a device, an intruder may download programmes that manage all the individual's data.

1.1. REPLAY ATTACK

A security attack known as a replay attack targets data delivered across a network.

In this assault, the hacker, or any other individual with unauthorized access intercepts the traffic and, in place of the original sender, delivers communication to its intended recipient. The communication was really sent by the attacker, but the recipient believes it to be a legitimate message. The Replay Attack's primary characteristic is that the user would get the message multiple times, hence the name.

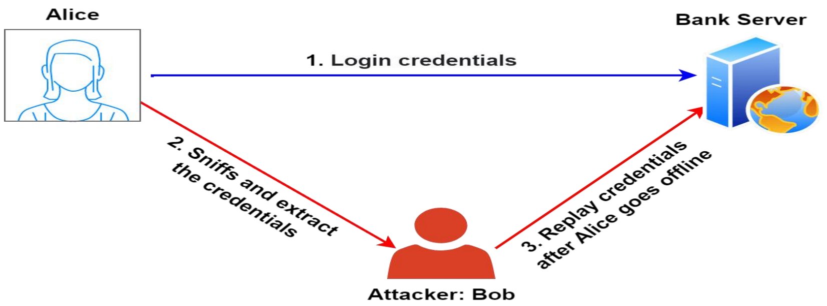

Alice
Bank Server
1. Login credentials
2. Sniffs and extract the credentials
3. Replay credentials after Alice goes offline
Attacker: Bob

Fig 1. Replay Attack

- The following steps are commonly included in replay attacks:
- ➢ A packet sniffing programme is installed on a network by an attacker.
- ➢ Filtered data is transferred to the attacker's machine.
- ➢ Attacker either sends back packets of interest right away or manipulates them before resending.
- ➢ Upon receiving a response from the target server, the attacker is now thought to be the message's sender.

- Defending against replay attacks:
- ➢ Timestamp method: If a timestamp is added to the data, protection against such attackers may be achievable. If a data's timestamp exceeds a certain threshold, it could be rejected and the sender might be prompted to submit information again.
- ➢ Session key technique: Using a session key is an additional preventative measure. This key cannot be reused and can only be used one (by transmitter and recipient) per transaction.

1.2. DIGI XBEE S2C 802.15. MODULE

A radio frequency (RF) module known as the XBee S2C is made specifically for wireless connection or data transmission. It uses ZigBee mesh communication protocols, which are built on top of IEEE 802.15.4 PHY. The module offers wireless access to all ZigBee endpoints, including those made by other manufacturers, in ZigBee mesh networks.

Fig 2. DIGI XBEE S2C 802.15. MODULE

- Electrical characteristics and features
- ➢ 2.4 to 2.5GHz for transmission frequency
- ➢ 16 Direct Sequential Channels are available.
- ➢ Range of the supply voltage: +2.1V to +3.6V

- How Zigbee Network works?

Three different types of devices are supported by ZigBee:

Coordinator, router, and end device.

- ➢ The coordinator is a fully working device with capabilities for network management.
- ➢ In tree & mesh networks only, the ZigBee router additionally serves as an entirely functioning component.
- ➢ The only functions of a ZigBee end device are to transmit and acquire packets.

1.3. WIRESHARK:

Fig 3. Wireshark

Wireshark is the name of the most widely used packet sniffer in the world. in the same league as other packet sniffers. It does three things:

- ➢ Packet Capture: Wireshark continuously scans a network link and records whole streams of communication, potentially thousands of packets at once.
- ➢ Filtering: All this random live data may be sliced and diced by Wireshark. By using a filter, the required information is obtained.

- Visualisation: Wireshark is a reliable packet sniffer which allows us to delve deeply into the core of a network packet. It even allows us to view whole conversations and network feeds.

In order to gather network packets and show them in detail, use Wireshark. after the separation of these packets. It is utilised for offline or in real-time analysis. This tool aids in network analysis and, eventually, network security by helping to magnify, filter, and dig down into network data in order to identify the source of issues.

1.4. DEEP LEARNING:

Artificial neural networks are the foundation of the machine learning subfield known as deep learning. It can recognise intricate links and patterns in data. Due to improvements in computational capacity and the accessibility of massive datasets, it has grown in popularity recently. as it is built on deep neural networks (DNNs), often referred to as artificial neural networks (ANNs). These artificial neural networks are built to learn from massive quantities of data and are modelled after the structure and operation of organic neurons in the human brain.

- Deep learning, a subfield of machine learning, uses neural networks to model and solve challenging problems. Layers of nodes that are linked that process and change data make up neural networks, which have been modelled after the composition and operation of the human brain.
- The utilisation of deep artificial neural networks, which include numerous layers of linked nodes, is the main feature of deep learning. By identifying hierarchical patterns and characteristics in the data, these networks may develop complicated representations of the data. Without explicit feature engineering, deep learning algorithms may automatically learn from data and get better.
- Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs) are a few of the well-known Deep Learning designs.

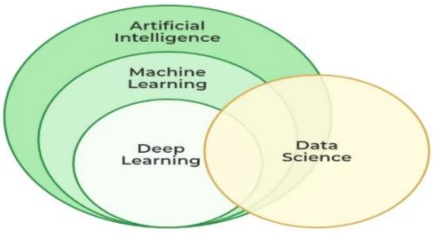


Fig 4. Deep Learning

1.4.1. RECURRENT NEURAL NETWORK

RNN operates on the premise of preserving the result of a certain layer and feeding it back to the input to forecast the layer's output. It is a Feed-Forward Neural Network. RNN is based on the idea of keeping the result of a particular layer and sending it over to the inputs to predict the layer's output.

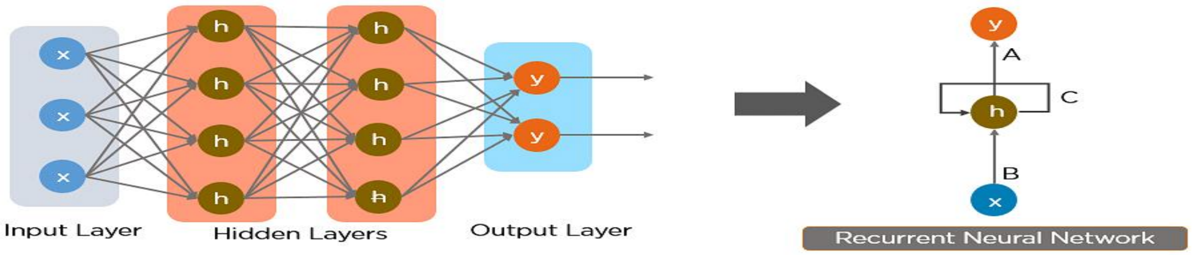


Fig 5: Simple Recurrent Neural Network

RECURRENT NEURAL NETWORK (RNN) VARIATION

To address issues such as disappearing gradient and explosive gradient descent, various more sophisticated variants of RNNs have been developed, some of which are as follows:

1. Bidirectional LSTM (Bi-LSTM)
2. Long Short-Term Memory (LSTM)

- ## LONG SHORT-TERM MEMORY

A type of recurrent neural network is the LSTM. This design is created specifically to address the issue of vanishing and bursting gradients. Furthermore, this network is better at preserving long-distance connections and recognising the association among values at the start and end of a series.

The LSTM model has expressions, namely gates. In reality, there are three kinds of gates:

- ➢ The forget gate determines the amount of data the cell with memory will get from the preceding step's memory cell.

- The update (input) gate determines whether or not the memory cell is going to be updated. It also determines how much information a present memory cell receives from a possible new memory cell.
- The output gate determines the significance of the following hidden state.

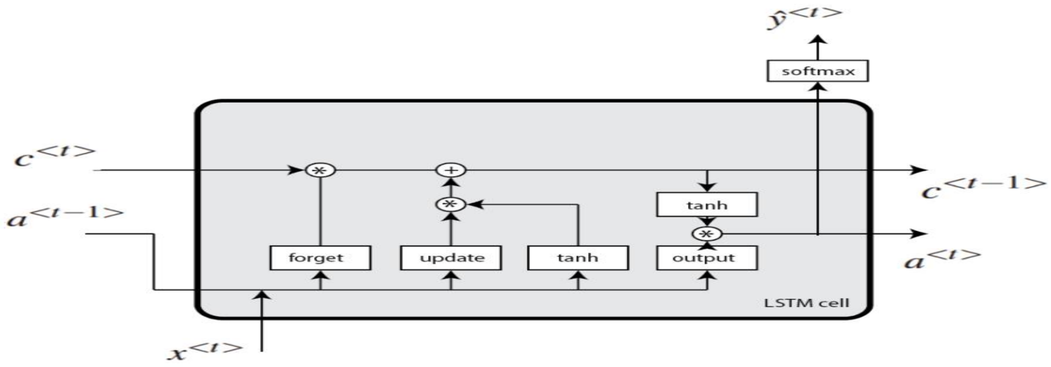


Fig 6. LSTM

- **BIDIRECTIONAL LSTM**

- Bidirectional LSTM is a kind of recurrent neural network that is mostly used for natural language processing. Unlike traditional LSTM, the input travels in the two directions, and it may use data from both sides.
- It's also an effective tool for simulating the sequential relationships between words and sentences in both directions.
- To summarise, Bi-LSTM adds an additional LSTM layer that reverse the path of information flow. In a nutshell, it implies that the data sequence is reversed in the extra LSTM layer.
- The outputs of both LSTM layers are then combined in a variety of methods, including average, sum, multiplication, and splicing. The unfolded Bi-LSTM is shown in the image below as an example:

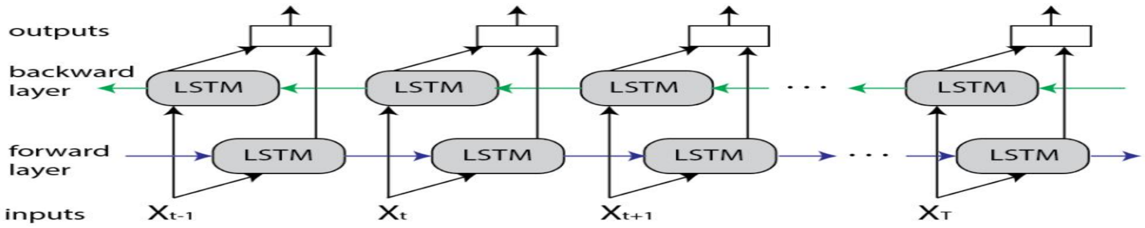


Fig 7. Bi- LSTM

1.5 MOTIVATION

Since, the IoT networks are being launched with replay attack due to their lack of security. To know more about the replay attack and to analyse them using deep learning algorithms this project has been taken.

CHAPTER 2
OBJECTIVES

To develop an intrusion Detection system for replay attack on IoT devices based on Deep Learning.

The objectives are:

- To create a IoT communication network using Digi XBee S2C 802.15.4 Modules.
- To send sensor data through Thing Speak using ZigBee network and simulating replay attack.
- To capture the traffic using Wireshark software and using this as the dataset for applying deep learning algorithms.
- To analyse the output of the LSTM and Bi-LSTM deep learning algorithms.

CHAPTER 3

METHODOLOGY:

In this project, the primary stage is formation of zigbee network, then the second stage is collect the data in zigbee network using Wireshark. In the third stage, using the TensorFlow the model is trained using LSTM and Bi-LSTM. To test the data, the large set of train dataset is included to find the performance metrics. Using the result, the replay attack is detected. The flowchart of this methodology is show below:

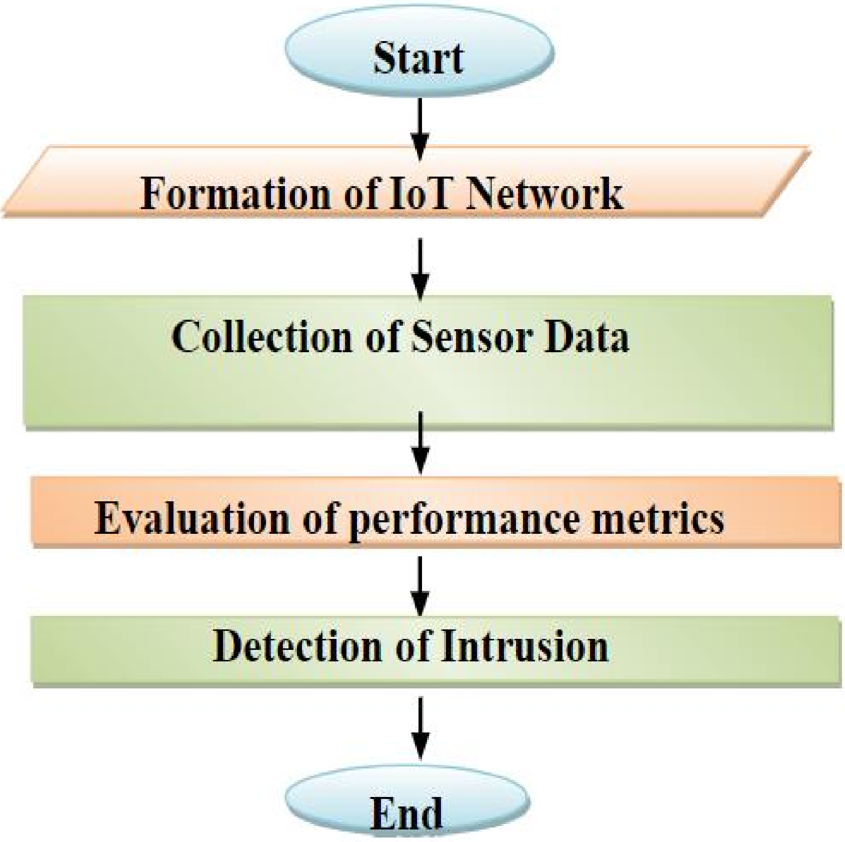


Fig 8. Module Flowchart

3.1 TESTBED SETUP:

The testbed setup consists of the 2 Digi XBee S2C 802.15.4 modules as Coordinator and Router, DHT11 temperature and humidity sensor, Arduino Uno, ESP8266, monitor to view using Thing Speak, capture the traffic using Wireshark. The block diagram of testbed setup is shown below:

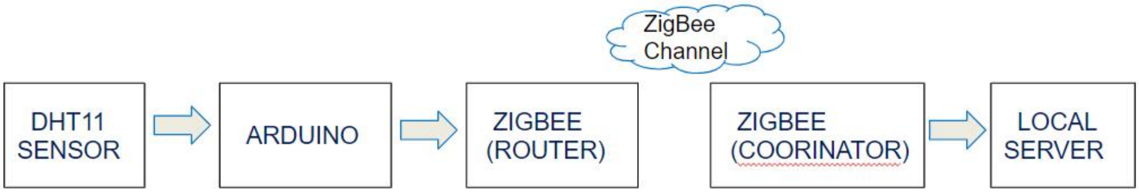


Fig 9. Proposed Testbed setup

The testbed setup of ZigBee network with DHT11 sensor is shown below:

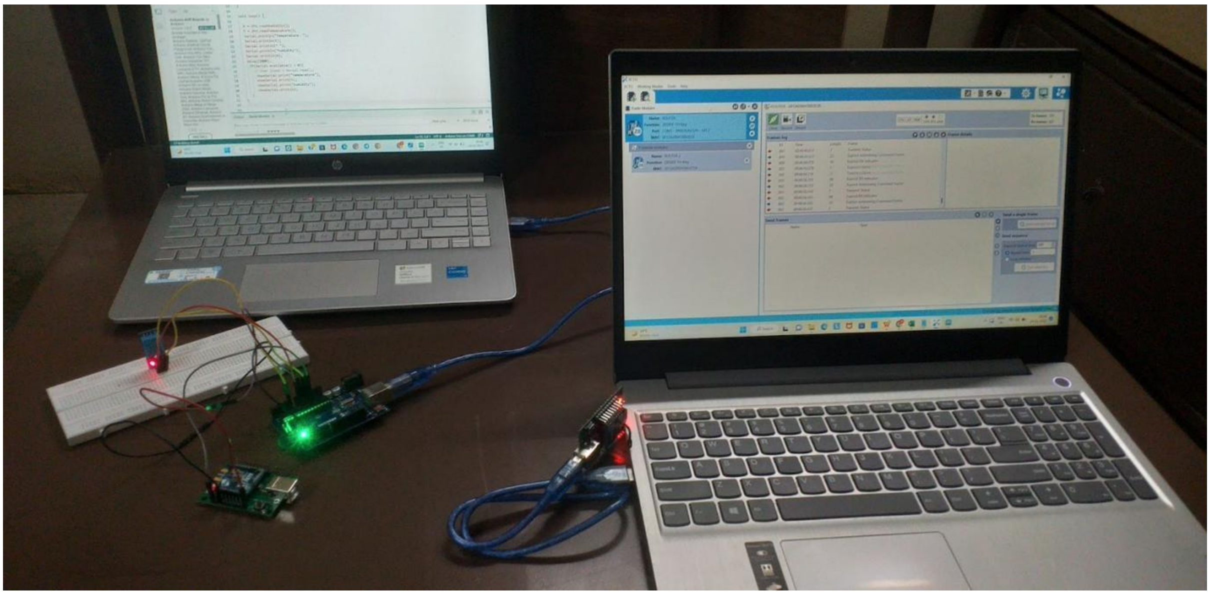
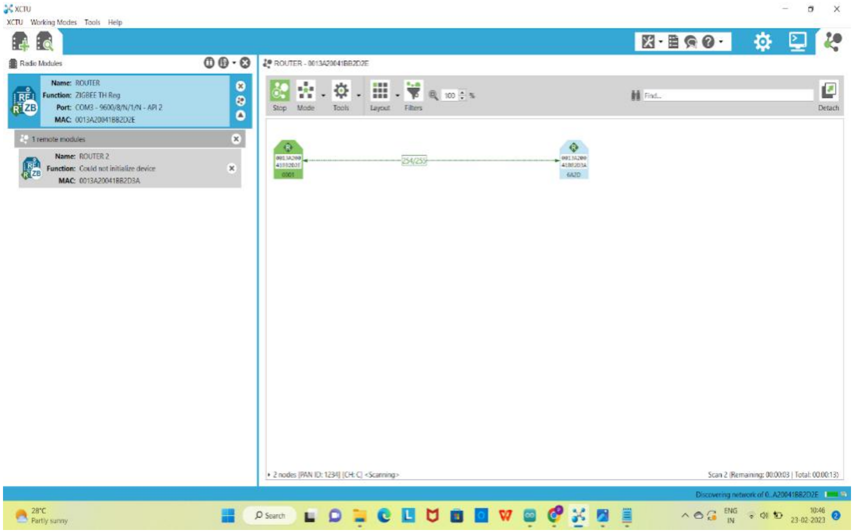


Fig 10. Testbed setup

3.2 STEPS FOLLOWED:

- The testbed is setup using Digi XBee S2C 802.15.4 modules and it is configured using XCTU software. ZigBee connectivity is established using two XBee devices.
- Firmware should be installed in the XBee devices.
- Configure the one module as Coordinator and other module as Router.
- Personal area networks (PANs) identification is known as a PAN ID in XCTU. A distinct ID must be assigned to each network. For both the XBee modules, the same PAN ID and same Channel name is given. The presence of both XBee modules in an identical network is indicated by this.
- Zigbee communication network is formed by swapping destination and source address to both coordinator and router. This shows that the information supplied by the source is correctly received at the target.
- The two XBee devices interact using an Arduino Nano & a NodeMCU.
- In the transmitter side, the router XBee interfaced with Arduino uno and DHT11 temperature and humidity sensor.
- In the receiver side, the coordinator XBee interfaced with ESP8266.
- The temperature and humidity data transfer from one place to another wirelessly in short range i.e., 60 m through ZigBee network.
- This data is analysed and monitored using Thing Speak, IoT analysis platform. It allows to visualize and analyse live data streams in the cloud. It provides instant visualizations of data.
- The network traffic dataset is captured with Wireshark, a network monitoring tool.
- The normal traffic and the attacking traffic in the dataset created are given with 0's and 1's respectively to differentiate in the target column.
- These datasets are given to the LSTM and Bi-LSTM deep learning algorithms for analysis.

In order to figure out the MAC address of the router and coordinator, as well as making sure the router joined XBEE ZigBee coordinator, the Digi XCTU Software discovery mode is used. The ZigBee network setup using XCTU is shown below:



Radio Modules

Name: ROUTER
Function: ZIGBEE TH Reg
Port: COM3 - 9600/8/N/1/N - API 2
MAC: 0013A20041BB2D2E

1 remote modules

Name: ROUTER 2
Function: Could not initialize device
MAC: 0013A20041BB2D3A

ROUTER - 0013A20041BB2D2E

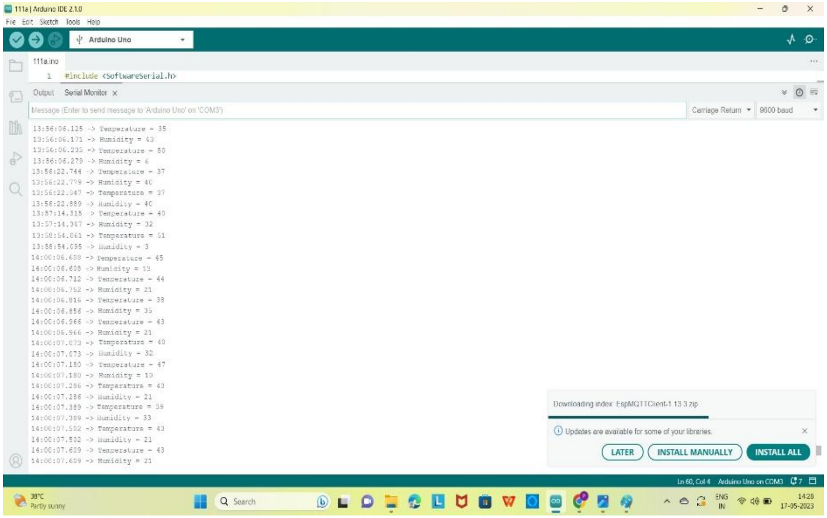
2 nodes [PAN ID: 1234] [CH: C] <Scanning>

Scan 2 [Remaining: 00:00:03 | Total: 00:00:13]

Discovering network of 0...A20041BB2D2E

Fig 11. Network setup in XCTU

Once the code executed in Arduino, the DHT11 temperature and humidity sensor data values is seen in serial monitor of Arduino.



111a.ino

```
#include <SoftwareSerial.h>
```

Message (Enter to send message to 'Arduino Uno' on 'COM3')

Carriage Return   9600 baud

```
13:56:06.125 -> Temperature = 35
13:56:06.171 -> Humidity = 43
13:56:06.235 -> Temperature = 50
13:56:06.279 -> Humidity = 6
13:56:22.744 -> Temperature = 37
13:56:22.779 -> Humidity = 40
13:56:22.847 -> Temperature = 37
13:56:22.889 -> Humidity = 40
13:57:14.315 -> Temperature = 40
13:57:14.347 -> Humidity = 32
13:58:54.061 -> Temperature = 51
13:58:54.095 -> Humidity = 3
14:00:06.608 -> Temperature = 45
14:00:06.608 -> Humidity = 13
14:00:06.712 -> Temperature = 44
14:00:06.752 -> Humidity = 21
14:00:06.816 -> Temperature = 38
14:00:06.856 -> Humidity = 35
14:00:06.966 -> Temperature = 43
14:00:06.966 -> Humidity = 21
14:00:07.073 -> Temperature = 40
14:00:07.073 -> Humidity = 32
14:00:07.180 -> Temperature = 47
14:00:07.180 -> Humidity = 10
14:00:07.286 -> Temperature = 43
14:00:07.286 -> Humidity = 21
14:00:07.389 -> Temperature = 39
14:00:07.389 -> Humidity = 33
14:00:07.502 -> Temperature = 43
14:00:07.502 -> Humidity = 21
14:00:07.609 -> Temperature = 43
14:00:07.609 -> Humidity = 21
```

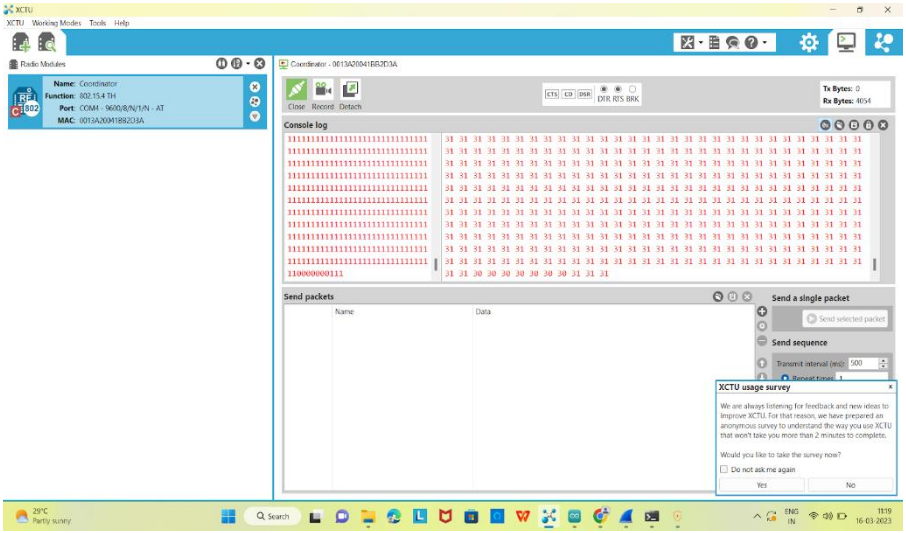
Downloading index: EspMQTTClient-1.13.3.zip

Updates are available for some of your libraries.

Ln 60, Col 4   Arduino Uno on COM3

Fig 12. Sensor data in Arduino Serial Monitor

The sensor data collected in coordinator ZigBee and it had been checked through the terminal of XCTU Software.



Name: Coordinator
Function: 802.15.4 TH
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A20041BB2D3A

Tx Bytes: 0
Rx Bytes: 4054

```
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
111111111111111111111111111111  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
110000000111                    31 31 30 30 30 30 30 30 30 31 31 31
```

Transmit interval (ms): 500
Repeat times: 1

XCTU usage survey

We are always listening for feedback and new ideas to improve XCTU. For that reason, we have prepared an anonymous survey to understand the way you use XCTU that won't take you more than 2 minutes to complete.

Would you like to take the survey now?

□ Do not ask me again

Yes No

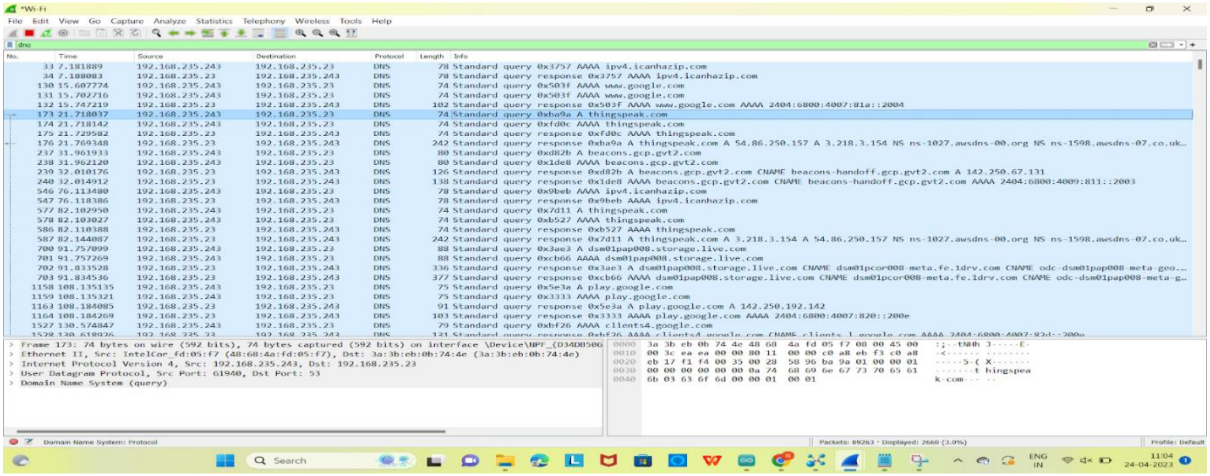
29°C Partly sunny

ENG IN 11:19 16-03-2023

Fig 13. Data collection

3.3 DATASET:

Wireshark is a reliable packet. It allows us to view whole conversations and network feeds. This tool aids for network analysis, filter, and dig down into network data in order to identify the source of issues with the help of network parameters like time, source and destination address, protocol, length, information about that packet etc..,


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Search

dns

```
No.   Time        Source            Destination       Protocol Length Info
   33 7.181889    192.168.235.243   192.168.235.23    DNS       78 Standard query 0x3757 AAAA ipv4.icanhazip.com
   34 7.188083    192.168.235.23    192.168.235.243   DNS       78 Standard query response 0x3757 AAAA ipv4.icanhazip.com
  130 15.607774   192.168.235.243   192.168.235.23    DNS       74 Standard query 0x503f AAAA www.google.com
  131 15.702716   192.168.235.243   192.168.235.23    DNS       74 Standard query 0x503f AAAA www.google.com
  132 15.747219   192.168.235.23    192.168.235.243   DNS      102 Standard query response 0x503f AAAA www.google.com AAAA 2404:6800:4007:81a::2004
  173 21.718037   192.168.235.243   192.168.235.23    DNS       74 Standard query 0xba9a A thingspeak.com
  174 21.718142   192.168.235.243   192.168.235.23    DNS       74 Standard query 0xfd0c AAAA thingspeak.com
  175 21.729582   192.168.235.23    192.168.235.243   DNS       74 Standard query response 0xfd0c AAAA thingspeak.com
  176 21.769348   192.168.235.23    192.168.235.243   DNS      242 Standard query response 0xba9a A thingspeak.com A 54.86.250.157 A 3.218.3.154 NS ns-1027.awsdns-00.org NS ns-1598.awsdns-07.co.uk…
  237 31.961933   192.168.235.243   192.168.235.23    DNS       80 Standard query 0xd82b A beacons.gcp.gvt2.com
  238 31.962120   192.168.235.243   192.168.235.23    DNS       80 Standard query 0x1de8 AAAA beacons.gcp.gvt2.com
  239 32.010176   192.168.235.23    192.168.235.243   DNS      126 Standard query response 0xd82b A beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com A 142.250.67.131
  240 32.014912   192.168.235.23    192.168.235.243   DNS      138 Standard query response 0x1de8 AAAA beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com AAAA 2404:6800:4009:811::2003
  546 76.113480   192.168.235.243   192.168.235.23    DNS       78 Standard query 0x9beb AAAA ipv4.icanhazip.com
  547 76.118386   192.168.235.23    192.168.235.243   DNS       78 Standard query response 0x9beb AAAA ipv4.icanhazip.com
  577 82.102950   192.168.235.243   192.168.235.23    DNS       74 Standard query 0x7d11 A thingspeak.com
  578 82.103027   192.168.235.243   192.168.235.23    DNS       74 Standard query 0xb527 AAAA thingspeak.com
  586 82.110388   192.168.235.23    192.168.235.243   DNS       74 Standard query response 0xb527 AAAA thingspeak.com
  587 82.144087   192.168.235.23    192.168.235.243   DNS      242 Standard query response 0x7d11 A thingspeak.com A 3.218.3.154 A 54.86.250.157 NS ns-1027.awsdns-00.org NS ns-1598.awsdns-07.co.uk…
  700 91.757099   192.168.235.243   192.168.235.23    DNS       88 Standard query 0x3ae3 A dsm01pap008.storage.live.com
  701 91.757269   192.168.235.243   192.168.235.23    DNS       88 Standard query 0xcb66 AAAA dsm01pap008.storage.live.com
  702 91.833528   192.168.235.23    192.168.235.243   DNS      336 Standard query response 0x3ae3 A dsm01pap008.storage.live.com CNAME dsm01pcor008-meta.fe.1drv.com CNAME odc-dsm01pap008-meta-geo.…
  703 91.834536   192.168.235.23    192.168.235.243   DNS      377 Standard query response 0xcb66 AAAA dsm01pap008.storage.live.com CNAME dsm01pcor008-meta.fe.1drv.com CNAME odc-dsm01pap008-meta-g…
 1158 108.135135  192.168.235.243   192.168.235.23    DNS       75 Standard query 0x5e3a A play.google.com
 1159 108.135321  192.168.235.243   192.168.235.23    DNS       75 Standard query 0x3333 AAAA play.google.com
 1163 108.184085  192.168.235.23    192.168.235.243   DNS       91 Standard query response 0x5e3a A play.google.com A 142.250.192.142
 1164 108.184269  192.168.235.23    192.168.235.243   DNS      103 Standard query response 0x3333 AAAA play.google.com AAAA 2404:6800:4007:820::200e
 1527 130.574847  192.168.235.243   192.168.235.23    DNS       79 Standard query 0xbf26 AAAA clients4.google.com
 1528 130.618926  192.168.235.23    192.168.235.243   DNS      131 Standard query response 0xbf26 AAAA clients4.google.com CNAME clients.l.google.com AAAA 2404:6800:4007:82d::200e
```

```
> Frame 173: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{D34DB506
> Ethernet II, Src: IntelCor_fd:05:f7 (48:68:4a:fd:05:f7), Dst: 3a:3b:eb:0b:74:4e (3a:3b:eb:0b:74:4e)
> Internet Protocol Version 4, Src: 192.168.235.243, Dst: 192.168.235.23
> User Datagram Protocol, Src Port: 61940, Dst Port: 53
> Domain Name System (query)
```

```
0000  3a 3b eb 0b 74 4e 48 68  4a fd 05 f7 08 00 45 00   :;··tNHh J·····E·
0010  00 3c ea ea 00 00 80 11  00 00 c0 a8 eb f3 c0 a8   ·<······ ········
0020  eb 17 f1 f4 00 35 00 28  58 96 ba 9a 01 00 00 01   ·····5·( X·······
0030  00 00 00 00 00 00 0a 74  68 69 6e 67 73 70 65 61   ·······t hingspea
0040  6b 03 63 6f 6d 00 00 01  00 01                     k·com··· ··
```

Domain Name System: Protocol

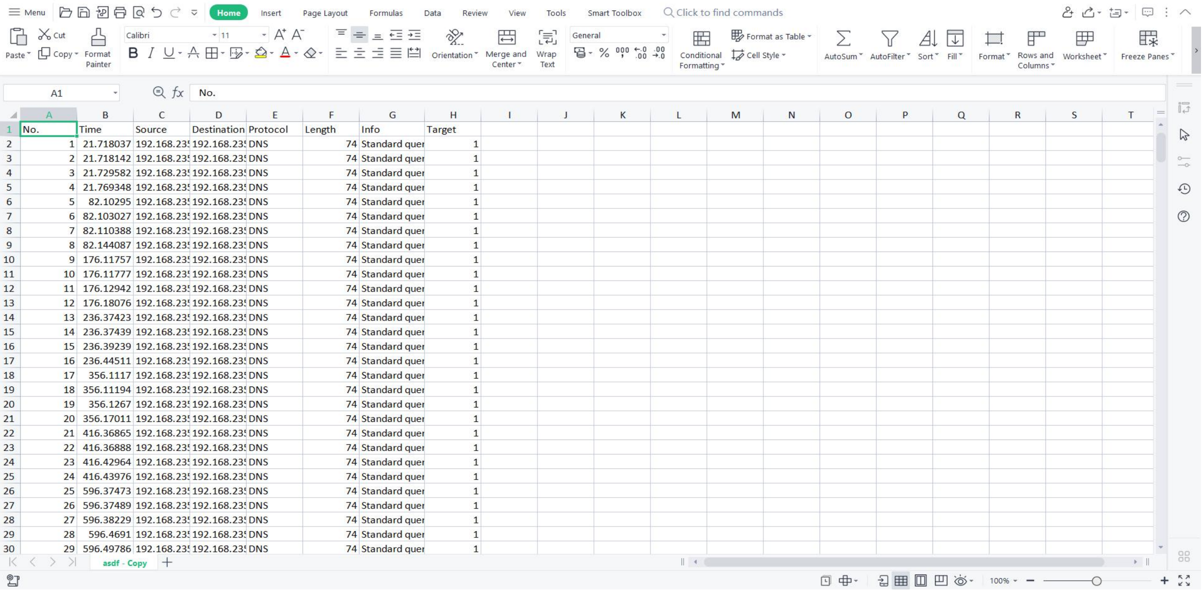
Packets: 89263 · Displayed: 2660 (3.0%)

Profile: Default

ENG IN 11:04 24-04-2023

Fig 14. Capturing packets with Wireshark

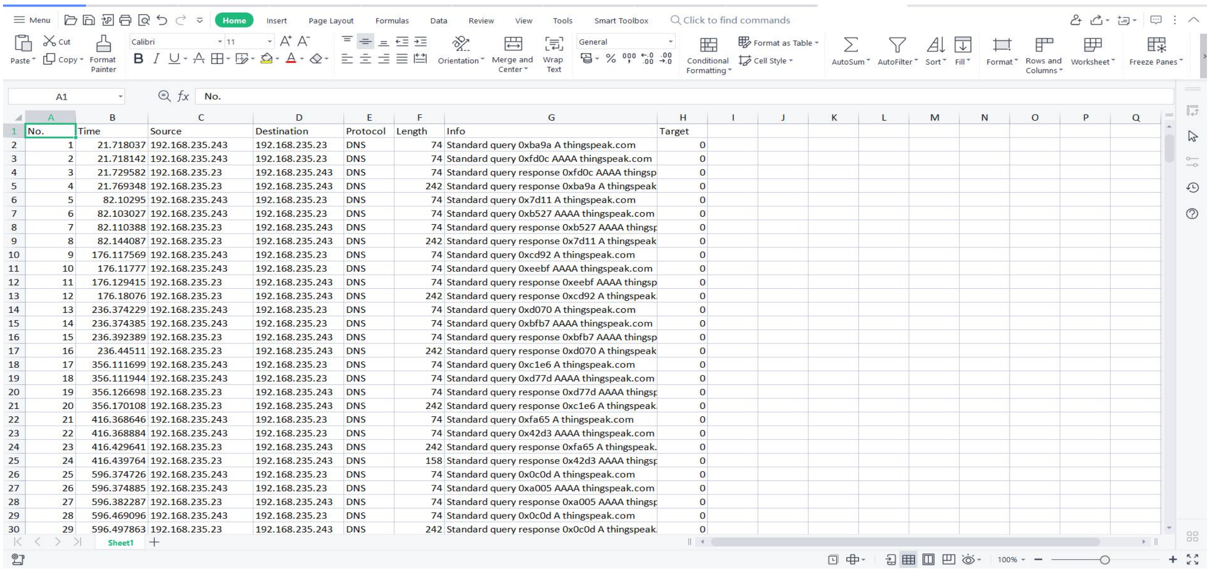
A dataset is a collection of various packets from Wireshark of two classes that are used to train, validate, and test the model. The dataset consists of images of two classes such as normal and attacked traffic with many numbers of packets. The dataset of normal traffic i.e., with delay of 1000ms is collected using Wireshark packets is shown below:



No.	Time	Source	Destination	Protocol	Length	Info	Target
1	21.718037	192.168.235	192.168.235	DNS	74	Standard quer	1
2	21.718142	192.168.235	192.168.235	DNS	74	Standard quer	1
3	21.729582	192.168.235	192.168.235	DNS	74	Standard quer	1
4	21.769348	192.168.235	192.168.235	DNS	74	Standard quer	1
5	82.10295	192.168.235	192.168.235	DNS	74	Standard quer	1
6	82.103027	192.168.235	192.168.235	DNS	74	Standard quer	1
7	82.110388	192.168.235	192.168.235	DNS	74	Standard quer	1
8	82.144087	192.168.235	192.168.235	DNS	74	Standard quer	1
9	176.11757	192.168.235	192.168.235	DNS	74	Standard quer	1
10	176.11777	192.168.235	192.168.235	DNS	74	Standard quer	1
11	176.12942	192.168.235	192.168.235	DNS	74	Standard quer	1
12	176.18076	192.168.235	192.168.235	DNS	74	Standard quer	1
13	236.37423	192.168.235	192.168.235	DNS	74	Standard quer	1
14	236.37439	192.168.235	192.168.235	DNS	74	Standard quer	1
15	236.39239	192.168.235	192.168.235	DNS	74	Standard quer	1
16	236.44511	192.168.235	192.168.235	DNS	74	Standard quer	1
17	356.1117	192.168.235	192.168.235	DNS	74	Standard quer	1
18	356.11194	192.168.235	192.168.235	DNS	74	Standard quer	1
19	356.1267	192.168.235	192.168.235	DNS	74	Standard quer	1
20	356.17011	192.168.235	192.168.235	DNS	74	Standard quer	1
21	416.36865	192.168.235	192.168.235	DNS	74	Standard quer	1
22	416.36888	192.168.235	192.168.235	DNS	74	Standard quer	1
23	416.42964	192.168.235	192.168.235	DNS	74	Standard quer	1
24	416.43976	192.168.235	192.168.235	DNS	74	Standard quer	1
25	596.37473	192.168.235	192.168.235	DNS	74	Standard quer	1
26	596.37489	192.168.235	192.168.235	DNS	74	Standard quer	1
27	596.38229	192.168.235	192.168.235	DNS	74	Standard quer	1
28	596.4691	192.168.235	192.168.235	DNS	74	Standard quer	1
29	596.49786	192.168.235	192.168.235	DNS	74	Standard quer	1

Fig 15. Dataset with normal traffic

The dataset of attacking traffic i.e., with delay of 100ms is collected using Wireshark packets is shown below:



No.	Time	Source	Destination	Protocol	Length	Info	Target
1	21.718037	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xba9a A thingspeak.com	0
2	21.718142	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xfd0c AAAA thingspeak.com	0
3	21.729582	192.168.235.23	192.168.235.243	DNS	74	Standard query response 0xfd0c AAAA thingsp	0
4	21.769348	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0xba9a A thingspeak	0
5	82.10295	192.168.235.243	192.168.235.23	DNS	74	Standard query 0x7d11 A thingspeak.com	0
6	82.103027	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xb527 AAAA thingspeak.com	0
7	82.110388	192.168.235.23	192.168.235.243	DNS	74	Standard query response 0xb527 AAAA thingsp	0
8	82.144087	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0x7d11 A thingspeak	0
9	176.117569	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xcd92 A thingspeak.com	0
10	176.11777	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xeebf AAAA thingspeak.com	0
11	176.129415	192.168.235.23	192.168.235.243	DNS	74	Standard query response 0xeebf AAAA thingsp	0
12	176.18076	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0xcd92 A thingspeak.	0
13	236.374229	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xd070 A thingspeak.com	0
14	236.374385	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xbfb7 AAAA thingspeak.com	0
15	236.392389	192.168.235.23	192.168.235.243	DNS	74	Standard query response 0xbfb7 AAAA thingsp	0
16	236.44511	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0xd070 A thingspeak	0
17	356.111699	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xc1e6 A thingspeak.com	0
18	356.111944	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xd77d AAAA thingspeak.com	0
19	356.126698	192.168.235.23	192.168.235.243	DNS	74	Standard query response 0xd77d AAAA thingsp	0
20	356.170108	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0xc1e6 A thingspeak.	0
21	416.368646	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xfa65 A thingspeak.com	0
22	416.368884	192.168.235.243	192.168.235.23	DNS	74	Standard query 0x42d3 AAAA thingspeak.com	0
23	416.429641	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0xfa65 A thingspeak.	0
24	416.439764	192.168.235.23	192.168.235.243	DNS	158	Standard query response 0x42d3 AAAA thingsp	0
25	596.374726	192.168.235.243	192.168.235.23	DNS	74	Standard query 0x0c0d A thingspeak.com	0
26	596.374885	192.168.235.243	192.168.235.23	DNS	74	Standard query 0xa005 AAAA thingspeak.com	0
27	596.382287	192.168.235.23	192.168.235.243	DNS	74	Standard query response 0xa005 AAAA thingsp	0
28	596.469096	192.168.235.243	192.168.235.23	DNS	74	Standard query 0x0c0d A thingspeak.com	0
29	596.497863	192.168.235.23	192.168.235.243	DNS	242	Standard query response 0x0c0d A thingspeak.	0

Fig 16. Dataset with attacking traffic

3.4 LSTM and Bi-LSTM:

SPLIT DATASET:

A dataset with normal and attacking traffic split into train and test data with 75:25 ratio respectively. Then, the training and validation dataset is split into an 80:20 ratio. Thus, 20% of the data is set for fine-tuning the model after each epoch.

PREPROCESSING DATA:

It is text formatting process before model training, which includes splitting, resizing, normalizing etc.., Data generator in keras allows users to augment the text data to extract various parameters from the original data set. The train and validation datasets are processed and reduced between 0 and 1 by rescaling the text data and then get various valuable features for comparison.

MODEL TRAINING:

The model is trained using RNN which has an input layer, hidden layers, and output layers. The Sequential API model of keras allows the linear stacking of the layers. On input data arrays from Numpy, Keras models are trained.

A LSTM model is proposed for effective training of the model. Similarly, a dropout function is used to randomly drop some neurons connecting the layers to reduce the pressure in the model. The LSTM layers modelling is flattened using a flattening function where it is converted into a single array vector. To train the model on training dataset, four output dense layers with an activation function of sigmoid function and hyperbolic tangent function are used.

- A particular kind of logistical activation function is the sigmoid activation function. It is employed in neural networks' hidden layers to change linear outputs to a quadratic one. It is employed to generate thresholder output in the range of 0 and 1.
- Tanh is a nonlinear function that can be used as an alternative for the sigmoid logistic (0-1) outputting function.. It is used to produce thresholder output between –1 and 1.

CHAPTER 4
RESULT AND DISCUSSION

4.1 COMPILATION

- Importing Libraries and Dataset.
- Now load the data using the pandas data frame.
- Transform raw data into normalized data, then only LSTM model can be built with that dataset.
- Check whether the dataset has been loaded and transformed properly or not by printing the dataset.
- Dividing the data into test and training sets.

- Resize data to perform computations in a fast and accurate manner. After dataset is processed and analysed by defining Loss function, Optimizer, and metrics.

- 2 LSTM layers have to be enough to detect more complex features with tanh activation function.

- 4 output dense layers have be enough to get prominent features with sigmoid activation function. This layer aids in altering the output's dimensionality from the preceding layer it in order that the model may more easily establish the relationship among each value of the data in which the model is working.

- Define LSTM model using TensorFlow API.

- Loss function: Binary cross entropy is used to calculate the loss. It remains unchanged when the estimated probability is scaled and altered. It aids in evaluating the precision of our model in 0s and 1s.
- Adam is used as an optimization tool to rectify and optimize the model. It is used for minimize loss. It improves the accuracy and speed up the deep learning model. It helps the neural network to learn more faster and quickly to minimize the loss function.

- Train the model on training dataset using the above optimizer and loss function.

- Output the accuracy. It is used as a metric which tells the performance of the model.

4.2 PERFORMANCE METRICS:

● Confusion matrix: It is a tabular representation containing true positives, true negatives, false

positives, and false negatives used to describe the performance of a classifier with a variety

of metrics such as accuracy, precision, sensitivity, or recall, F1 score, false positive rate.

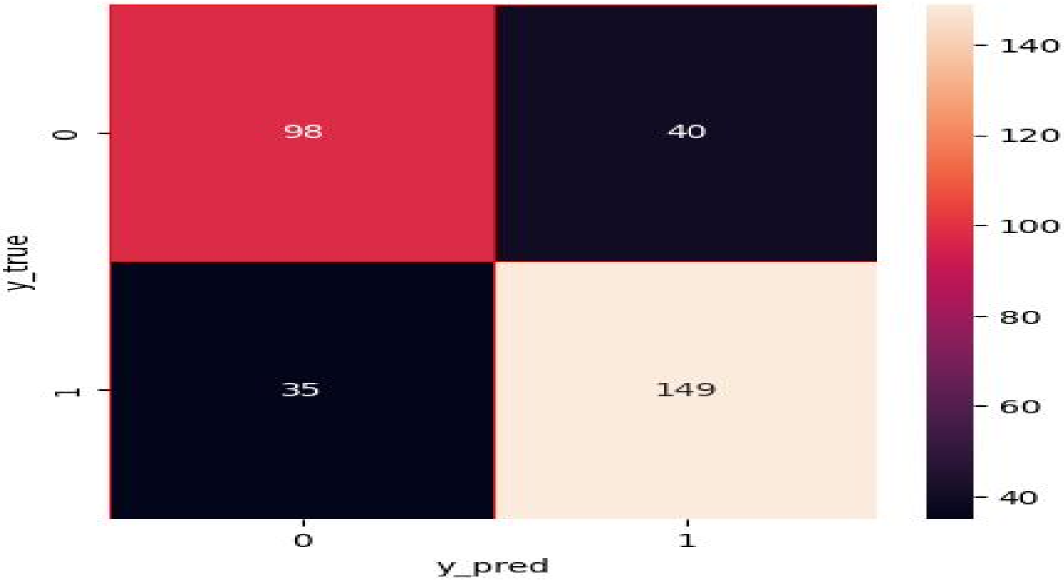


Fig 17. Confusion matrix

● Accuracy: The effectiveness of the classifier for occurrences that are correctly categorised is what is meant by accuracy. The classifier is better when the accuracy is higher.

Accuracy= TP+TN/TP+TN+FP+FN

● Precision: It is the ratio of true positives to the total positives. If the precision is high then

the classifier is a better one.

Precision= TP/TP+FP

● Sensitivity or recall: It is a measure of true positive rate. It is also termed as recall. If the

recall is higher the classifier is better.

Sensitivity= TP/TP+FN

- Specificity: It is a measure of true negative rate. If the value is higher the classifier is better.

Specificity= TN/TN+FP

- False positive rate: It is the probability that a positive result will be given when the true value is negative.

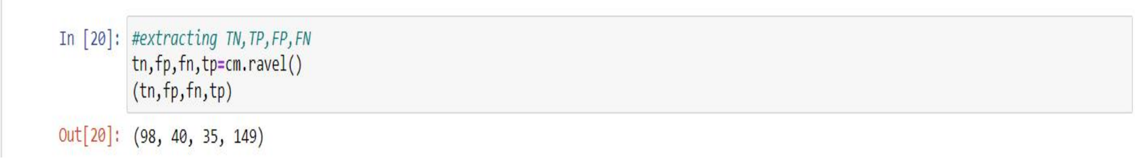
False positive rate=FP/FP+TN

- False negative rate: This is the likelihood that a real positive may slip through the cracks of the test.

False negative rate=FN/FN+TP

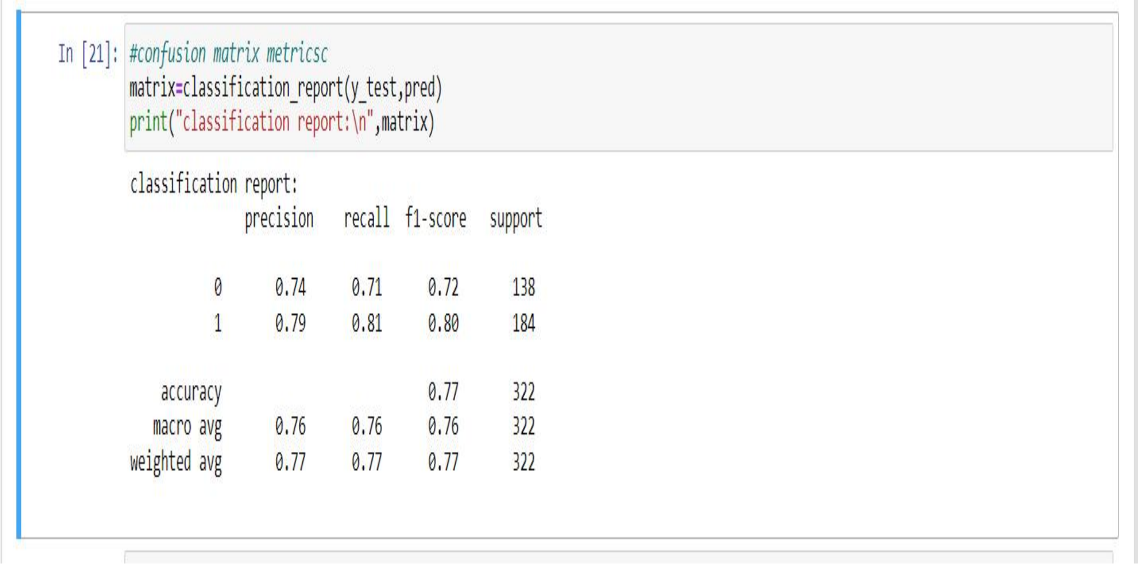
- F1 score: It is weighted average of the recall and the precision. It is used to observe the balance between precision and recall.

F1 score=2*((Precision * Recall)/(Precision + Recall)



```
In [20]: #extracting TN,TP,FP,FN
         tn,fp,fn,tp=cm.ravel()
         (tn,fp,fn,tp)

Out[20]: (98, 40, 35, 149)
```

Fig 18. Extracting matrix value

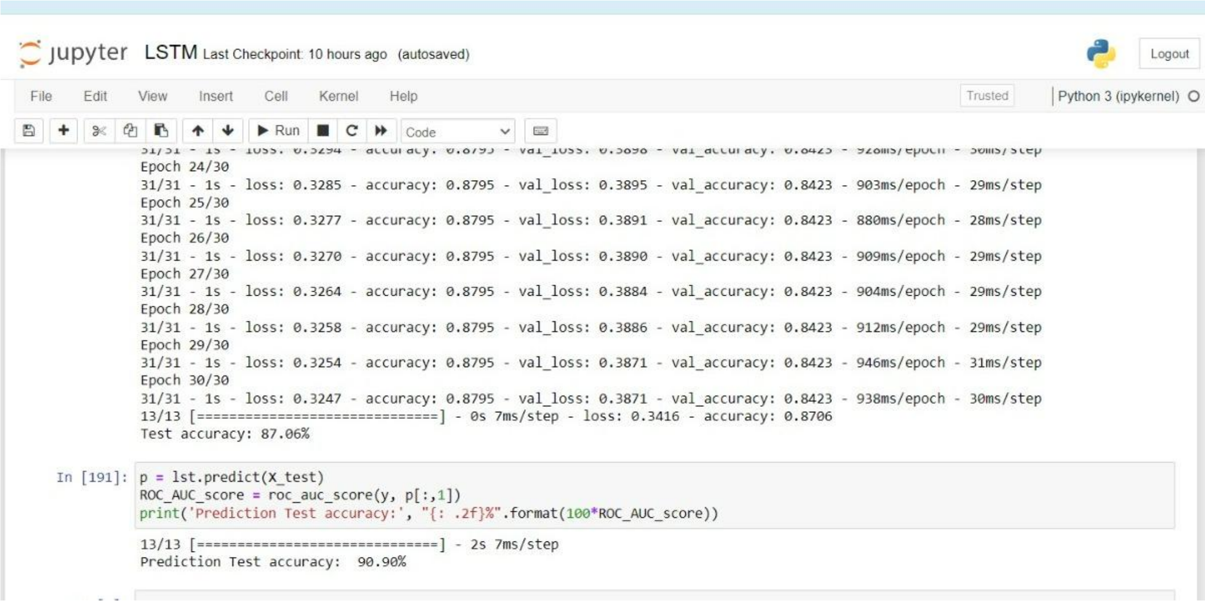


```
In [21]: #confusion matrix metricsc
         matrix=classification_report(y_test,pred)
         print("classification report:\n",matrix)

         classification report:
                       precision    recall  f1-score   support

                    0       0.74      0.71      0.72       138
                    1       0.79      0.81      0.80       184

             accuracy                           0.77       322
            macro avg       0.76      0.76      0.76       322
         weighted avg       0.77      0.77      0.77       322
```

Fig 19. Confusion Matrix Metrics

Finally, we can determine the values of the different metrics in our confusion matrix by utilising the classification_report.

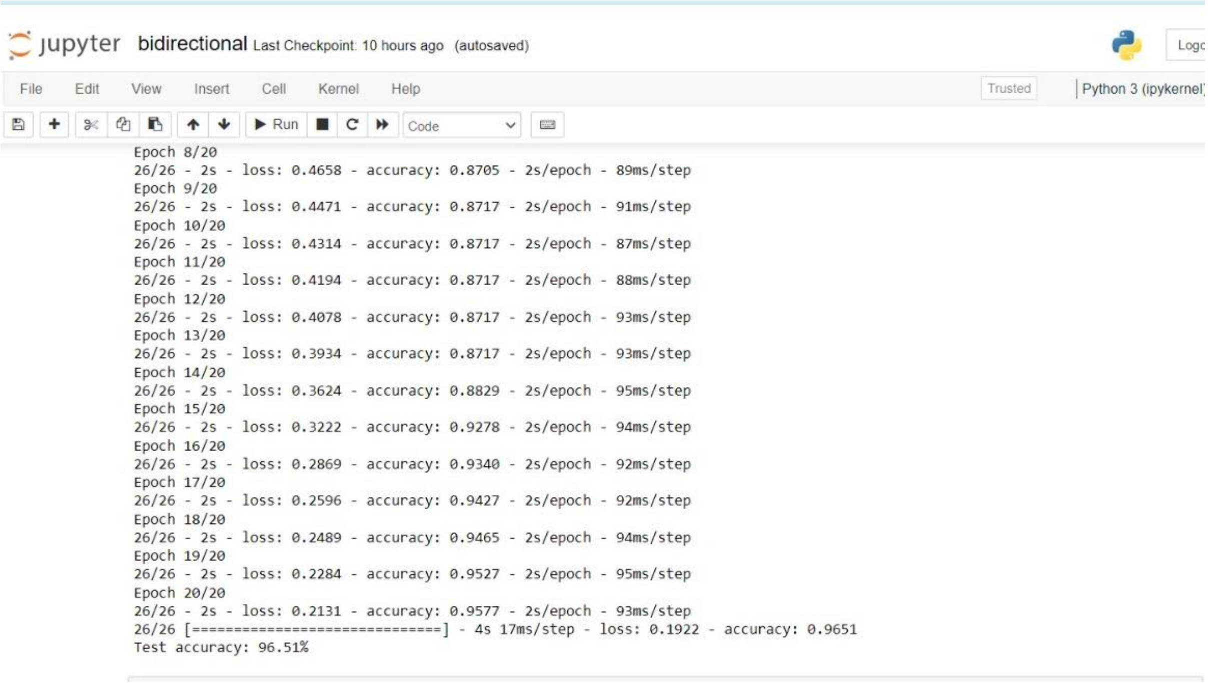
4.3 TEST ACCURACY:


jupyter LSTM Last Checkpoint: 10 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

```
31/31 - 1s - loss: 0.3294 - accuracy: 0.8795 - val_loss: 0.3898 - val_accuracy: 0.8423 - 928ms/epoch - 30ms/step
Epoch 24/30
31/31 - 1s - loss: 0.3285 - accuracy: 0.8795 - val_loss: 0.3895 - val_accuracy: 0.8423 - 903ms/epoch - 29ms/step
Epoch 25/30
31/31 - 1s - loss: 0.3277 - accuracy: 0.8795 - val_loss: 0.3891 - val_accuracy: 0.8423 - 880ms/epoch - 28ms/step
Epoch 26/30
31/31 - 1s - loss: 0.3270 - accuracy: 0.8795 - val_loss: 0.3890 - val_accuracy: 0.8423 - 909ms/epoch - 29ms/step
Epoch 27/30
31/31 - 1s - loss: 0.3264 - accuracy: 0.8795 - val_loss: 0.3884 - val_accuracy: 0.8423 - 904ms/epoch - 29ms/step
Epoch 28/30
31/31 - 1s - loss: 0.3258 - accuracy: 0.8795 - val_loss: 0.3886 - val_accuracy: 0.8423 - 912ms/epoch - 29ms/step
Epoch 29/30
31/31 - 1s - loss: 0.3254 - accuracy: 0.8795 - val_loss: 0.3871 - val_accuracy: 0.8423 - 946ms/epoch - 31ms/step
Epoch 30/30
31/31 - 1s - loss: 0.3247 - accuracy: 0.8795 - val_loss: 0.3871 - val_accuracy: 0.8423 - 938ms/epoch - 30ms/step
13/13 [==============================] - 0s 7ms/step - loss: 0.3416 - accuracy: 0.8706
Test accuracy: 87.06%
```

```
In [191]: p = lst.predict(X_test)
          ROC_AUC_score = roc_auc_score(y, p[:,1])
          print('Prediction Test accuracy:', "{: .2f}%".format(100*ROC_AUC_score))

          13/13 [==============================] - 2s 7ms/step
          Prediction Test accuracy:  90.90%
```

Fig 20. LSTM Accuracy


jupyter bidirectional Last Checkpoint: 10 hours ago (autosaved)
Logo
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel
Run
Code

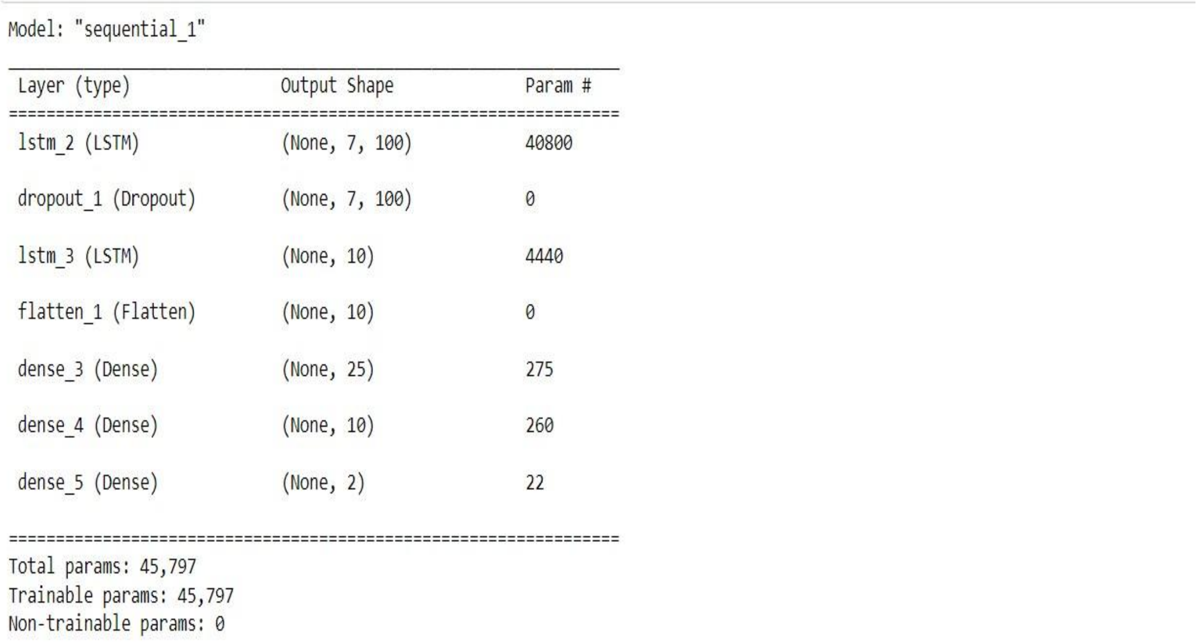
```
Epoch 8/20
26/26 - 2s - loss: 0.4658 - accuracy: 0.8705 - 2s/epoch - 89ms/step
Epoch 9/20
26/26 - 2s - loss: 0.4471 - accuracy: 0.8717 - 2s/epoch - 91ms/step
Epoch 10/20
26/26 - 2s - loss: 0.4314 - accuracy: 0.8717 - 2s/epoch - 87ms/step
Epoch 11/20
26/26 - 2s - loss: 0.4194 - accuracy: 0.8717 - 2s/epoch - 88ms/step
Epoch 12/20
26/26 - 2s - loss: 0.4078 - accuracy: 0.8717 - 2s/epoch - 93ms/step
Epoch 13/20
26/26 - 2s - loss: 0.3934 - accuracy: 0.8717 - 2s/epoch - 93ms/step
Epoch 14/20
26/26 - 2s - loss: 0.3624 - accuracy: 0.8829 - 2s/epoch - 95ms/step
Epoch 15/20
26/26 - 2s - loss: 0.3222 - accuracy: 0.9278 - 2s/epoch - 94ms/step
Epoch 16/20
26/26 - 2s - loss: 0.2869 - accuracy: 0.9340 - 2s/epoch - 92ms/step
Epoch 17/20
26/26 - 2s - loss: 0.2596 - accuracy: 0.9427 - 2s/epoch - 92ms/step
Epoch 18/20
26/26 - 2s - loss: 0.2489 - accuracy: 0.9465 - 2s/epoch - 94ms/step
Epoch 19/20
26/26 - 2s - loss: 0.2284 - accuracy: 0.9527 - 2s/epoch - 95ms/step
Epoch 20/20
26/26 - 2s - loss: 0.2131 - accuracy: 0.9577 - 2s/epoch - 93ms/step
26/26 [==============================] - 4s 17ms/step - loss: 0.1922 - accuracy: 0.9651
Test accuracy: 96.51%
```

Fig 21. Bi-LSTM Accuracy

PARAMETER	LSTM	BIDIRECTIONAL LSTM
LOSS	34.16%	19.22%
ACCURACY	87.06%	96.51%
SPEED	7ms/step	17ms/step

Table 1: LSTM Vs BI-LSTM

4.5 OUTPUTS



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 7, 100)            40800

 dropout_1 (Dropout)         (None, 7, 100)            0

 lstm_3 (LSTM)               (None, 10)                4440

 flatten_1 (Flatten)         (None, 10)                0

 dense_3 (Dense)             (None, 25)                275

 dense_4 (Dense)             (None, 10)                260

 dense_5 (Dense)             (None, 2)                 22

=================================================================
Total params: 45,797
Trainable params: 45,797
Non-trainable params: 0
```

Fig 22. Summary of LSTM model

The algorithm's performance is assessed using an accurate metric that is simple to comprehend. The model parameters are typically used to determine a model's accuracy, which is then calculated as a percentage.

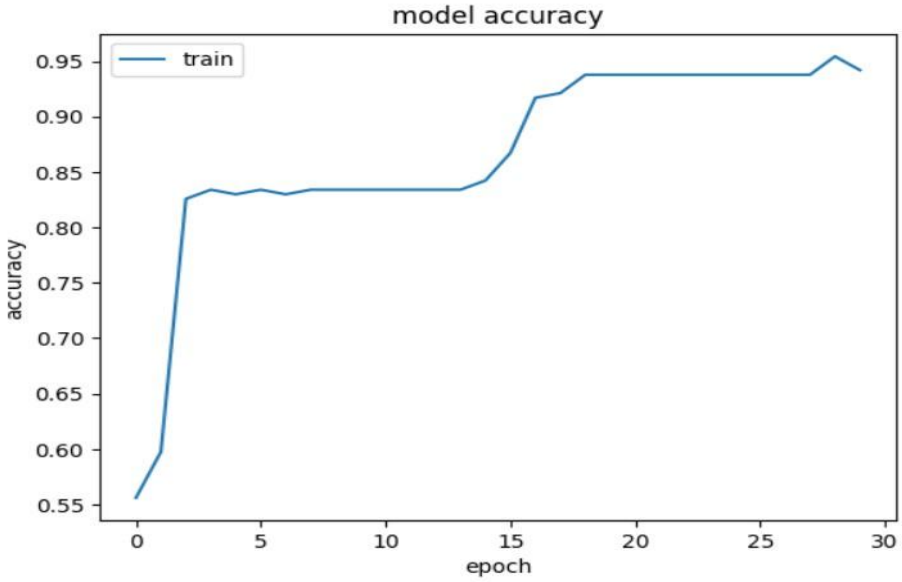


Fig 23. Accuracy graph of LSTM

A model's loss value shows how well or poorly it performs after each optimisation cycle.

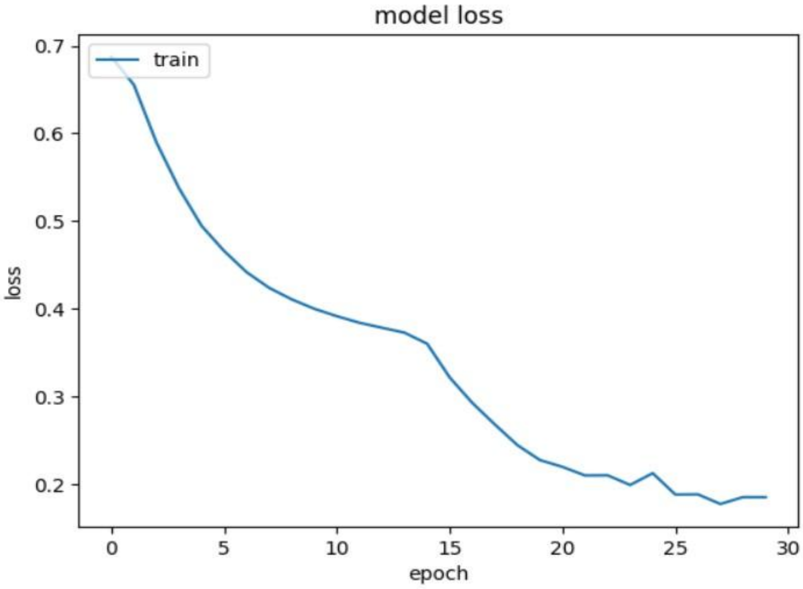


Fig 24. Model loss graph of LSTM

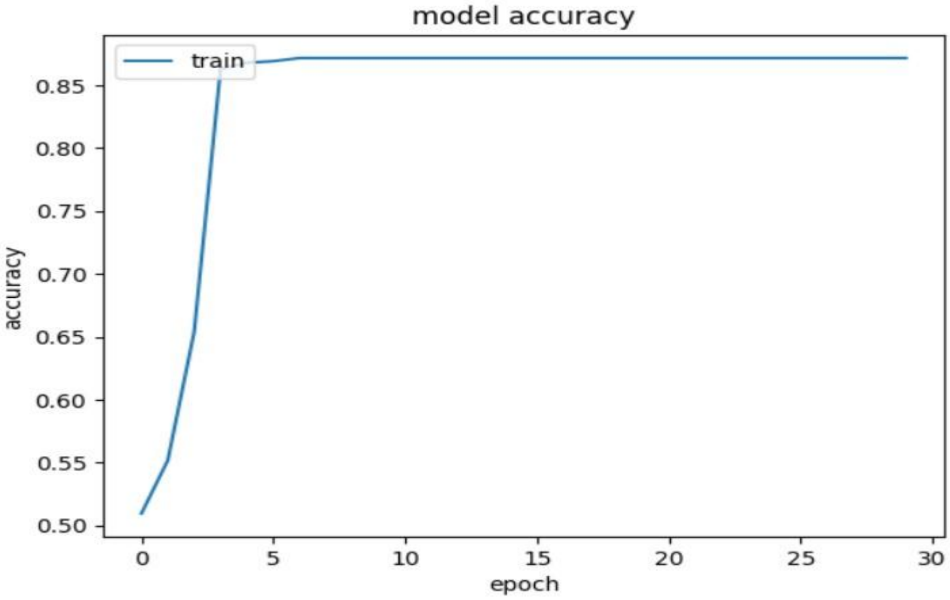


Fig 25. Accuracy graph of Bi-LSTM

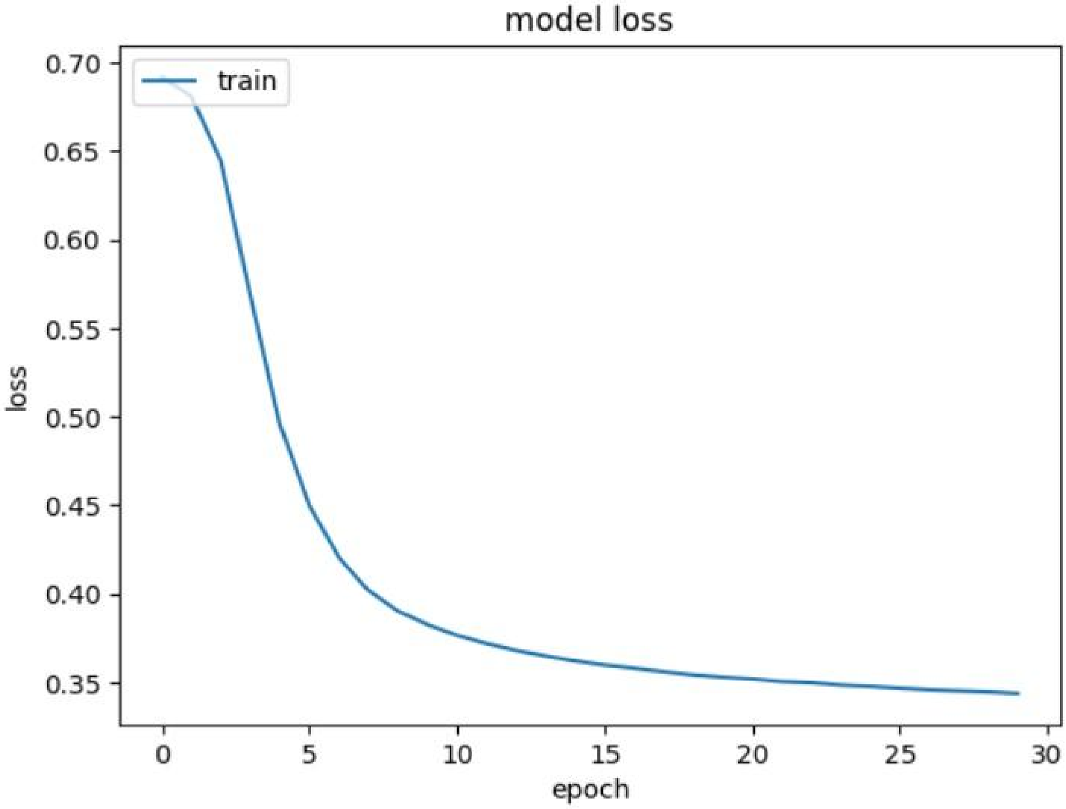


Fig 26. Model loss graph of Bi-LSTM

4.5 ANALYSIS:

- The deep learning algorithm is given with the dataset for replay attack with normal traffic as 1's and attacking traffic as 0's, the outputs are noted down in the table. The accuracy is taken as the main metric to decide the best algorithm because it is dependent on all the parameters of the confusion matrix.
- This study compares and analyses the behaviour of the BiLSTM & LSTM models. The goal is to determine the extent to which extra layers of training data can be used to fine-tune the relevant parameters.
- The findings demonstrate that BiLSTM-based modelling, which is based on additional training of data, provides more accurate predictions than conventional LSTM-based models. More particular, it was found that BiLSTM models outperform LSTM algorithms in terms of accuracy of predictions. Additionally, it was found that BiLSTM models arrive at equilibrium a lot more slowly than LSTM-based models.

CHAPTER 5

CONCLUSIONS AND FURTHER WORK

- Deep Learning Model for intrusion detection is considered in this study.
- Thus, the ZigBee network is created using Digi XBee S2C 802.15.4 Module and the replay attack are simulated on it and the data captured in the Wireshark software tool is taken as dataset for the deep learning algorithms and the results are also tabulated. The deep learning algorithms are trained with these datasets. Further, the dataset is processed and analysed by defining Loss function, Optimizer, and metrics.
- The Bi-LSTM will give more accurate results than LSTM for the replay attack if we use these datasets for training, to test any other datasets.
- In future this work can be extended to real life situations like smart grid to analyse the attacking traffic and we can develop software to defend against these attacks. The obtained experimental results illustrated better with classification performance with hyperparameters.

REFERENCES

1) R. Sriranjani, B. K. M, P. A. K, M. Saleem, N. Hemavathi and A. Parvathy, "Machine Learning Based Intrusion Detection Scheme to Detect Replay Attacks in Smart Grid," *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2023, pp. 1-5, doi: 10.1109/SCEECS57921.2023.10063021.

2)F. Farha, H. Ning, S. Yang, J. Xu, W. Zhang and K. -K. R. Choo, "Timestamp Scheme to Mitigate Replay Attacks in Secure ZigBee Networks," in *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 342-351, 1 Jan. 2022, doi: 10.1109/TMC.2020.3006905.

3) M. S. Wara and Q. Yu, "New Replay Attacks on ZigBee Devices for Internet-of-Things (IoT) Applications," *2020 IEEE International Conference on Embedded Software and Systems (ICESS)*, Shanghai, China, 2020, pp. 1-6, doi: 10.1109/ICESS49830.2020.9301593

4)https://www.tutorialspoint.com/interface-zigbee-with-arduino

5) Setting up Zigbee Communication to Transfer Data Between Arduino and NodeMCU using XBee Modules

6) https://jupyter.org/install

7) https://how2electronics.com/dht11-humidity-temperature-nodemcu-thingspeak

8)https://www.researchgate.net/publication/337293483_Enhanced_Timestamp_Scheme_for_Mitigating_Replay_Attacks_in_Secure_ZigBee_Networks

9) https://www.zendesk.com/in/blog/machine-learning-and-deep-learning/

10) https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning#:~:text=A%20confusion%20matrix%20presents%20a,actual%20values%20of%20a%20classifier.

11). https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56

12). https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

13). https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

14). Xbee/Zigbee Setup with Arduino and NodeMCU - Hackster.io

15). RNN, LSTM, and Bidirectional LSTM: Complete Guide | DagsHub

16). ZIGBEE XBEE S2C–How to configure as Coordinator, Router / End Device - Robolab Technologies Pvt. Ltd.