



KLE Technological
University
Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Minor Project Report

On

Detection DDoS in Private 5G Network

submitted in partial fulfillment of the requirements for the award of the degree
of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Tulip Maurya 01FE22BCS073

Namita Bhat 01FE22BCS200

Avinash Naik 01FE22BCS234

Vaishak Kolhar 01FE22BEC212

Under the guidance of

Dr. Narayan D. G.

School of Computer Science and Engineering



KLE Technological
University
Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

2024-2025

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that project entitled “Detection DDoS in Private 5G Network” is a bonafied work carried out by the student team Tulip Maurya-01fe22bcs073 , Namita Bhat-01fe22bcs200 , Avinash Naik-01fe22bcs204 ,Vaishak Kolhar-01fe22bec212, in partial fulfillment of the completion of 6th semester B. E. course during the year 2024 – 2025. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Guide Name

Dr. Narayan D. G.

Head , SoCSE

Dr. Vijayalakshmi M.

External Viva-Voce

Name of the examiners

Signature with date

1 _____

2 _____

ABSTRACT

The rapid deployment of 5G networks introduces new security challenges, particularly at the edge, where large volumes of data are processed in real time. This project focuses on developing a real-time 5G testbed designed to detect Distributed Denial-of-Service (DDoS) attacks using deep learning (DL) techniques. The testbed integrates key components of a 5G system—User Equipment (UE), Radio Access Network (RAN), and Core Network—using open-source tools including USRP B210 for radio communication, srsRAN for the RAN stack, and Open5GS for the 5G core. By simulating both normal and malicious traffic across the testbed, we collect network flow data at the edge to train DL-based models capable of identifying DDoS patterns with high accuracy. The goal is to enable intelligent, low-latency threat detection directly within the mobile edge computing (MEC) environment. This project not only demonstrates the feasibility of edge-based DDoS mitigation in private 5G networks but also contributes to the broader field of cybersecurity in next-generation mobile systems. The integration of DL and 5G testbed experimentation paves the way for secure and reliable network operations in real-world 5G deployments.

Keywords : *5G Testbed, Private 5G, DDoS Detection, Deep Learnings , Radio Access Network (RAN), Core Network, User Equipment (UE)*

ACKNOWLEDGEMENT

We would like to thank our faculty and management for their professional guidance towards the completion of the mini project work. We take this opportunity to thank Dr. Ashok Shettar, Pro-Chancellor, Dr. P.G Tewari, Vice-Chancellor and Dr. B.S.Anami, Registrar for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Dean of Faculty, SoCSE and Dr. Vijayalakshmi M, Professor and Head, SoCSE for having provided us direction and facilitated for enhancement of skills and academic growth.

We thank our guide Mr. K M M Rajashekharaiiah, Associate professor and SoCSE for the constant guidance during interaction and reviews.

We extend our acknowledgment to the reviewers for critical suggestions and inputs. We also thank Project coordinator Dr. Uday Kulkarni, and reviewers for their suggestions during the course of completion.

Tulip Maurya - 01FE22BCS073

Namita Bhat - 01FE22BCS200

Avinash Naik - 01FE22BCS204

Vaishak Kolhar - 01FE22BEC212

CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENT	i
CONTENTS	iii
LIST OF FIGURES	iv
1 INTRODUCTION	1
1.1 Preamble	1
1.2 Motivation	1
1.3 Objectives of the project	2
1.4 Literature Review	2
1.5 Background Study:	3
1.5.1 Open-5GS	4
1.5.2 5G Architecture	4
1.6 Problem Definition	5
2 PROPOSED SYSTEM	6
2.1 Description of Proposed System	6
2.2 Modules of the Framework	6
2.2.1 Core Network – Open5GS	6
2.2.2 Radio Access Network (RAN) – srsRAN	7
2.2.3 Base Station – USRP B210	7
2.2.4 User Equipment (UE)	7
2.2.5 Traffic Capture – Wireshark	7
2.2.6 Machine Learning Model Script	7
2.3 Description of Target Users	8
2.3.1 Stakeholders	8
2.3.2 Design Principles Identified and Used	8
2.4 Advantages/Applications of Proposed System	9
2.4.1 Advantages	9
2.4.2 Applications	9
2.5 Scope of Proposed System	10

3	SYSTEM DESIGN	11
3.1	System Design	11
3.1.1	UE Registration Process	12
3.1.2	Connectivity and communication	13
3.1.3	DDos detection	14
4	IMPLEMENTATION	16
4.1	Setup	16
4.1.1	Components Required	16
4.1.2	5G Core Setup (Open5GS)	17
4.1.3	RAN Setup (srsRAN Project)	18
4.1.4	Programmable SIM (Open-Cells)	19
4.1.5	USRP B210 Setup	19
4.1.6	Starting the Private 5G Network	20
4.1.7	DDoS Detection Setup	21
5	RESULTS AND DISCUSSIONS	22
5.1	Model Evaluation:	22
5.2	Real-Time Deployment and Testing	23
6	CONCLUSIONS AND FUTURE SCOPE	26
	REFERENCES	27
7	Plagiarism Report	28

LIST OF FIGURES

1.1	5G Architecture	4
3.1	5G testbed setup	11
3.2	UE Registration Process	12
3.3	UE Registration Process	14
4.1	5G testbed setup	16
4.2	registering sim configurations	18
4.3	sim registered	18
4.4	registering sim configurations	20
5.1	Normal Traffic Simulation	24
5.2	Attack Traffic Simulation	24

Chapter 1

INTRODUCTION

This project develops a real-time 5G testbed for detecting Distributed Denial-of-Service (DDoS) attacks using deep learning (DL). A private 5G setup is built using USRP B210 for radio hardware, srsRAN for the RAN, and Open5GS as the core network. The system simulates network traffic between User Equipment (UE) and the core to generate real-time data. DL models are trained and deployed at the interface between RAN and UPF. This setup enables intelligent, edge-based DDoS detection, enhancing the security of next-generation mobile networks.

1.1 Preamble

The rapid evolution of 5G networks introduces unprecedented speed, connectivity, and low-latency services, but also exposes new vulnerabilities, particularly to Distributed Denial-of-Service (DDoS) attacks. As traditional rule-based security mechanisms fall short in real-time threat detection, this project proposes an AI-driven approach to securing private 5G infrastructure. By setting up a complete 5G testbed with RAN, core, and user equipment, we simulate realistic traffic, including DDoS scenarios. Deep learning models are applied for real-time anomaly detection with low-latency inference at the edge. This research aims to enhance the resilience and security of 5G networks in mission-critical environments.

1.2 Motivation

- **Rising DDoS Threats** – Increasingly sophisticated attacks are targeting 5G and edge networks, creating an urgent need for advanced and adaptive security mechanisms.
- **Limitations of Traditional Methods** – Conventional rule-based security systems struggle to analyze high-speed, real-time traffic efficiently, making them inadequate for modern network environments.
- **AI for Better Detection** – Deep Learning (DL) and Large Language Models (LLMs) provide enhanced accuracy, efficiency, and adaptability in detecting and mitigating DDoS attacks.

- **Strengthening 5G Networks** – Improving the security and reliability of 5G infrastructure is critical for supporting next-generation mobile communication and services.

1.3 Objectives of the project

1. To set up a private 5G infrastructure comprising the Radio Access Network (RAN), 5G Core, and User Equipment (UE) using tools like srsRAN, Open5GS, and USRP B210.
2. To simulate both normal and DDoS traffic within the testbed, capture real-time network logs, and perform anomaly detection using deep learning with optimized low-latency inference at the edge.
3. To evaluate and compare the performance of different detection models and deploy the most effective one within the testbed environment.

1.4 Literature Review

Mobile Edge Computing (MEC) has emerged as a critical paradigm in next-generation networks, offering low latency and high bandwidth. However, its distributed nature introduces new security challenges, notably Distributed Denial-of-Service (DDoS) attacks. A lightweight Kubernetes-based mitigation framework was proposed to address this in containerized MEC environments. It leverages Kubernetes' auto-scaling and load balancing to deploy Intrusion Detection and Prevention System (IDPS)-based Containerized Network Functions (CNFs), which dynamically scale in response to attack detection. This approach effectively countered DNS flood and Yo-Yo attacks in a Telco-grade testbed, showcasing efficient resource management and service continuity [1].

Building robust and testable 5G infrastructures is essential for evaluating such security mechanisms. A modular, end-to-end 5G+ testbed was developed to simulate realistic network scenarios, supporting UE registration and full network functionality testing. This testbed integrates analytics and automation to emulate and validate next-generation services, forming a foundation for 6G experimentation [2]. It aligns with our project goals of building a private 5G testbed with MEC and evaluating DDoS mitigation strategies under realistic conditions.

Several researchers have focused on improving DDoS resilience in MEC environments through intelligent traffic analysis. One such approach incorporates AI-driven anomaly detection and traffic redirection to quarantine suspected malicious flows. By leveraging a flow collector and deep packet inspection on separate VMs, this model ensures uninterrupted service delivery, even under attack. MEC orchestration is employed to restore compromised

nodes automatically, showcasing dynamic system resilience [3]. This methodology informs our design choice of using AI/ML models for real-time DDoS detection in private 5G.

Beyond infrastructure, efficient resource allocation during attacks is crucial. CODE4MEC introduces a cooperative framework that deploys containerized security functions dynamically across MEC nodes. It uses a control plane with triggering, scheduling, and coordination modules, and an online combinatorial auction mechanism for real-time resource allocation. Evaluations in testbeds validate CODE4MEC’s adaptability and minimal hardware dependency—making it a strong reference for our system’s distributed mitigation mechanism within Kubernetes-based MEC clusters [4].

Machine learning-based IDS solutions also offer promising results in edge environments. One hybrid IDS model combines Linear Discriminant Analysis (LDA) and Logistic Regression (LR) to detect and isolate intrusions in IoT-edge networks. The model rapidly classifies threats while ensuring minimal disruption to legitimate operations. Isolation mechanisms stealthily quarantine compromised nodes, which aligns with our project’s objective of maintaining MEC integrity under attack [5].

A core enabler for these ML models is high-quality data. The 5G-NIDD dataset addresses this by providing a fully labeled dataset generated on a functional 5G network. It enables effective training and benchmarking of AI-based intrusion detection solutions. Initial analysis confirms the dataset’s suitability for real-world deployment, making it a valuable asset for model evaluation in our project [6].

Lastly, within the Open RAN (O-RAN) context, a multi-layered defense framework was introduced using a dApp at the RAN level and xApps at the near-real-time RIC. These components conduct anomaly detection, UE behavior analysis, and service monitoring. Integrated with firewalls and external dynamic lists, this architecture achieved over 97% accuracy with low latency [7]. This layered approach closely resembles the structure we aim to implement in our private 5G testbed using Kubernetes and MEC with xApps/dApps-like modular defense layers.

Together, these studies provide a comprehensive foundation for our work, guiding the design of an AI-enabled, scalable, and real-time DDoS detection and mitigation system in a private 5G MEC testbed.

1.5 Background Study:

The fifth-generation (5G) wireless technology is a transformative leap in telecommunications, promising to revolutionize the way we connect and communicate. Unlike previous generations, 5G brings lightning-fast internet speeds, ultra-low latency, and massive connectivity, opening doors to innovative applications across industries. With download speeds up

to 100 times faster than 4G, 5G enables seamless streaming, real-time gaming, and immersive experiences. Moreover, it empowers industries like healthcare, autonomous vehicles, and smart cities with real-time data transmission and automation. As 5G networks continue to expand globally, they are poised to redefine how we live, work, and interact in our increasingly digital world.

1.5.1 Open-5GS

Open5GS is an advanced, open-source project designed for building and managing your own NR/LTE mobile network. Whether you're setting up a private network for testing, research, or deployment, Open5GS offers a robust solution for configuring both 5G (NR) and LTE (evolved) networks.

1.5.2 5G Architecture

The 5G architecture consists of three key components: the User Equipment (UE), the Radio Access Network (RAN), and the Core Network (5GC). The UE includes devices like smartphones, IoT sensors, and autonomous vehicles that connect to the network. The RAN manages the wireless link between the UE and the base stations, using advanced technologies such as massive MIMO and beamforming to enhance capacity and coverage. The Core Network is responsible for data routing, user authentication, and managing services with ultra-low latency. Together, these components create a robust and adaptable 5G network that supports a wide range of innovative applications and services.

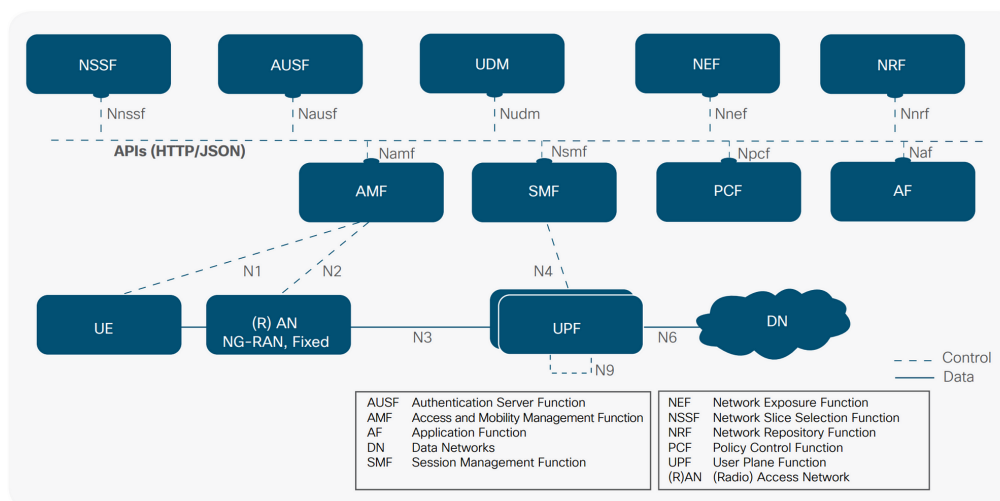


Figure 1.1: 5G Architecture

- AMF (Access and Mobility Management Function) – Handles device registration, connection, and mobility management.

- SMF (Session Management Function) – Manages session establishment, modification, and release, and allocates IP addresses.
- UPF (User Plane Function) – Manages data routing and forwarding between the user equipment and data networks.
- PCF (Policy Control Function) – Enforces policies for network resource usage, QoS, and charging.
- UDM (Unified Data Management) – Stores and manages subscriber data and authentication information.
- AUSF (Authentication Server Function) – Authenticates users and devices trying to connect to the network.
- NSSF (Network Slice Selection Function) – Selects appropriate network slices for different services and users.
- NRF (Network Repository Function) – Maintains a registry of available network functions and their capabilities.
- NEF (Network Exposure Function) – Provides secure APIs for third-party applications to access network services.
- AF (Application Function) – Interacts with the core to request network resources and policies for specific applications.

1.6 Problem Definition

To detect and mitigate DDoS attacks in a private 5G network, a real-time traffic monitoring and anomaly detection system is to be deployed. The system will analyze network traffic, identify malicious patterns, and block suspicious IP addresses to prevent service disruptions and ensure network security.

Chapter 2

PROPOSED SYSTEM

2.1 Description of Proposed System

The proposed system establishes an open-source private 5G testbed for real-time DDoS attack detection and mitigation. The testbed integrates **Open5GS** as the 5G Core, **srsRAN** for the Radio Access Network (RAN), and **USRP B210** as the software-defined radio. Two physical User Equipments (UEs) are connected to simulate realistic mobile network traffic and behavior.

To generate the dataset, **Wireshark** is employed to capture packet-level traffic. Malicious traffic is generated by launching DDoS attacks—**TCP flood**, **ICMP flood**, and **UDP flooding** using **hping**—targeting the 5G Core components, primarily the **AMF** and **UPF**. These packets are labeled as **1 (malicious)**. In parallel, normal user activity traffic is also captured and labeled as **0 (benign)** to ensure balanced dataset representation.

The captured and labeled data is used to train multiple machine learning models, including:

- **Bi-directional LSTM (Bi-LSTM)** for sequence learning,
- **XGBoost** (a gradient-boosted decision tree model),

After model training and validation, the best-performing model is selected based on accuracy and latency. The trained detection model is then deployed at the **N3 interface**—the data plane link between the **User Plane Function (UPF)** and the **gNB (RAN)**—to perform real-time anomaly detection and drop malicious packets proactively with minimal delay.

2.2 Modules of the Framework

The proposed DDoS detection system for private 5G networks comprises the following modules:

2.2.1 Core Network – Open5GS

The 5G Core (5GC) is implemented using **Open5GS**, an open-source core network project that provides standalone 5G functionalities. It includes key network functions such as the Access and Mobility Management Function (AMF), Session Management Function (SMF),

and User Plane Function (UPF), supporting user registration, session establishment, and data transfer. Open5GS is also the primary target for simulating DDoS attacks in this framework.

2.2.2 Radio Access Network (RAN) – srsRAN

The **srsRAN** stack serves as the 5G Radio Access Network (RAN) implementation. It manages radio communication between the gNB and the UEs, allowing for testing and traffic generation. srsRAN connects directly to Open5GS and is essential for establishing reliable wireless communication in the testbed.

2.2.3 Base Station – USRP B210

The **USRP B210** software-defined radio functions as the base station (gNB) hardware, enabling over-the-air signal transmission and reception. It interfaces with srsRAN and the UEs, providing real-time radio communication and supporting the physical layer of the 5G network.

2.2.4 User Equipment (UE)

Two physical or simulated UEs are used to generate realistic traffic and connect to the 5G network. These UEs initiate both benign and malicious network behaviors (such as TCP, ICMP, and UDP floods) to test the detection capability of the system.

2.2.5 Traffic Capture – Wireshark

Wireshark is employed to monitor and capture packet-level traffic on key network interfaces, particularly the **N3 interface** between the UPF and the RAN. During the experiment, traffic is generated by both benign user activity and simulated DDoS attacks (e.g., TCP, ICMP, and UDP floods). Captured traffic is labeled as **malicious (1)** or **benign (0)** to create a supervised dataset for training machine learning models.

Initially, Wireshark extracts five core packet-level attributes from live traffic. This raw packet data is then exported in PCAP format and processed using the **CICFlowMeter** tool, which converts the PCAP files into a structured CSV dataset with **83 network flow features**. From this feature set, the **12 most relevant attributes** are selected through feature selection techniques for efficient training and inference by the ML models.

2.2.6 Machine Learning Model Script

Python-based scripts are used to preprocess network traffic data, extract relevant features, and train multiple machine learning models:

- Bi-directional Long Short-Term Memory (**Bi-LSTM**)
- eXtreme Gradient Boosting(**XGBoost**)

Model performance is evaluated based on accuracy and latency. The best-performing model is deployed at the **N3 interface** to perform real-time DDoS detection with minimal inference delay.

2.3 Description of Target Users

2.3.1 Stakeholders

The target users of this 5G DDoS detection framework include:

1. **Network Operators and Service Providers:** To enhance the security and reliability of private and enterprise 5G networks by detecting and mitigating DDoS attacks in real time.
2. **Cybersecurity Researchers and Developers:** For developing, testing, and evaluating advanced AI-driven security solutions in next-generation mobile networks.
3. **Telecommunication Equipment Vendors:** To integrate intelligent threat detection capabilities into 5G infrastructure products and services.
4. **Academia and Educational Institutions:** As a practical testbed for studying 5G security challenges and training students on real-world network attack detection.

2.3.2 Design Principles Identified and Used

The following key design principles guide the development of the proposed 5G DDoS detection framework:

- **Modularity:** The system is designed as a collection of independent modules (core network, RAN, base station, UEs, traffic capture, ML models) to allow easy maintenance, upgrades, and testing.
- **Real-time Processing:** The framework supports low-latency data capture and inference, enabling immediate detection and mitigation of DDoS attacks at the network edge.
- **Scalability:** The use of open-source, containerized components like Open5GS and srsRAN allows scaling the testbed to multiple UEs and larger network deployments.

- **Open-source Technologies:** Leveraging open-source tools (Open5GS, srsRAN, USRP, Wireshark) ensures cost-effectiveness, transparency, and community support.
- **Data-Driven Security:** The detection relies on supervised machine learning models trained on labeled network traffic, providing adaptive and accurate threat identification.
- **Edge-based Deployment:** Deploying the detection model at the N3 interface (between UPF and RAN) reduces detection latency and network overhead by localizing security functions close to the data source.

These principles collectively ensure that the system meets the requirements of its stakeholders while delivering high performance and ease of use in virtualized environments.

2.4 Advantages/Applications of Proposed System

2.4.1 Advantages

- **Realistic Network Environment:** By using a physical testbed comprising Open5GS, srsRAN, and USRP B210 hardware instead of pure simulation, the system provides a realistic environment that closely mimics real-world 5G network behavior. This allows for more accurate evaluation of DDoS detection models and their practical effectiveness.
- **Low-Latency Detection:** Deploying the detection model at the N3 interface enables near real-time anomaly identification and mitigation, minimizing the impact of attacks on network performance.
- **Scalability:** The modular design with open-source components allows easy extension and adaptation to more UEs, different attack types, and advanced network scenarios.
- **Comprehensive Dataset Generation:** Traffic capture in the testbed includes real protocol stacks and hardware interactions, producing high-quality labeled datasets that improve machine learning model training and generalization.

2.4.2 Applications

- **5G Network Security:** Enhances the security posture of private and public 5G deployments by providing an intelligent mechanism for detecting and mitigating DDoS attacks.
- **Research and Development:** Provides a flexible platform for academia and industry to test, validate, and improve new security algorithms and network functions in a controlled, realistic 5G setup.

- **Operator Training and Testing:** Allows network operators to train personnel and evaluate security policies on a live testbed before deployment in production networks.

2.5 Scope of Proposed System

The proposed system focuses on building a real-time, open-source 5G testbed integrating **Open5GS** (core), **srsRAN** (RAN), **USRP B210** (gNB), and user equipment (UEs) to simulate realistic network traffic scenarios, including Distributed Denial-of-Service (DDoS) attacks.

It aims to capture network traffic using **Wireshark**, label the traffic based on behavior—malicious or benign—and use this labeled dataset to train multiple machine learning models, including **Bi-LSTM**, **XGBoost**, and **Random Forest**. These models are validated on metrics such as accuracy and latency to select the most effective solution. The selected model is deployed at the **N3 interface** (between UPF and RAN) for real-time inference and anomaly detection. The system supports live monitoring, low-latency classification, and intelligent identification of DDoS patterns, demonstrating the feasibility of AI-based security mechanisms in 5G testbeds.

This work enhances research in 5G , providing a scalable and adaptive DDoS detection framework for future deployment in real-world next-generation networks.

Chapter 3

SYSTEM DESIGN

3.1 System Design

In the proposed private 5G setup, user equipment (UE) must first register with the 5G Core (Open5GS) via the Radio Access Network (RAN) managed by srsRAN and USRP B210. The Authentication and Mobility Function (AMF) verifies and registers the UE before establishing session parameters through the Session Management Function (SMF).

Once registration is successful, a communication channel is established, and data transmission begins. Traffic flows from the UE to the RAN (gNB), through the N3 interface to the User Plane Function (UPF), and finally to the internet. This path enables secure and isolated communication within the private 5G environment.

The third step in this project is simulating a DDoS attack using tools like `hping3`, targeting the core. Traffic is captured using Wireshark and collected at the N3 interface. DDoS detection is performed in the user-plane path, making the N3 interface ideal for real-time monitoring and machine learning-based anomaly detection.

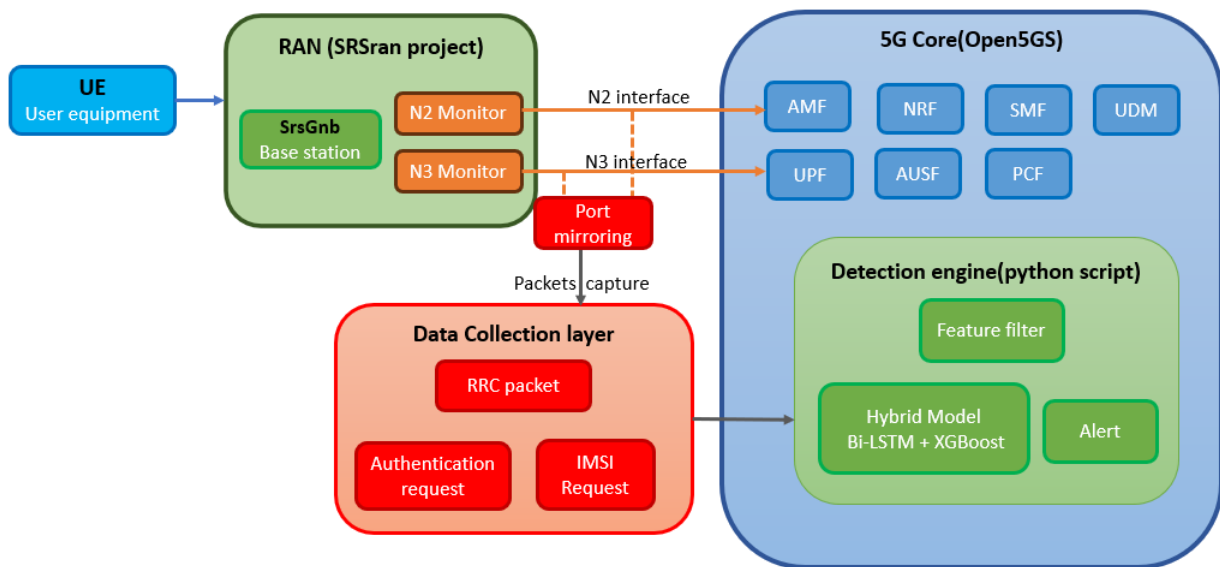


Figure 3.1: 5G testbed setup

3.1.1 UE Registration Process

When a phone with a rewritable SIM card is powered on and attempts to connect to the 5G network, it initiates a registration request. This request is transmitted through the phone's antenna and received by a USRP B210, functioning as the base station. The USRP, working in coordination with **srsRAN**, processes the radio signal within the Radio Access Network (RAN) by tuning to the appropriate frequency and decoding the message.

The decoded signal is then forwarded to the **5G Core Network** (Open5GS), where the **Authentication and Mobility Function (AMF)** validates the request. This involves verifying the user identity based on the SIM card and related credentials. Once authentication is successful, the UE is registered, establishing eligibility to access 5G services such as internet and other network functionalities.

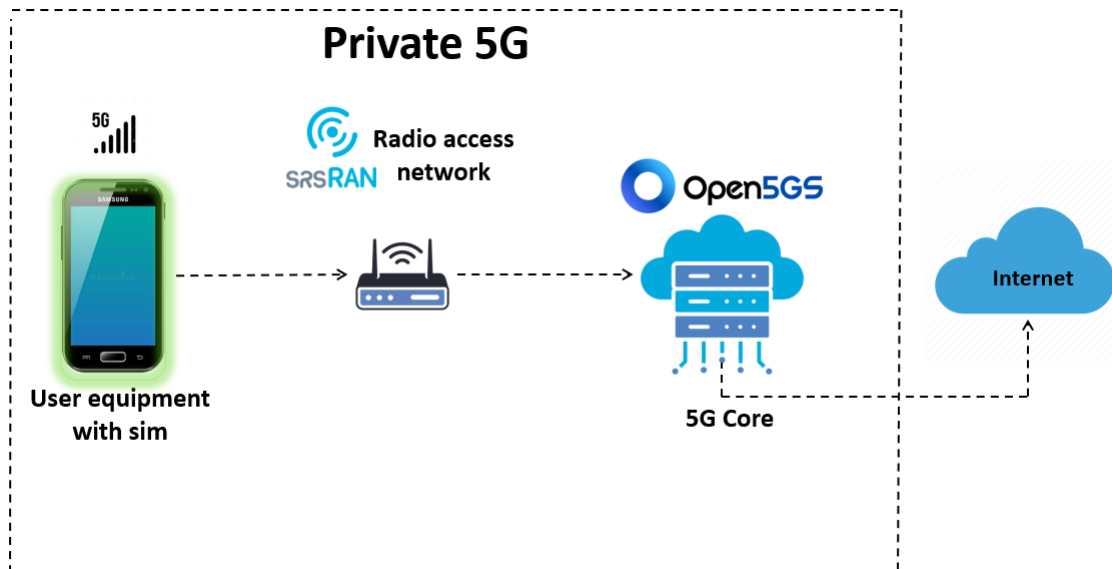


Figure 3.2: UE Registration Process

Algorithm 1 UE Registration Process in Private 5G Network

- 1: **Input:** Powered-on UE with rewritable SIM card
 - 2: UE initiates a registration request
 - 3: UE transmits signal via antenna
 - 4: USRP B210 receives and demodulates the signal
 - 5: Signal is processed by srsRAN within the RAN
 - 6: RAN forwards the decoded signal to Open5GS Core
 - 7: AMF in the core receives the registration request
 - 8: AMF performs authentication using SIM credentials
 - 9: **if** authentication is successful **then**
 - 10: UE is registered to the 5G network
 - 11: Network services (e.g., internet access) enabled for UE
 - 12: **else**
 - 13: Registration fails; UE denied access
 - 14: **end if**
-

3.1.2 Connectivity and communication

Once devices are registered to the private 5G network, they can initiate communication with each other or with external services such as the internet. In this setup, a CCTV camera acts as a data-generating user equipment (UE). It transmits video packets via the USRP B210, which functions as the radio unit in the RAN. The packets are received and processed by the 5G core (Open5GS), which determines the routing path. If the destination is another UE, like a smartphone, the core routes the data to the appropriate USRP serving the receiving UE. This USRP then transmits the data over 5G radio signals, allowing secure, low-latency device-to-device communication.

Algorithm 2 UE-to-UE Communication via Private 5G Network

- 1: **Input:** Registered CCTV camera (UE_1) and smartphone (UE_2)
 - 2: UE_1 captures video stream
 - 3: UE_1 transmits video packets via USRP B210 (RAN)
 - 4: srsRAN processes signal and forwards it to Open5GS Core
 - 5: 5G Core identifies destination as UE_2
 - 6: Core routes packets to corresponding USRP for UE_2
 - 7: USRP transmits packets over 5G signal to UE_2
 - 8: UE_2 receives and decodes video stream
-

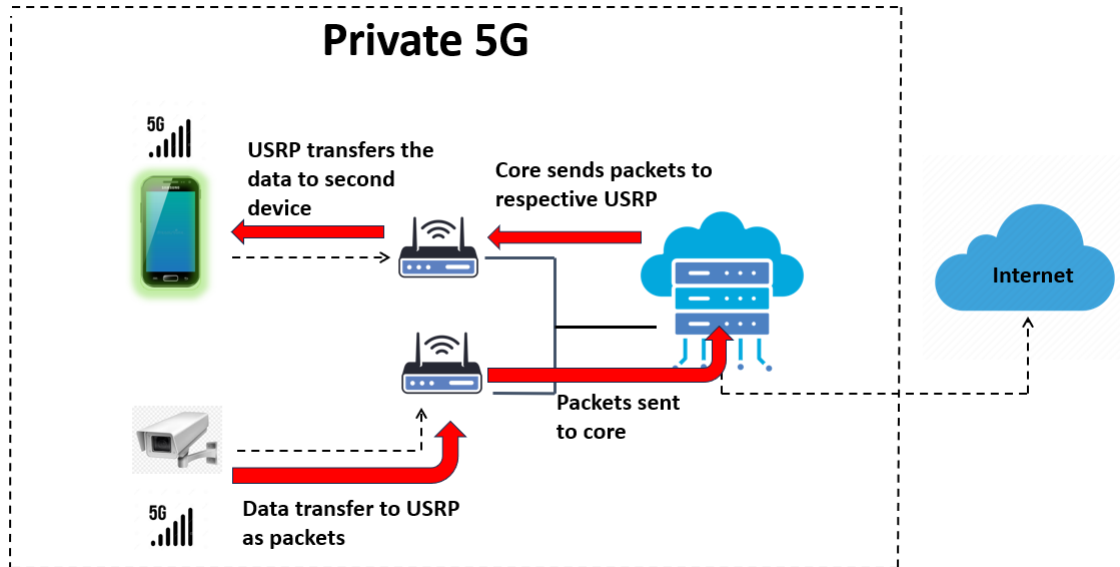


Figure 3.3: UE Registration Process

3.1.3 DDoS detection

To detect DDoS attacks, simulated traffic is generated using **hping** with TCP, UDP, and ICMP flood attacks targeting the core network. Traffic is captured via **Wireshark**, where five key attributes are initially recorded. From these, *FlowMeter* extracts 83 additional flow-based features and converts the capture files from **.pcap** to **.csv** format. The resulting data is fed into a Python-based machine learning detection model. If the traffic is benign, it is allowed to pass; if suspicious, it undergoes further confirmation, and malicious flows are dropped. This detection system is strategically placed at the N3 interface between the UPF and RAN, enabling real-time inspection of user-plane traffic.

Algorithm 3 DDoS Detection Workflow at N3 Interface

- 1: **Input:** Network traffic (normal and attack flows)
 - 2: Generate attack traffic using **hping** (TCP, UDP, ICMP floods)
 - 3: Capture traffic at N3 and N2 interface with Wireshark (5 attributes recorded)
 - 4: Use FlowMeter to extract 83 flow-based features and convert **.pcap** to **.csv**
 - 5: Pass **.csv** data to Python ML detection script
 - 6: **if** traffic classified as benign **then**
 - 7: Allow traffic to pass
 - 8: **else if** traffic classified as suspicious **then**
 - 9: Send traffic for further confirmation
 - 10: **else**
 - 11: Drop malicious traffic
 - 12: **end if**
-

Chapter 4

IMPLEMENTATION

this chapter provides a step-by-step guide to set up a private 5G network using open-source software and hardware. The setup includes a 5G core (Open5GS), Radio Access Network (RAN) with srsRAN, a programmable SIM card, and a USRP B210 software-defined radio (SDR) on Ubuntu 22.04 LTS

4.1 Setup

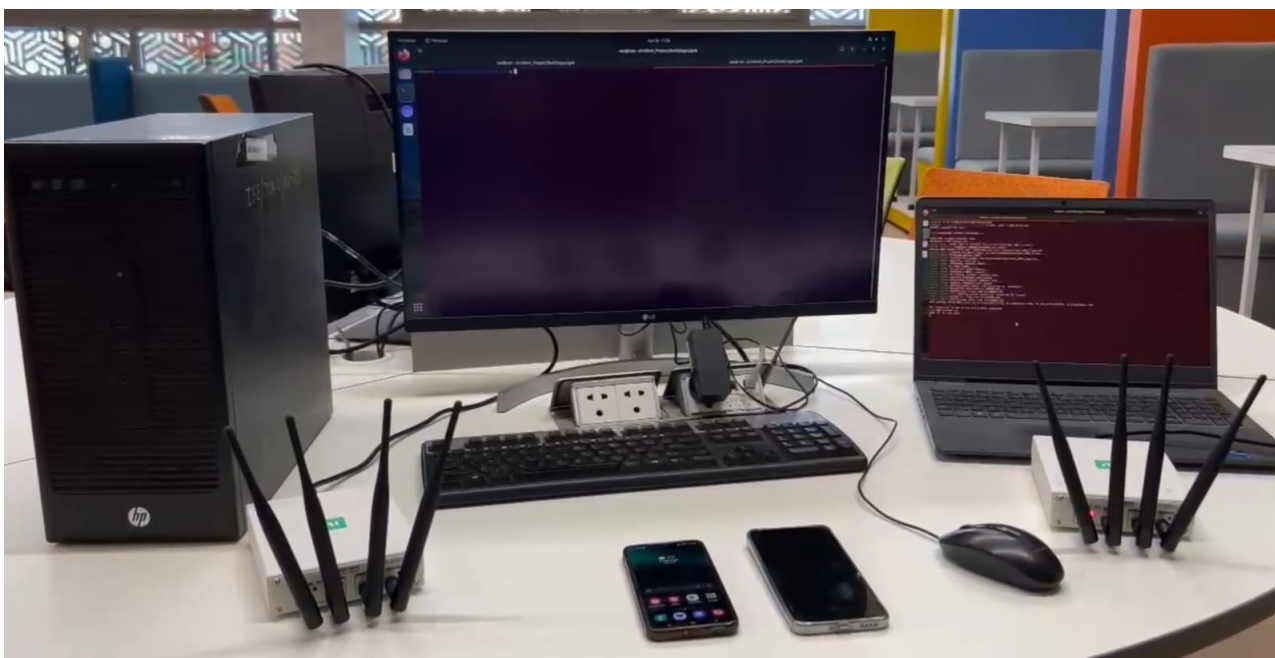


Figure 4.1: 5G testbed setup

4.1.1 Components Required

1. **Hardware:**

- (a) PC with Intel i7 processor, 32 GB RAM, running Ubuntu 22.04 LTS
- (b) USRP B210 Software Defined Radio (connected via USB 3.0)

- (c) Programmable SIM card and USB SIM card reader

2. Software:

- (a) 5G Core: `Open5GS`
- (b) RAN: `srsRAN Project`
- (c) Programmable SIM: `Open-Cells UICC Tools`
- (d) Required Dependencies: `MongoDB`, `Node.js`, `UHD library`

3. Network:

- (a) Stable internet connection for downloading software packages
- (b) Configured network interface for NAT (Network Address Translation) forwarding

4.1.2 5G Core Setup (Open5GS)

1. Prerequisites

- Ubuntu 22.04 LTS installed
- Root or `sudo` access
- Internet connectivity for package installation

2. MongoDB Installation

- Install MongoDB as a database backend required by Open5GS.

3. Open5GS Installation

- Clone the Open5GS repository
- Compile and install core network services (AMF, SMF, UPF, etc.)

4. WebUI Setup

- Launch the Web User Interface to manage subscribers and monitor the core
- Access URL: `http://localhost:9999`
- Username: `admin`, Password: `1423`

5. NAT Port Forwarding

- Enable IP forwarding using `sysctl`
- Configure iptables to set up NAT for connecting the 5G Core to external networks

Open5GS 2.4.3

localhost:3000

Edit Subscriber

Subscriber Configuration

IMSI*

001010000000001

+

Subscriber Key (K)*

00112233445566778899AABBCCDDEEFF

Authentication Management Field (AMF)*

9001

USIM Type

OPc

Operator Key (OPc/OP)*

000102030405060708090A0B0C0D0E0F

UE-AMBR Downlink*

1

Unit

Gbps

UE-AMBR Uplink*

1

Unit

Gbps

CANCEL SAVE

Figure 4.2: registering sim configurations

Open5GS 2.4.3

127.0.0.1:3000

Open5GS

Subscriber

Profile

Account

Search

001010000000001

+

Figure 4.3: sim registered

4.1.3 RAN Setup (srsRAN Project)

1. Prerequisites

- Install required build tools: `cmake`, `gcc`, `g++`, `libboost`, etc.
- Install UHD (USRP Hardware Driver) library for USRP B210 support.

2. Build and Installation

- Clone the srsRAN repository from GitHub.
- Compile the source code using `cmake` and `make`.

- Install the binaries using `make install`.

3. Configuration

- Edit the gNB configuration file: `gnb_rf_b200_tdd_n78_20mhz.yml`
- Set correct RF frequency, bandwidth, and TDD configuration.
- Ensure synchronization with the 5G Core (Open5GS) using appropriate IP and interface settings.

4.1.4 Programmable SIM (Open-Cells)

1. Prerequisites

- A programmable SIM card compatible with 5G authentication.
- USB SIM card reader/writer.
- Downloaded UICC tools package: `uicc-v3.3.tgz`

2. Compilation and Programming

- Extract the UICC tools using `tar -xvzf uicc-v3.3.tgz`
- Navigate to the directory and compile the tools using `make`
- Program the SIM card using the provided scripts (e.g., `python write_sim.py`) by providing IMSI, K, OPc, and other parameters.
- Verify the programmed values with a SIM card reader application.

4.1.5 USRP B210 Setup

1. Installation

- Install the UHD (USRP Hardware Driver) library using `sudo apt install libuhd-dev uhd-host`
- Download and install the firmware and FPGA images using `uhd_images_downloader`

2. Verification

- Connect the USRP B210 to a USB 3.0 port
- Run `uhd_usrp_probe` to verify that the device is detected and working correctly

4.1.6 Starting the Private 5G Network

1. Core Verification and Subscriber Configuration

- Ensure that the Open5GS core is running.
- Open the WebUI at <http://localhost:9999> and log in using the admin credentials.
- Add a new subscriber with IMSI and authentication keys matching the programmable SIM card.

2. USRP Verification

- Connect the USRP B210 via USB 3.0.
- Run `uhd_usrp_probe` to confirm device recognition.

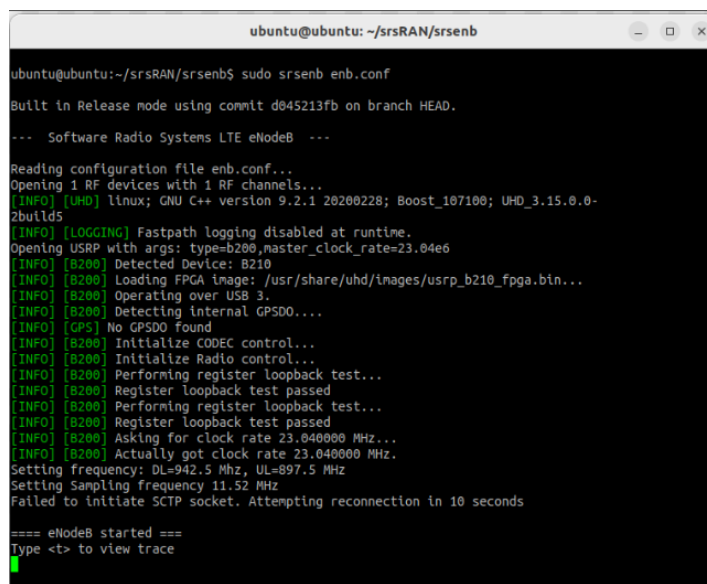
3. RAN Configuration and Execution

- Navigate to the gNB directory:

```
cd srsRAN_Project/build/apps/gnb/
```

- Start the gNB using the configuration file:

```
sudo ./gnb -c gnb_rf_b200_tdd_n78_20mhz.yml
```



```

ubuntu@ubuntu: ~/srsRAN/srsenb
ubuntu@ubuntu:~/srsRAN/srsenb$ sudo srsenb enb.conf
Built in Release mode using commit d045213fb on branch HEAD.

--- Software Radio Systems LTE eNodeB ---

Reading configuration file enb.conf...
Opening 1 RF devices with 1 RF channels...
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200228; Boost_107100; UHD_3.15.0.0-2build5
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP with args: type=b200, master_clock_rate=23.04e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Loading FPGA image: /usr/share/uhd/images/usrp_b210_fpga.bin...
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Detecting internal GPSDO....
[INFO] [GPS] No GPSDO found
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Setting frequency: DL=942.5 MHz, UL=897.5 MHz
Setting Sampling frequency 11.52 MHz
Failed to initiate SCTP socket. Attempting reconnection in 10 seconds

==== eNodeB started ====
Type <t> to view trace

```

Figure 4.4: registering sim configurations

4.1.7 DDoS Detection Setup

1. Traffic Generation and Capture

- Launch DDoS attacks on the 5G Core Server IP using **hping3** with TCP flood, UDP flood, and ICMP flood.
- Capture network traffic using **Wireshark** at the **N3 interface** (between UPF and RAN).
- Save the captured packets in **.pcap** format.

2. Feature Extraction with FlowMeter

- Convert **.pcap** files to **.csv** using **CICFlowMeter**.
- Extract 83 flow-based features from 12 core attributes through feature selection.

3. Model Training and Evaluation

- Train multiple ML models (Bi-LSTM, Random Forest, XGBoost) on the extracted feature dataset.
- Evaluate performance using accuracy, precision, recall, and F1-score.
- Select the best-performing model for deployment.

4. Real-time Detection and Action

- Deploy the selected model in a Python script.
- Classify incoming traffic as **benign**, **suspicious**, or **malicious**.
- Allow benign traffic; flag suspicious traffic; drop malicious packets.

Chapter 5

RESULTS AND DISCUSSIONS

In this section, we present the evaluation of our proposed DDoS detection framework implemented within a private 5G network setup. The system leverages BiLSTM for temporal behavior modeling and XGBoost for feature-based anomaly classification. The test environment was built using Open5GS as the 5G core network and srsRAN for the radio access network, enabling realistic simulation of both normal and attack traffic scenarios. AMF logs were used as the primary data source, capturing signaling interactions between UEs and the core. Our goal was to measure the accuracy, latency, and overall effectiveness of the hybrid detection model in identifying DDoS patterns without disrupting normal network performance. The results are discussed in terms of detection performance, resource overhead, and system responsiveness under various load conditions. In this section, we present the evaluation of our proposed DDoS detection framework implemented within a private 5G network setup. The system leverages BiLSTM for temporal behavior modeling and XGBoost for feature-based anomaly classification. The test environment was built using Open5GS as the 5G core network and srsRAN for the radio access network, enabling realistic simulation of both normal and attack traffic scenarios. AMF logs were used as the primary data source, capturing signaling interactions between UEs and the core. Our goal was to measure the accuracy, latency, and overall effectiveness of the hybrid detection model in identifying DDoS patterns without disrupting normal network performance. The results are discussed in terms of detection performance, resource overhead, and system responsiveness under various load conditions.

5.1 Model Evaluation:

To evaluate the detection capabilities of our system, we trained both BiLSTM and XGBoost models using real-time data captured from the AMF logs during controlled DDoS attack simulations and normal user activity.

BiLSTM Performance: The BiLSTM model was selected due to its ability to learn temporal dependencies and sequential patterns in the network signaling behavior, which are critical for identifying gradually intensifying DDoS attacks. It effectively captured fluctuations in UE registration, NAS message bursts, and session setup anomalies.

- Accuracy: 96.72%
- Precision: 95.44%
- Recall: 98.15%

XGBoost Performance: In contrast, the XGBoost model was trained using aggregated, windowed features derived from the same AMF logs and UPF logs but without maintaining strict temporal order. It was particularly effective in identifying sudden spikes in activity such as abrupt floods of registration or session requests that are typical in short-burst volumetric attacks.

- Accuracy: 74.72%
- Precision: 70.44%
- Recall: 72.15%

While not as sensitive as BiLSTM in detecting slowly growing attacks, XGBoost's lightweight nature makes it suitable for quick response scenarios and edge deployments.

5.2 Real-Time Deployment and Testing

Both the BiLSTM and XGBoost models were deployed in real-time within the private 5G testbed environment. The detection pipeline was integrated with the AMF component of the Open5GS core, allowing continuous monitoring and inference on live signaling traffic. During testing, the system was subjected to both legitimate user behavior and controlled DDoS attack scenarios. The models processed the incoming data streams in real-time, generating predictions on whether each sequence or feature window indicated normal or malicious activity. This real-time validation demonstrated the practical feasibility of using machine learning-based detection directly within the 5G core, enabling immediate identification of anomalies without requiring post-processing or offline analysis.

The BiLSTM model continuously evaluated sequences of message activity to detect evolving attack signatures, while XGBoost provided immediate feedback on sudden traffic anomalies over short time windows. Both models operated with minimal latency, with average inference times under 50 ms for BiLSTM and under 10 ms for XGBoost. Alerts were triggered upon

```

[15:05:38] Src: 127.0.0.1:41446 -> Dst: 127.0.0.5:38412 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:38] Src: 127.0.0.1:41446 -> Dst: 127.0.0.5:38412 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:38] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:38] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:38] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:38] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:05:39] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:41446 | XGBoost: Benign | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)

```

Figure 5.1: Normal Traffic Simulation

```

BiLSTM model loaded successfully.
XGBoost model loaded successfully.
Starting real-time traffic capture... Press Ctrl+C to stop.
[15:09:15] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: N/A
[15:09:15] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: N/A
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:15] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:15] Src: 127.0.0.5:38412 -> Dst: 127.0.0.1:59281 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: DDoS | BiLSTM: Benign
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 127.0.0.1:59281 -> Dst: 127.0.0.5:38412 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)
[15:09:16] Src: 10.45.0.1:2152 -> Dst: 10.45.0.1:2152 | XGBoost: DDoS | BiLSTM: DDoS
BiLSTM input shapes: protocol=(1, 10, 1), numerical=(1, 10, 10)

```

Figure 5.2: Attack Traffic Simulation

detection, and could be configured to initiate rate-limiting, block specific UE IPs, or notify administrators in a production setup.

This deployment validated the models' ability to function in a live 5G environment, maintaining detection performance without introducing noticeable delay or service degradation. It also demonstrated the potential for machine learning-based DDoS protection mechanisms to be embedded directly within mobile core networks, offering proactive threat mitigation in emerging private and enterprise 5G infrastructures.

Chapter 6

CONCLUSIONS AND FUTURE SCOPE

This work presents a real-time DDoS detection framework tailored for private 5G networks, leveraging the strengths of both sequential and non-sequential machine learning models. By integrating BiLSTM and XGBoost into the 5G core’s monitoring pipeline, we demonstrated the system’s ability to accurately detect both slowly evolving and sudden volumetric attacks using AMF signaling data. The BiLSTM model exhibited high detection accuracy due to its ability to capture temporal dependencies in network behavior, while XGBoost provided lightweight and responsive detection of abrupt anomalies. Real-time deployment and testing in a controlled 5G testbed showed that the proposed models could operate effectively with low latency and minimal resource overhead, making the solution viable for integration into operational private 5G deployments.

While the current system focuses on signaling-level anomalies using AMF and UPF logs, future extensions could include multi-protocol data fusion, incorporating SMF or gNodeB logs for a more holistic detection approach. Additionally, adapting the detection pipeline to support federated learning or edge-distributed inference would enable broader scalability across heterogeneous 5G environments without centralized data collection. Integration with automated mitigation tools—such as dynamic policy control, firewall rule updates, or intelligent traffic rerouting—could further enhance the system’s capability to not only detect but also respond to threats in real time. Exploring unsupervised or self-supervised approaches may also allow the framework to detect zero-day or previously unseen attack variants without requiring extensive labeled datasets.

REFERENCES

- [1] M. A. Ferrag and L. Maglaras. Flexible and lightweight mitigation framework for distributed denial-of-service attacks in container-based edge networks using kubernetes. *IEEE Transactions on Network and Service Management*, 2024.
- [2] Ali Chouman, Dimitrios Michael Manias, and Abdallah Shami. A modular, end-to-end next-generation network testbed: Towards a fully automated network management platform. *arXiv preprint*, 2024.
- [3] Ahmed H. B. M. Sayed and Yang Xiang. Improved security solutions for ddos mitigation in 5g mobile edge computing. *arXiv preprint*, 2021.
- [4] Zhiwei Li, Hui Li, Yaqiang Zhang, and Mugen Peng. A cooperative defense framework against application-level ddos attacks on mec services (code4mec). *IEEE Internet of Things Journal*, 8(11):8677–8691, 2021.
- [5] R. Nadarajan, R. Banu, and R. Kumar. A secure mobile edge computing model using machine learning and ids to detect and isolate intruders. *Future Generation Computer Systems*, 151:170–180, 2024.
- [6] M. F. Razzak, H. Abbas, W. Gharibi, and M. Imran. 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network. *arXiv preprint*, 2022.
- [7] Md Monzur Hossain, Gautam Srivastava, and Mohd Faizal Zolkipli. Multiaccess edge computing-based simulation as a service for 5g mobile applications. *Security and Privacy*, 3(3):e86, 2020.

Chapter 7

Plagiarism Report

Attach your plagiarism report of this mini report here. Make sure that plagiarism is below 20 %.