

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The advent of 5G technology has ushered in a new era of high-speed, low-latency communication, enabling widespread connectivity for IoT devices, mission-critical services, and large-scale networks. However, the increased complexity and diversity within 5G infrastructure also introduce greater exposure to security threats, including unauthorized access, data breaches, and denial-of-service (DoS) attacks. As 5G becomes an essential foundation for smart cities, autonomous systems, and other critical applications, ensuring its security and stability is of paramount importance.

Conventional security solutions, which typically rely on static rules and fixed configurations, may fall short in adapting to rapidly evolving threats within dynamic 5G ecosystems. This highlights the need for a more adaptive and intelligent security framework capable of real-time monitoring and anomaly detection to prevent potential breaches before they occur.

This project proposes a hybrid solution that integrates NS3-mmWave simulation with machine learning techniques to develop an effective intrusion detection system (IDS) tailored for 5G environments. The NS3-mmWave module replicates a realistic 5G network scenario, including base stations (eNBs), user equipment (UEs), and a remote host, all connected via millimeter wave (mmWave) links. Network traffic is monitored using FlowMonitor, which generates XML files containing critical performance indicators such as delay, jitter, packet loss ratio, and throughput.

These XML outputs are converted into CSV files for easier processing and feature extraction in Python. Machine learning algorithms are then trained on this data to classify traffic behavior as normal or suspicious based on learned

patterns. Detected anomalies are visualized using graphical representations, aiding in network assessment and interpretation.

By identifying unusual traffic activity early, this system supports proactive threat detection and timely mitigation. The combination of realistic network simulation and intelligent analysis offers a scalable, efficient, and flexible solution to enhance the security posture of future 5G networks.

1.2 OBJECTIVE

- To Develop a realistic 5G network scenario using the NS3-mmWave simulator, incorporating base stations (eNBs), user devices (UEs), and a centralized remote host for data communication.
- To Gather comprehensive network performance metrics—such as throughput, latency, and packet loss—through FlowMonitor during simulation runs.
- To Transform the XML output from FlowMonitor into CSV format to facilitate feature extraction and data analysis.
- To Build and assess a machine learning model capable of identifying deviations from normal traffic behavior, effectively distinguishing between regular and malicious activity.
- Using Python-based visualization tools to graphically represent detected anomalies and overall network performance for enhanced clarity and interpretation.
- To Design a scalable, adaptive, and real-time intrusion detection system tailored to the dynamic nature of 5G environments.
- To Strengthen the cybersecurity framework of 5G networks by enabling proactive threat detection and timely response to potential security incidents.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW WITH THIS PROJECT

[1]

- **TITLE:** Machine Learning based Anomaly Detection for 5G Networks
- **AUTHOR:** Jordan Lam, Robert Abbas
- **JOURNAL:** arXiv, 2020, arXiv:2003.03474v1, <https://ieeexplore.ieee.org/document/10724005/>
- **METHODOLOGY:**

This paper proposes a novel software-defined security (SDS) system that utilizes machine learning to detect anomalous network traffic in 5G environments. A Convolutional Neural Network (CNN), optimized through Neural Architecture Search (NAS), is trained on network flow data transformed into image form. The model classifies traffic as either benign or anomalous, achieving 100% accuracy for benign traffic and 96.4% for anomalous traffic. The CICIDS2018 dataset is used, containing labeled flow-based features derived from a realistic simulation environment. Traffic flows are processed into 100x100x3 images and analyzed using Google AutoML Vision. The SDS architecture captures traffic from backhaul and core network links without affecting performance, enabling real-time detection and policy updates. The approach shows promise for dynamic, end-to-end protection of 5G networks, especially in encrypted traffic scenarios where traditional deep packet inspection fails. The study emphasizes automated feature extraction, lightweight deployment on edge devices, and real-time IDS integration.

[2]

- **TITLE:** Network Anomaly Detection in 5G Networks
- **AUTHOR:** Atta-ur Rahman, Maqsood Mahmud, Tahir Iqbal, Linah Saraireh, et al.

- **JOURNAL:** Mathematical Modelling of Engineering Problems, Volume 9, Issue 2, April 2022, Pages 397–404,
<https://www.researchgate.net/publication/360250788>
- **METHODOLOGY:**
This paper explores various machine learning and behavior-based approaches for Network Anomaly Detection (NAD) in 5G networks. It discusses six core methodologies including pattern-based random walk, immune network with density peak clustering, hybrid data optimization using Isolation Forest and Random Forest, multi-view ensemble learning, and user intention-based traffic dependency analysis using CR-Miner. The integrated approach combining KNN, K-prototype clustering, and refinement mechanisms was found most effective in detecting anomalies from large system logs. CR-Miner enables detection based on user behavior and traffic dependencies without prior knowledge. The system also utilizes Bayesian State Space Models for time-series forecasting of cyber events, offering predictive insights for future attacks. The paper emphasizes that traditional IDS systems fall short in modern 5G environments, advocating instead for a hybrid anomaly-based detection system that integrates real-time monitoring, predictive modeling, and behavioral analysis.

[3]

- **TITLE:** Federated Learning Based Anomaly Detection as an Enabler for Securing Network and Service Management Automation in Beyond 5G Networks
- **AUTHOR:** Suwani Jayasinghe, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Mika Ylianttila
- **JOURNAL:** EuCNC & 6G Summit, IEEE, June 2022
<https://www.researchgate.net/publication/359685649>
- **METHODOLOGY:**
This paper introduces a hierarchical anomaly detection mechanism based on Federated Learning (FL) to enhance the security of Zero-touch network and Service Management (ZSM) in Beyond 5G networks. It presents a two-

stage FL framework using the UNSW-NB15 dataset. Stage 1 consists of lightweight models for edge-level anomaly detection, while Stage 2 contains more complex models for core-level filtering. FL enables local model training with data privacy preserved, and a central server aggregates the parameters to update the global model. TensorFlow Federated was used for simulation. Accuracy peaked at 94% when training data included 60% anomalous samples. The multi-stage detection boosts overall accuracy and allows real-time detection and adaptation within the ZSM architecture, making it suitable for future 6G network security frameworks.

[4]

- **TITLE:** Machine Learning-Enabled Techniques for Anomaly Detection in 5G Networks
- **AUTHOR:** Priyanka Dass, Anjali Rajak, Rakesh Tripathi]
- **JOURNAL:** IEEE, 15th ICCCNT Conference, June 2024, IIT Mandi, India
[IEEE Xplore DOI: 10.1109/ICCCNT61001.2024.10724005](https://doi.org/10.1109/ICCCNT61001.2024.10724005)
- **METHODOLOGY:**
This study presents a hybrid feature selection-based machine learning model for anomaly detection in 5G networks using the 5G-SliciNdd dataset. A two-step hybrid feature selection technique combining Correlation Coefficient (CC) filtering and Binary Particle Swarm Optimization (BPSO) was implemented to reduce dimensionality and enhance model performance. Several ML models, including Decision Tree, kNN, ANN, Naïve Bayes, XGBoost, and Hard Voting, were trained and evaluated using accuracy, F1-score, recall, precision, and false alarm rate (FAR). The Hard Voting model achieved the highest accuracy of 99.8% with the lowest FAR of 0.0014. The study demonstrates the efficacy of combining ensemble learning with optimized feature selection for real-time, low-latency anomaly detection in 5G networks.

[5]

- **TITLE:** Machine Learning Applied to Anomaly Detection on 5G O-RAN Architecture
- **AUTHOR:** Pedro V. A. Alves, Mateus A. S. S. Goldberg, Wysterlânia K. P. Barros, et al.
- **JOURNAL:** Procedia Computer Science, Elsevier, 2023, Vol. 222, Pages 81–93, <https://doi.org/10.1016/j.procs.2023.08.146>
- **METHODOLOGY:**

This paper investigates the feasibility of applying supervised machine learning techniques—Multilayer Perceptron (MLP), Decision Tree (DT), and Support Vector Machine (SVM)—for anomaly detection in 5G Open Radio Access Network (O-RAN) environments. Three real-world 5G datasets (DS-1, DS-2, DS-3), enriched with Key Performance Indicators (KPIs) like RSRP, RSRQ, and SNR, were used. The authors employed a novel labeling strategy based on t-SNE to balance data between anomalous (A) and non-anomalous (NA) classes. Models were trained using Python and tested across multiple scenarios (balanced, alternated, and unbalanced), achieving over 99% accuracy in many cases. MLP consistently outperformed the other models. The study demonstrates that anomaly signatures formed by combinations of KPIs can be effectively captured using supervised ML, supporting O-RAN's need for near-real-time and scalable anomaly detection.

[6]

- **TITLE:** Anomaly Detection for 5G Networks: Enhancing Scalability, Responsiveness, and Operational Efficiency
- **AUTHOR:** Gonela Kavya Pavani, Dr. Bobba Veeramallu
- **JOURNAL:** Journal of Neonatal Surgery, 2025, Volume 14, Issue 13s, Pages 71–82, <https://www.jneonatsurg.com>
- **METHODOLOGY:**

This work presents a deep learning-based anomaly detection system tailored for 5G networks, integrating Deep Neural Networks (DNNs) trained

on real-world Call Detail Records (CDRs). The approach emphasizes scalability and real-time responsiveness, achieved by deploying the system on Mobile Edge Computing (MEC) infrastructure. The authors implement a feedforward L-layer DNN model with Swish and ReLU activation functions and employ techniques like dropout, He/Xavier initialization, and L2 regularization for optimal performance. The model reaches up to 98.8% accuracy with a 0.44% false positive rate. The system supports adaptive learning, periodically retraining itself to handle evolving threats and dynamic network conditions. Key future directions include hybrid AI models, MEC deployment strategies, and human-in-the-loop integration for contextualized anomaly response.

[7]

- **TITLE:** 5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network
- **AUTHOR:** Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, et al.
- **JOURNAL:** arXiv, December 2022arXiv: 2212.01298v1, <https://arxiv.org/abs/2212.01298>
- **METHODOLOGY:**

This paper presents 5G-NIDD, a fully labeled network intrusion detection dataset generated over a real 5G test network at the University of Oulu using the 5GTN (5G Test Network) infrastructure. It covers nine intrusion types (including DoS, DDoS, and various port scans) and benign traffic captured using actual mobile devices. Attack scenarios were executed using open-source tools like Hping3, Nmap, Goldeneye, Slowloris, and Torshammer. The collected PCAP traffic was post-processed by removing GTP layers, converted into flow-based formats using Argus, labeled, normalized, and encoded. Feature selection was performed using Pearson Correlation and ANOVA F-score methods to select the top features. Machine learning models such as Decision Tree, Random Forest, KNN, Naive Bayes, and MLP were trained and evaluated using binary and multiclass classification. Random Forest and MLP achieved the best performance, with up to 99.5%

accuracy in multiclass scenarios. This dataset bridges the gap of realistic 5G traffic needed for effective ML-based anomaly detection and supports future work on distributed detection, active defenses, and adversarial machine learning.

[8]

- **TITLE:** Anomaly Detection in 5G using Variational Autoencoders
- **AUTHOR:** Amanul Islam, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim
- **JOURNAL:** SVCC 2024 Conference Paper;
[DOI:10.1109/SVCC61185.2024.10637312](https://doi.org/10.1109/SVCC61185.2024.10637312)

- **METHODOLOGY:**

This study applies Variational Autoencoders (VAE), an unsupervised machine learning technique, for detecting anomalies in the 5G-NIDD dataset. VAE is trained to reconstruct normal traffic patterns and flag anomalies via reconstruction errors. The dataset includes multiple real-world 5G attack scenarios, such as DoS and port scans. The paper highlights VAE's effectiveness in identifying both known and zero-day threats, offering a model-independent of prior attack signatures. Preprocessing steps include normalization, one-hot encoding, and feature engineering, with experiments confirming the VAE model's suitability for securing 5G NIDD-based networks.

[9]

- **TITLE:** Anomaly Detection Algorithms for Location Security in 5G Scenarios
- **AUTHOR:** Stefania Bartoletti, Ivan Palamà, Danilo Orlando, Giuseppe Bianchi, Nicola Blefari Melazzi
- **JOURNAL:** arXiv, March 2021, arXiv: 2103.12125v1,
<https://arxiv.org/abs/2103.12125>

- **METHODOLOGY:**

This paper proposes statistical anomaly detection algorithms to secure location services in 5G networks against noise-like jamming and

spoofing/meaconing attacks. The authors formulate both as binary hypothesis testing problems and propose solutions based on the Generalized Likelihood Ratio Test (GLRT) and Latent Variable Models (LVM). The methods are applicable to various types of location data like DOA, TOA, and RSRP. Using simulated data, the algorithms demonstrate high detection accuracy across different levels of interference and spoofing strength. The paper also introduces EM-based approaches for adaptive parameter estimation, highlighting a balance between computational complexity and detection effectiveness. These detection models can serve as early-warning systems for location-based 5G services.

[10]

- **TITLE:** Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks
- **AUTHOR:** Marouane Hachimi, Georges Kaddoum, Ghyslain Gagnon, Poulmanogo Illy
- **JOURNAL:** IEEE International Symposium on Networks, Computers and Communications (ISNCC), October 2020, arXiv: 2004.06077v, <https://arxiv.org/abs/2004.06077>
- **METHODOLOGY:** This paper introduces a multi-stage machine learning-based intrusion detection system (ML-IDS) for identifying and classifying jamming attacks (constant, random, deceptive, and reactive) in 5G Cloud Radio Access Networks (C-RAN). The architecture uses the LEACH routing protocol and is deployed in the virtualized Base Band Unit (BBU) pool. The system applies a two-layer classification approach: the first uses a Multilayer Perceptron (MLP) to detect traffic anomalies, and the second applies a Kernelized Support Vector Machine (KSVM) to re-evaluate MLP-classified normal traffic to reduce false negatives. The WSN-DS dataset, with over 374,000 labeled records and 23 features, is used for training and testing (70/30 split). The model achieved 94.51% classification accuracy, outperforming standalone MLP classifiers. This system offers enhanced

jamming detection capabilities with low latency impact and supports robust security in 5G C-RAN environments.

INFERENCE:

The development of our Machine Learning-based Anomaly Detection System for 5G Networks was shaped by insights gathered from a diverse range of studies. From [1] and [4], we incorporated the idea of transforming network flow data into image representations and applying CNNs and hybrid ensemble models for high-accuracy detection of anomalies in real-time. The use of Federated Learning (FL) to ensure data privacy while supporting distributed detection at edge and core levels, as discussed in [3], strongly influenced our architecture. To enhance model efficiency and minimize computational costs, techniques like Binary Particle Swarm Optimization (BPSO), correlation-based filtering, and ANOVA F-score—drawing inspiration from the methodologies in [4] and [7]—were utilized for feature selection and dimensionality reduction.

Furthermore, [5] and [6] guided us in applying supervised deep learning models like MLP and DNN on realistic datasets, achieving scalability and low-latency inference suitable for deployment on Mobile Edge Computing (MEC) infrastructures. From [2] and [6], we adopted behavior-based detection and predictive modeling to identify threats based on user behavior and traffic dependency using CR-Miner and Bayesian forecasting. The statistical anomaly detection approach presented in [9] inspired us to include hypothesis-based models for location-based threat identification, particularly for spoofing and jamming scenarios.

We also integrated insights from [8] on using unsupervised techniques like Variational Autoencoders (VAE) to capture zero-day threats without relying on prior signatures, thus strengthening the generalizability of our system. The comprehensive real-world dataset generation and multi-class classification strategy from [7] played a pivotal role in training and validating our models,

ensuring relevance to practical 5G scenarios. Finally, [10] contributed to our multi-stage classification approach by combining MLP and Kernelized SVM for enhanced detection of complex jamming attacks within 5G Cloud-RAN environments, improving detection robustness while minimizing false negatives.

CHAPTER 3

PROPOSED METHOD

This chapter outlines a novel approach for identifying anomalies in 5G networks by combining simulation-based analysis with machine learning algorithms, utilizing the NS3-mmWave framework to generate network traffic data and train models for effective anomaly detection.. The complete approach is illustrated using the schematic diagram and detailed module-wise explanation.

3.1 FLOW DIAGRAM

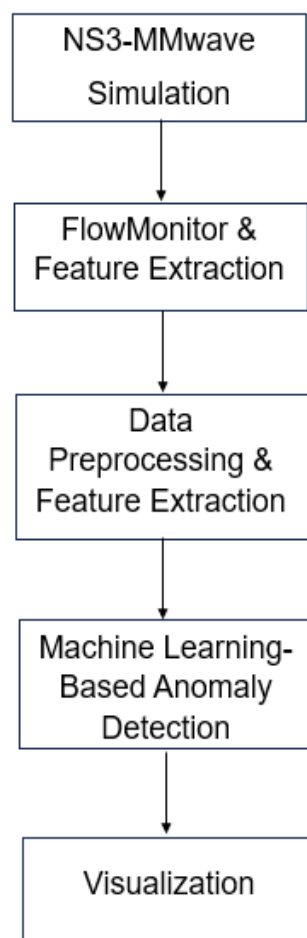


Figure 3.1 Flow Diagram for Machine Learning Based Anomaly Detection in 5G Network

The Figure 3.1 illustrates the overall process of machine learning-based anomaly detection in a 5G network, starting from simulation and data collection to model training and anomaly visualization.

3.2 MODULES

NS3-MMWAVE NETWORK SIMULATION MODULE:

This module emulates a realistic 5g network environment using the ns-3 mmwave extension. It includes user equipment (ue), evolved nodebs (enbs), and a remote host communicating via mmwave channels.

- **Input:** user-defined simulation parameters including traffic type, mobility, network topology, and link characteristics.
- **Output:** generates flowmonitor xml files and packet capture (pcap) files that contain detailed logs of packet transmission and flow statistics for further analysis.

FLOWMONITOR AND DATA CAPTURE MODULE:

Flowmonitor is integrated into the ns-3 simulation to track network behavior in real-time, such as throughput, delay, jitter, and packet loss.

- **Input:** packets transmitted in the simulation.
- **Output:** generates a detailed xml file (flowmon_results.xml) capturing flow-level statistics used for performance evaluation and anomaly detection.

DATA PREPROCESSING AND FEATURE EXTRACTION MODULE:

This module parses and converts flowmonitor xml output into csv format for easier processing by machine learning models. It extracts relevant features such as average delay, packet loss ratio, throughput, and jitter.

- **Input:** flowmon_results.xml or converted csv data.

- **Output:** structured dataset containing relevant features and labels for model training and inference.

MACHINE LEARNING-BASED ANOMALY DETECTION MODULE:

A trained machine learning model (e.g., svm, random forest, cnn, or autoencoder) is used to classify network traffic as normal or anomalous.

- **Input:** feature vectors obtained from simulation.
- **Output:** binary or multiclass labels indicating the presence of intrusion or anomaly types, and confidence scores or probabilities.

VISUALIZATION MODULE:

This module helps visualize real-time and historical detection results, enabling authorities or network administrators to act on threats quickly.

- **Input:** inference results from the ml model.
- **Output:** graphs, anomaly trends, heat maps, and alert messages (email, dashboard alert, etc.) Upon detecting unusual patterns in the traffic.

3.3 WORKING PRINCIPLE:

The proposed intrusion detection system operates by continuously analyzing network activity within a simulated 5G mmWave environment. Using the NS3-mmWave module, a realistic network scenario is created, involving several user devices and communication infrastructure. Throughout the simulation, FlowMonitor collects essential flow-level statistics, capturing metrics related to packet transmission and network performance.

This raw data is then parsed and converted into structured formats like CSV files, allowing for further processing through statistical methods and machine learning models. Key features such as delay, jitter, packet loss, and throughput are extracted to assess the network's operational state. These features are input into a machine learning classifier that has been trained on labeled data to effectively differentiate between typical and abnormal network behavior.

Anomalies are flagged when patterns deviate significantly from learned norms, indicating possible security incidents like DoS attacks or other network performance issues. A visualization layer complements the detection process, providing real-time dashboards and historical visual analytics to help users understand network conditions.

The system also supports integration with local or cloud-based dashboards, delivering real-time alerts and enabling quick intervention. Additionally, the framework allows for continuous learning by retraining the model with newly labeled data to refine its detection capabilities.

Overall, this approach not only replicates realistic 5G conditions but also embeds intelligent analysis tools, promoting proactive threat identification and response in next-generation mobile networks.

CHAPTER 4

SOFTWARE SETUP

This chapter provides a detailed overview of the software tools and platforms utilized for the development, simulation, data analysis, and anomaly detection involved in the proposed intrusion detection framework for 5G networks. Leveraging NS3-mmWave and machine learning techniques, each software element contributed to distinct phases of the project—from network simulation to the identification of abnormal behaviors through advanced data analysis.

4.1 WINDOWS SUBSYSTEM FOR LINUX (WSL)

Description:

The Windows Subsystem for Linux (WSL) is a robust compatibility framework created by Microsoft, enabling users to run complete Linux distributions directly within a Windows environment. This eliminates the need for dual-boot configurations or traditional virtual machines. WSL effectively connects Windows and Linux systems by allowing native Linux executables to operate on a Windows platform.

WSL supports various popular Linux distributions, including Ubuntu, Debian, and Kali, and grants access to a full Linux terminal and file system. This makes it possible for developers and researchers to utilize Linux-based tools, applications, and libraries seamlessly on a Windows machine.

WSL is available in two primary versions:

- **WSL 1:** The initial release, which converts Linux system calls into their Windows equivalents.
- **WSL 2:** An improved version that leverages an actual Linux kernel through a lightweight virtual machine, delivering enhanced performance and complete system call support.

Role in Project:

In this project, WSL plays a foundational role as the base environment required to run Linux-native software like NS-3 and its mmWave module. Since NS-3 is primarily developed for Linux, WSL allows its installation and usage on a Windows system, eliminating the need for a dedicated Linux machine.

The Figure 4.1 shows the official logo of the Windows Subsystem for Linux (WSL), representing its integration of Linux functionality within the Windows operating system.

Key roles in this project include:

- Providing a **Linux-compatible shell environment** (like Ubuntu) inside Windows for installing dependencies and running simulation scripts.
- Enabling direct execution of **NS-3 simulations** including the mmWave module and FlowMonitor.
- Allowing the use of **Linux commands** such as make, wget, cmake, g++, etc., which are essential for compiling and managing NS-3 code.
- Facilitating access to **Linux package managers** (like apt) to install libraries and tools needed for network simulation and anomaly detection.

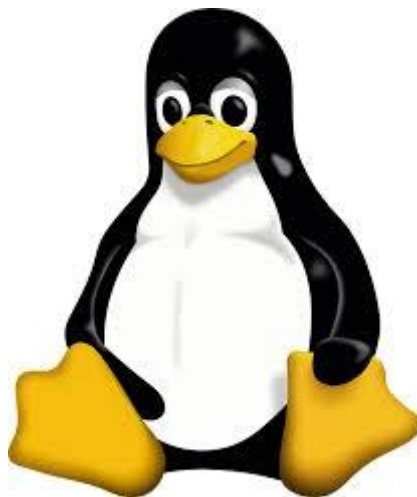


Figure 4.1 Windows Subsystem for Linux (WSL)

Installation:

To install WSL along with Ubuntu on Windows 10/11:

1. Open Windows Terminal or Command Prompt as Administrator.
2. Run the following command:

```
“ wsl – install ”
```

- Enables the required Windows features (such as Virtual Machine Platform and Windows Subsystem for Linux).
 - Downloads and installs the latest version of Ubuntu.
 - Sets up the default user account for Ubuntu.
3. Restart the system if prompted.
 4. After restarting, launch Wsl either through the Start menu or by typing:

```
“wsl”
```

5. Update and upgrade packages once inside wsl:

```
“sudo apt update
```

```
sudo apt upgrade”
```

6. Next step is want to install WSL 2 specifically and set it as default, you can use:

```
“wsl --set-default-version 2”
```

4.2 UBUNTU (VIA WSL)

Version Used: Ubuntu -22.04.5 LTS

Description:

Ubuntu is a popular, open-source Linux distribution based on Debian. It is known for its user-friendliness, strong community support, and compatibility with a vast array of open-source software packages. For research and development environments, especially in network simulation and academic experimentation,

Ubuntu is often the first choice due to its stability and long-term support (LTS) versions.

In this project, Ubuntu 20.04 LTS is installed and run via WSL (Windows Subsystem for Linux), which allows us to use Linux-based tools directly inside Windows. Ubuntu provides a rich command-line interface and access to key development tools needed for compiling and running NS-3, working with Python scripts, and managing simulation output files.

Role in Project:

Ubuntu plays a central role in this project by serving as the execution environment for the complete simulation and analysis pipeline. It provides the infrastructure and command-line tools necessary for:

- Installing and building NS-3 mmWave module, including all dependencies like g++, cmake, build-essential, and libraries required for the simulator to function.
- Running 5G network simulations via NS-3 and controlling packet-level behaviors across network entities (eNB, UE, remote host, etc.).
- Managing FlowMonitor and NetAnim outputs, including XML files that are later parsed and analyzed using Python.
- Installing and running Python libraries used for machine learning-based anomaly detection.
- Executing shell commands, writing and editing scripts, automating workflows, and troubleshooting compilation or runtime issues using standard Linux tools.

Ubuntu ensures a reliable and controlled simulation environment which is difficult to achieve using native Windows systems, especially when dealing with tools originally developed for Unix-like systems.

The Figure 4.2 illustrates the logo of the ubuntu Subsystem version 22.04.5

.



Figure 4.2 Ubuntu -22.04.5 LTS

Installation:

Once WSL is set up and Ubuntu is installed using the command:

```
"wsl --install -d Ubuntu -22.04.5 LTS"
```

Follow these steps to update and install the necessary packages:

1. Open Ubuntu terminal (via WSL):
2. Update and upgrade the system to ensure all packages are current:

```
"sudo apt update && sudo apt upgrade"
```

3. Install essential development tools and packages:

```
"sudo apt install build-essential git cmake python3-pip"
```

Explanation of key packages installed:

- **build-essential:** Includes compiler tools (like g++, make, and libc) needed to build NS-3 from source.
- **git:** used for cloning NS-3 and its mmWave module from GitHub repositories.
- **cmake:** Used for managing the build process of C++ projects like NS-3.
- **python3-pip:** Package manager for Python used to install ML libraries such as pandas, numpy, scikit-learn, and more.

4.3 NS-3 NETWORK SIMULATOR

Version Used: ns3-mmwave (latest stable release supporting 5G simulations)

Description:

NS-3 (Network Simulator 3) is a widely used discrete-event network simulator, primarily employed in networking research. Written in C++ with Python bindings, it is designed to simulate the behavior of various network protocols and systems in a realistic and manageable environment. The ns3-mmwave module enhances the standard NS-3 by adding support for 5G millimeter-wave (mmWave) communication models, allowing researchers to assess high-frequency wireless systems, such as those deployed in next-generation mobile networks.

Role in Project:

- Serves as the primary simulation platform for replicating an authentic 5G network environment.
- Supports the modeling of network topologies that include User Equipments (UEs), evolved NodeBs (eNBs), and remote servers.
- Simulates various network parameters such as traffic flows, packet transmissions, mobility patterns, and environmental factors.
- Produces FlowMonitor XML output, essential for assessing network performance and identifying anomalies.
- Enables seamless integration with custom 5G scenarios, allowing the study of the effects of irregular events like packet loss or network congestion.

The Figure 4.3 illustrates the logo of network simulator ns-3.



Figure 4.3 NETWORK SIMULATOR (NS-3)

Installation:

1.Update and install dependencies

Run the following command to update and install the required packages:

```
sudo apt update && sudo apt install -y git cmake g++ python3-pip
```

2.Clone the ns-3 mmwave module

Use the command below to clone the ns-3 mmwave module repository:

```
git clone https://github.com/nyuwireless/ns3-mmwave.git
```

Then, navigate into the directory:

```
cd ns3-mmwave
```

3.Configure and compile the simulator

Execute these commands to configure and build the simulator:

```
./waf configure
```

```
./waf build
```

After installation, NS-3 can be utilized to run example simulations or custom scripts, where users can define network topology, traffic patterns, and flow monitoring configurations for 5G networks.

4.4 FLOW MONITOR TOOL

Version Used: Integrated module within ns3-mmwave

Description:

Flow Monitor is an efficient, integrated performance monitoring tool within NS-3 that is designed to collect and analyze data on packet flows in a simulated network. It works by being attached to the nodes in the simulation, where it tracks key performance indicators like packet loss, delay, throughput, and jitter for each individual flow. The results are typically saved in an XML file format, which can be parsed later for detailed evaluation.

Role in Project:

- Captures detailed flow-level metrics during 5G simulations performed with the ns3-mmwave module.
- Generates an XML output (flowmon_results.xml) that contains information about all active data flows.
- Enables extraction of critical network behavior data used for machine learning-based anomaly detection.
- Serves as the bridge between NS-3 and the machine learning environment by providing structured, analyzable data.
- Helps identify performance degradation caused by anomalies, such as unexpected delays or high packet loss ratios.

Installation:

FlowMonitor is already included by default in most NS-3 installations, including ns3-mmwave. However, ensure it is enabled during configuration:

```
“./waf configure --enable-examples  
./waf build”
```

The generation of XML file becomes the input for the Python-based data extraction and machine learning processing pipeline, making FlowMonitor a vital component of the anomaly detection system.

4.5 NETANIM (NS-3 NETWORK ANIMATOR)**Description:**

NetAnim is a graphical tool that animates the movement and communication between network nodes in NS-3 simulations. It provides a visual representation of how data is transferred across the simulated 5G network.

Role in Project:

NetAnim is used to:

- Visually analyze packet movement between eNBs and UEs.
- Observe node behavior and packet drops in real-time.
- Validate the simulation logic and traffic flow visually.
- Support documentation and presentations with animated proof of simulation.

Installation:

1. Make sure Qt5 is installed:

```
“sudo apt-get install qt5-default”
```

2. Clone and build NetAnim from the NS-3 website or GitHub:

```
“cd ~
```

```
git clone https://gitlab.com/nsnam/netanim.git
```

```
cd netanim
```

```
qmake NetAnim.pro
```

```
make”
```

3. Run NetAnim:

```
“./NetAnim”
```

4.6 PYTHON PROGRAMMING LANGUAGE**Description:**

Python is a high-level, interpreted programming language known for its simplicity and powerful libraries for data analysis, scientific computing, and

machine learning. Its syntax is user-friendly and widely adopted in both research and industry for prototyping and implementation.

Role in Project:

Python plays a central role in this project by handling the post-processing and analysis of simulation data generated by NS-3. It is used to:

- Parse the FlowMonitor XML output.
- Convert and clean data into CSV format.
- Apply machine learning algorithms to detect network anomalies.
- Generate visualization plots for result analysis.

The Figure 4.4 illustrates the logo of python programming language.



Figure 4.4 Python Programming Language

Installation:

Python is typically pre-installed in most Ubuntu distributions. If not, can install it using:

```
“sudo apt update
```

```
sudo apt install python3 python3-pip”
```

4.7 PCAP FILE GENERATION

Description:

PCAP (Packet Capture) files store network traffic data, capturing the details

of every packet sent or received during simulation. These files are readable using tools like Wireshark for in-depth packet-level analysis.

Role in Project:

Although the main analysis is done using FlowMonitor, PCAP files are optionally used to:

- Verify individual packet transmission behavior.
- Identify packet-level anomalies that may not be captured by flow-level metrics.
- Cross-validate the flow-based anomaly detection process.

Installation:

To enable PCAP generation in NS-3, include this line in your C++ script:

```
“AsciiTraceHelper ascii;  
  
pointToPoint.EnablePcapAll("outputfile");”
```

No separate installation is needed as it is a part of NS-3 modules.

4.8 XML PARSING TOOLS (xml.etree.ElementTree)

Description:

xml.etree.ElementTree is a built-in Python library for reading and parsing XML files, which are commonly used for storing structured data.

Role in Project:

FlowMonitor generates simulation output as an XML file (flowmon_results.xml). This library is used to:

- Traverse XML elements.
- Extract flow-specific metrics (like throughput, packet loss).
- Transform this data into structured formats for machine learning.

Installation:

This library comes bundled with Python, so no separate installation is required.

4.9 CSV FILE HANDLING (pandas)**Description:**

Pandas is a Python library that offers robust data structures, such as DataFrames, to handle structured data like tables or CSV files efficiently.

Role in Project:

After parsing the XML file, pandas is used to:

- Store extracted features in tabular format.
- Clean, normalize, and prepare datasets.
- Export and import data for machine learning model training and evaluation.

Installation:

To install pandas the following command is used to run

```
"pip install pandas"
```

4.10 NUMPY**Description:**

NumPy is a core Python library used for numerical computations and array-based processing. It enables matrix operations, statistical analysis, and efficient handling of large datasets.

Role in Project:

NumPy is used to:

- Perform mathematical computations on flow statistics.
- Handle large numerical datasets efficiently.
- Support feature engineering and data normalization processes.

Installation:

To install numpy the following command is used to run

```
"pip install numpy"
```

4.11 SCIKIT-LEARN (SKLEARN)**Description:**

Scikit-learn is a widely used Python library for building and evaluating machine learning models. It offers tools for classification, regression, clustering, model validation, and preprocessing.

Role in Project:

Scikit-learn is essential for:

- Training models like Decision Trees, SVM, Random Forests.
- Splitting datasets into training and testing sets.
- Evaluating models using metrics like accuracy, precision, recall, and F1-score.

Installation:

To use scikit-learn use the following command

```
"pip install scikit-learn"
```

4.12 VISUALIZATION LIBRARIES (matplotlib and seaborn)

Description:

- **Matplotlib** is a plotting library used for creating static, interactive, and animated visualizations.
- **Seaborn** is built on top of matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.

Role in Project:

These libraries are used to:

- Visualize model performance metrics.
- Plot anomaly detection graphs and flow behavior.
- Generate bar charts, confusion matrices, heatmaps, etc.

Installation:

Command to Install Visualization Libraries for Anomaly Detection Analysis

```
"pip install matplotlib seaborn"
```

4.13 FILE HANDLING AND AUTOMATION TOOLS (os, glob)

Description:

- **os**: A built-in Python module for interacting with the operating system.
- **glob**: A module used to find files and directories using patterns (wildcards).

Role in Project:

Used to:

- Navigate and automate access to simulation output directories.
- Search and read multiple XML or CSV files.
- Build scripts that automate preprocessing steps across multiple runs.

Installation:

Both `os` and `glob` are part of the Python Standard Library and do not require installation.

CHAPTER 5

SOFTWARE IMPLEMENTATION

5.1 OVERVIEW

This chapter details the practical implementation of the machine learning-based anomaly detection system integrated with the 5G network simulation using the ns3-mmwave module. It explains how each component—ranging from network simulation to machine learning integration—was configured, executed, and evaluated. This includes packet transmission analysis, XML/CSV data handling, and training models to classify normal vs. anomalous behavior.

The Figure 5.1 illustrates a diagram of a system architecture .

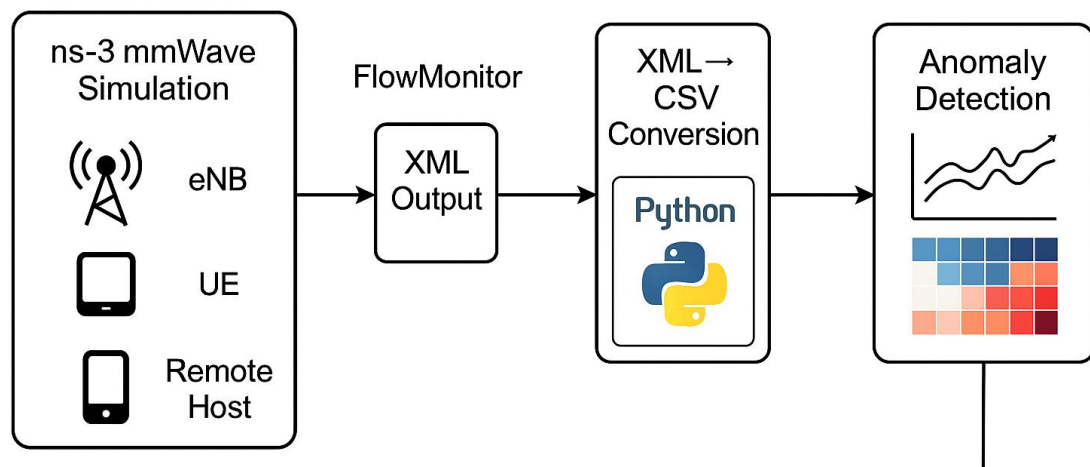
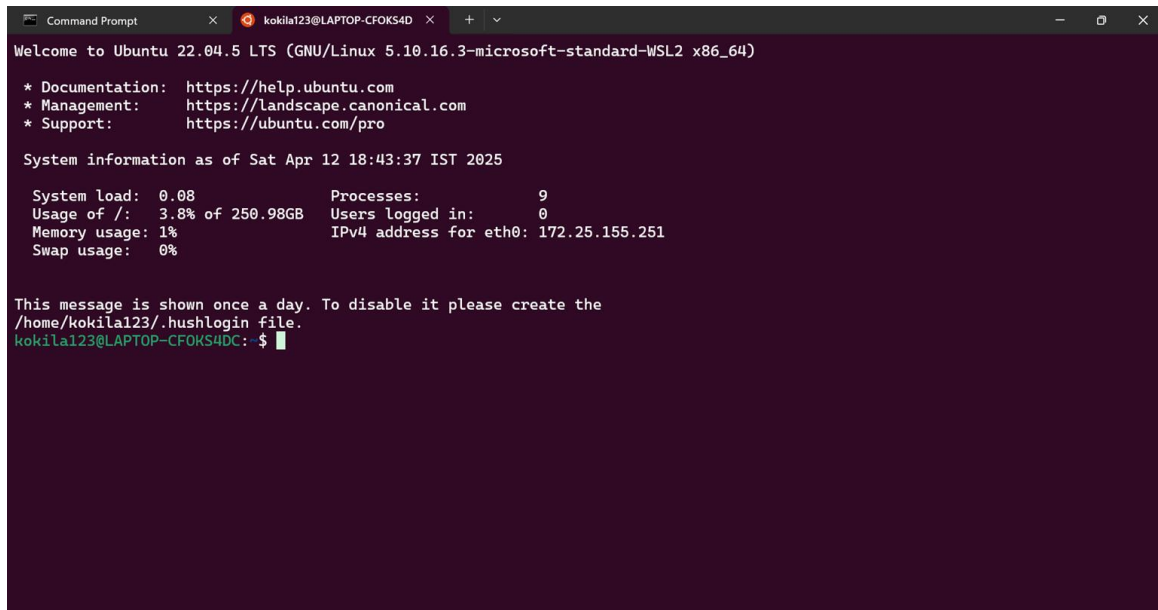


Figure 5.1 System Architecture

5.2 SETTING UP THE ENVIRONMENT

Before implementing the anomaly detection system, the environment is configured with essential tools such as Windows Subsystem for Linux (WSL), Ubuntu 20.04 LTS, and Python 3. These form the base to execute ns-3 simulations and perform subsequent data processing.

The Figure 5.2 illustrates the ubuntu terminal.



```
Command Prompt x kokila123@LAPTOP-CFOK54D x + v
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

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

System information as of Sat Apr 12 18:43:37 IST 2025

System load:  0.08          Processes:            9
Usage of /:   3.8% of 250.98GB Users logged in:       0
Memory usage: 1%          IPv4 address for eth0: 172.25.155.251
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/kokila123/.hushlogin file.
kokila123@LAPTOP-CFOK54DC:~$
```

Figure 5.2 Ubuntu terminal

5.3 NS-3 MMWAVE SIMULATION SETUP

Details how NS-3 mmWave is installed and configured to simulate a 5G network. It includes the script modifications, eNodeB, UE setup, mobility models, and application traffic.

- **Cloning and Building:** The ns3-mmwave module is downloaded from its official repository using git. It is then compiled using the waf build system.
- **Dependencies:** Essential build tools, Python packages, and simulation modules are installed.

This module supports mmWave (millimetre-wave) communication simulations which are crucial for realistic 5G network modelling. Figure 5.3 illustrates the visually represents the simulation environment built using the ns3-mmwave module. It includes multiple User Equipments (UEs), one or more eNodeBs (base stations), and a remote host, all connected through the mmWave channel. Data traffic flows between the remote host and UEs via the EPC (Evolved Packet Core), simulating real-world 5G communication.

The Figure 5.3 illustrates the NS-3 mmWave Simulation Topology.

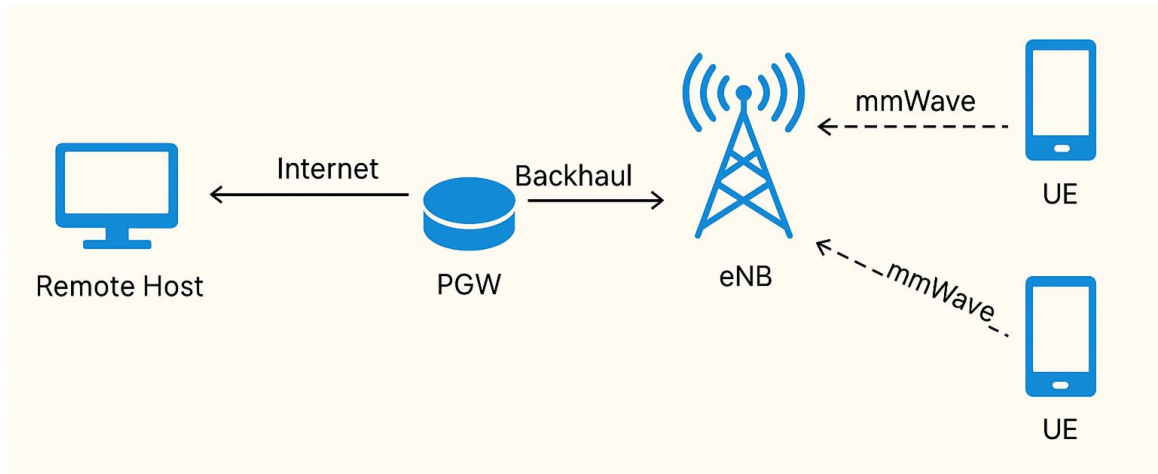


Figure 5.3 NS-3 mmWave Simulation Topology

5.4 NETWORK DESIGN AND SIMULATION (CUSTOM_ANOMALY.CC)

In this section, we define the network design and simulation configuration for the 5G mmWave simulation using NS-3. This code simulates a basic 5G network with eNB (evolved Node B) and UE (User Equipment) nodes, establishes traffic flows between nodes, and prepares the network for anomaly detection. Additionally, this code integrates NetAnim for network visualization and FlowMonitor to capture flow statistics for later analysis.

5.4.1 Code writing and Setup

To create the `custom_anomaly.cc` file in the scratch directory of the NS-3 setup, follow these steps:

1. The scratch/ folder in NS-3 is used to write and test custom simulation scripts.

```
"cd ~/ns3-mmwave/scratch"
```

2. Use the nano text editor to create a new C++ file named `custom_anomaly.cc`.

```
"nano custom_anomaly.cc"
```

3. Paste the code which creates the network in the nano editor and After pasting the code, press CTRL + X to exit, then press Y to confirm saving the file, and press Enter to confirm the file name.

4. The script performs the following actions:

- **Creates nodes:** 2 eNBs nodes (representing base stations) and 4 UEs (representing mobile devices).
- **Sets mobility:** Grid-based constant positions.
- **Traffic Flow Setup:** A UDP client-server application is used to simulate traffic between one of the UEs (UE4) and another UE (UE3). The client sends UDP packets at a specified interval, while the server receives these packets.
- **Integrates:**
 - **FlowMonitor** to collect traffic statistics.
 - **NetAnim** to visualize the topology.

Once you've written the code, you can build and run the simulation.

5.4.2. Building and Running the Code:

1. Build NS-3: Navigate back to the root directory of the ns3-mmwave folder and build the NS-3 simulation:

```
"cd ~/ns3-mmwave  
./ns3 build"
```

2. Run the Simulation:

After building the simulation, execute it with the following command:

```
"./ns3 run scratch/custom_anamoly.cc"
```

This will start the simulation, where the flow of packets will be generated, and the FlowMonitor will capture the statistics.

5.4.3. Network Visualizing with NetAnim:

After running the simulation, you can visualize the network using the NetAnim tool. Go to the build/bin directory and run:

```
"cd ~/netanim/build/bin  
./netanim"
```

This will open the NetAnim graphical user interface (GUI), where you can load the `anamolynetwork.xml` file generated by the simulation. This file contains the network topology and node movements.

The Figure 5.4 illustrates the netanim window.



Figure 5.4 Netanim Window

5.5 FLOWMONITOR CONFIGURATION AND XML GENERATION

This section, explain how the FlowMonitor is configured within the simulation script (`custom_anomaly.cc`) and how it collects statistics during the 5G mmWave network simulation. These statistics are crucial for further anomaly detection using Machine Learning.

5.5.1 INTEGRATING FLOWMONITOR

The FlowMonitor module in NS-3 is used to monitor packet flows, detect drops, and generate performance metrics such as delay, jitter, and throughput.

Steps to Configure FlowMonitor in the Script:

1. Add Header: Include the FlowMonitor header in your C++ script:

```
"#include "ns3/flow-monitor-helper.h" "
```

2. Install FlowMonitor: At the end of your simulation script (after setting up all applications), install FlowMonitor on all nodes:

```
"FlowMonitorHelper flowmonHelper;
```

```
Ptr<FlowMonitor> monitor = flowmonHelper.InstallAll();"
```

3. Generate XML File: After the simulation is run, instruct NS-3 to save the FlowMonitor results to an XML file using:

```
"monitor->SerializeToXmlFile("flowmon_results.xml", true, true);"
```

The file flowmon_results.xml will be created in the ns3-mmwave base directory. This file contains detailed statistics on packet transmission, delays, jitter, lost packets, throughput, and flow IDs.

5.5.2 RUN AND GENERATE FLOWMONITOR OUTPUT

Once FlowMonitor is set up, can build and run the script as shown before:

```
"cd ~/ns3-mmwave
```

```
./ns3 run scratch/custom_anomaly.cc"
```

After the simulation completes, confirm that the flowmon_results.xml file has been generated.

The output XML file includes entries like:

```
"<FlowStats flowId="1" timeFirstTxPacket="..."  
timeLastRxPacket="..." delaySum="..." rxBytes="..."  
txPackets="..." />"
```

This file will later be parsed and converted to CSV for use in machine learning analysis (explained in later sections).

5.6 PCAP GENERATION AND WIRESHARK ANALYSIS

This section describes how to collect traffic data during the simulation using NS-3's built-in pcap tracing and analyze the captured packets using Wireshark. This step is essential for identifying anomalies in 5G mmWave networks, as it generates detailed traffic data that can be converted into CSV format and utilized in machine learning algorithms.

5.6.1 PACKET CAPTURE MECHANISM IN NS-3

NS-3 offers native support for packet tracing in the form of .pcap files. These files NS-3 provides built-in support for packet tracing through the use of .pcap files, which contain comprehensive data on each packet sent or received during the simulation.

How it works:

- The EnablePcapAll() function, available in NS-3's helper classes, is used to capture packets.
- Packet tracing is activated after mmWave devices are installed on eNB and UE nodes.
- The resulting .pcap files capture information like timestamps, protocols, IP addresses, packet sizes, and more.

5.6.2 CODE MODIFICATIONS FOR ENABLING PACKET TRACING

The 5G mmWave network topology is designed and the necessary network nodes are configured, packet capture functionality is incorporated by updating the simulation script. This ensures that during the simulation, all traffic is recorded for post-simulation analysis.

To activate packet capture in the simulation:

1.Navigate to NS-3 scratch folder:

```
"cd ~/ns3-mmwave/scratch"
```

2.Open the simulation script for editing:

```
"nano custom_anamoly.cc"
```

3.Insert the packet capture line Locates the section immediately after the mmWave devices are installed on eNB and UE nodes.

Paste the following line after installing mmWave devices:

```
" cpp mmWaveHelper->EnablePcapAll("anamoly_trace");"
```

This command enables pcap tracing on all NetDevices created by the mmWaveHelper instance. The captured pcap files will be stored in the NS-3 working directory with the prefix anamoly_trace.

4.Save and exit the file:

Press CTRL + X, then Y, and Enter to save.

5.6.3 BUILDING AND RUNNING THE SIMULATION

Before running the simulation, it is essential to rebuild the project to incorporate recent modifications.

Build the Simulation Navigate to the NS-3 project directory and initiate the build process:

```
"cd ~/ns3-mmwave"
```

```
./ns3 build"
```

Execute the Simulation Once the build process completes without errors, run the simulation:

```
“./ns3 run scratch/custom_anamoly.cc”
```

During execution, NS-3 generates several .pcap files in the project's root directory. Each pcap file corresponds to a network interface on a simulated node, containing timestamped packet records captured during the simulation runtime.

5.6.4 TRAFFIC ANALYSIS AND CSV CONVERSION USING WIRESHARK

While FlowMonitor provides flow-level data, **Wireshark** enables **packet-level inspection** which is necessary for accurate anomaly detection. Steps to analyze and export

1.Launching Wireshark:

Open Wireshark by running the following command in the terminal:

```
“wireshark”
```

2.Loading and Viewing PCAP Files:

- Navigate to File → Open
- Select the desired pcap file (e.g., anamoly_trace-0-0.pcap)
- Click Open to visualize the captured traffic

3.To isolate specific types of traffic (such as UDP, TCP, or ICMP), apply a display filter in the Wireshark filter bar.

```
“udp”
```

4.Export as CSV:

- Go to File → Export Packet Dissections → As CSV...
- Choose to export all packets or filtered packets.
- Select relevant fields like:
 - Packet number

- Timestamp
- Source/Destination IP
- Protocol
- Packet size
- Save as anamolny_trace.csv

This CSV file will contain structured packet data that can be used for machine learning.

5.6.5 ROLE OF CSV DATA IN ANOMALY DETECTION.

- CSV files act as the **input dataset** for anomaly detection algorithms.
- By analyzing attributes like **packet count, sizes, protocol patterns**, the system can detect **suspicious traffic behavior**.

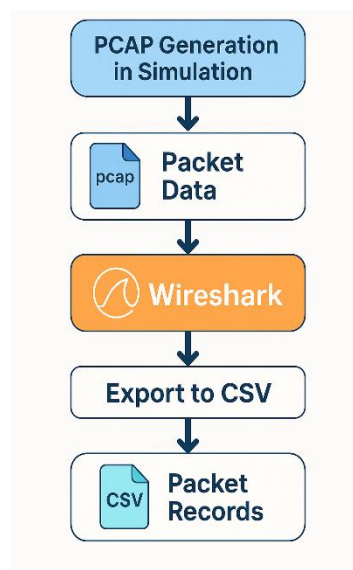


Figure 5.5 PCAP Capture and CSV Export using Wireshark.

- This helps in identifying scenarios such as **DDoS attacks, traffic spikes, or unauthorized access patterns**.
- Consistent CSV logs also enable benchmarking across different simulation runs. The Figure 5.5 represents PCAP Capture and CSV Export using Wireshark

5.7 FLOW DATA CONVERSION AND PREPROCESSING

This section explains how the flow-level network performance data—collected by the FlowMonitor module in NS-3—is converted from XML format into a structured CSV file suitable for further analysis by machine learning algorithms. The conversion process involves parsing the XML to extract key metrics such as delay, jitter, transmitted and received bytes, packets, lost packets, and the computed packet loss ratio. The resulting CSV file forms the core dataset for the subsequent anomaly detection phase.

5.7.1 SCRIPT CREATION AND SETUP

1. Navigate to the Results Directory:

The results directory is used to store all output files and conversion scripts.

```
“cd ~/ns3-mmwave-anomaly-project/results/”
```

2. Create and Edit the Python Script:

Use the Nano editor to create the conversion script.

```
“nano flowmon_xml_to_csv.py”
```

This command opens the text editor, allowing you to type or paste the Python code.

3. Save the Script: After entering the code, press CTRL + X, then Y, and finally Enter to save the file.

5.7.2 SCRIPT DESCRIPTION

This Python script leverages the `xml.etree.ElementTree` module to parse the XML file generated by the FlowMonitor module in NS-3. The process extracts relevant

metrics for each network flow and organizes them into a structured CSV format. The selected metrics include:

- **DelaySum**: Total cumulative delay for all packets in a flow.
- **JitterSum**: Total jitter experienced in the flow.
- **TxByte**: Total bytes transmitted.
- **RxByte**: Total bytes received.
- **TxPacket**: Number of packets transmitted.
- **RxPacket**: Number of packets received.
- **LostPacket**: Number of packets lost during transmission.
- **PacketLossRatio**: Ratio of lost packets to transmitted packets.

The script ensures that this data is systematically converted into a CSV file, which can be easily accessed by machine learning libraries such as scikit-learn and pandas.

5.7.3 RUNNING THE SCRIPT

After saving the script, execute it by navigating to the results directory and running the following command:

```
python3 flowmon_xml_to_csv.py
```

Upon successful execution, a message will confirm that the CSV file has been created in the directory.

5.8 MACHINE LEARNING MODEL AND ANOMALY DETECTION

This section outlines how to employ a machine learning approach for detecting anomalies in network simulation data. The primary goal is to identify abnormal network behaviors—such as excessive packet delay, jitter, and high packet loss ratios—using an unsupervised algorithm. The process involves converting the preprocessed CSV data into a format suitable for model training,

applying the Isolation Forest algorithm for anomaly detection, and visualizing the results.

5.8.1 CODE WRITING AND SETUP

1. Navigate to the Results Directory

Change to the directory where your simulation output and analysis scripts are stored:

```
cd ~/ns3-mmwave-anomaly-project/results
```

2. Create the Anomaly Detection Script

Create and edit the Python file using a text editor like nano:

```
nano detect_anomalies.py
```

Paste the code into the editor, then press **CTRL + X**, **Y**, and **Enter** to save.

5.8.2 RUNNING THE SCRIPT

1. Execute the Python Script

From the results directory, run:

```
python3 detect_anomalies.py
```

2. Upon execution:

- **anomaly_detection_results.csv** will be generated, containing the detection outcomes.
- **anomaly_plot.png** will be created to visually represent anomalies based on the DelaySum of each flow.

5.9 GRAPHICAL REPRESENTATION AND ANALYSIS

This section explains how to generate visual graphs that represent the anomaly detection results. Graphical visualization is essential for interpreting network performance metrics such as delay, jitter, and packet loss, and for distinguishing between normal and anomalous network flows.

5.9.1 CODE WRITING AND SETUP

1. Navigate to the Results Directory

```
cd ~/ns3-mmwave-anomaly-project/results
```

This directory contains all the output files from the simulation and conversion scripts, including CSV files and anomaly detection results.

2. Create and Edit the Python Script

```
nano visualize_results.py
```

This command opens the nano text editor, where you will write the Python code for visualizing the anomaly detection results.

3. Save the Script

After inserting the code, press **CTRL + X** to exit, **Y** to confirm saving, and **Enter** to finalize the file name.

5.9.2 RUNNING THE SCRIPT

After saving the script, execute the following command in the results directory:

```
python3 visualize_results.py
```

This command runs the script, generates the plots, and saves the images in the same directory. Upon execution, three pop-up windows will display the scatter plots sequentially. Each plot represents the anomaly detection results based on different network performance metrics.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 OVERVIEW

This chapter presents the results of the simulation and machine learning-based anomaly detection carried out in the 5G mmWave network using the NS-3 mmWave module. It includes the visual and analytical interpretation of network performance metrics, anomaly identification, and the performance of the detection system.

6.2 NETANIM VISUALIZATION OUTPUT

NetAnim, a powerful tool integrated with NS-3, was used to visualize and observe the dynamic behavior of the 5G mmWave network in a graphical environment. The use of NetAnim enabled the visual validation of network topology, node interactions, and packet flow, providing insights that are often difficult to derive from raw data or terminal logs.

In the simulation, several User Equipment (UE) nodes were configured to move according to predefined mobility models, communicating with stationary base stations (eNBs) and a remote host. The animation provided a clear depiction of the UEs' real-time mobility patterns, handover events between eNBs, and continuous data flow. Each node was represented by unique IDs and icons, and data packets were visualized as moving arrows, with color changes based on their transmission status.

Key insights from the NetAnim visualization include:

- **Node Mobility and Interaction:** UE nodes were shown moving across the simulation area, with mobility traces validating the accuracy of movement models and proper handovers between base stations.

- **Data Transmission Flow:** The graphical interface demonstrated successful end-to-end communication from the UE to the eNB and then to the remote host, confirming proper routing and link establishment.
- **Network Behavior Over Time:** High-traffic periods were represented by increased packet flow density, while idle periods were marked by reduced activity. Congested nodes were visible through high packet queueing or delay, facilitating the observation of performance bottlenecks.

Overall, NetAnim served as a crucial verification tool, enabling the visualization of underlying communication mechanisms in the 5G mmWave simulation. It reinforced the correctness of node configuration, routing logic, and packet delivery, thereby supporting the authenticity of the experiment.

The Figure 6.1 shows the selection of xml directory

The Figure 6.1, 6.2, 6.3 illustrates the selection of the ns3-mmwave folder, which is located within the primary NS-3 directory.

The ns3-mmwave module extends the NS-3 simulator with support for 5G mmWave simulations, including customizable PHY and MAC layers, beamforming, channel modeling, and scheduler functionalities. Accessing this directory is a crucial step before building, modifying, or executing simulation scripts like `custom_anomaly.cc`. It allows users to explore source files, example scripts, helper classes, and configuration files specific to mmWave network simulations. This step is typically performed using a Linux file manager or terminal navigation inside the NS-3 development environment (e.g., in Ubuntu within WSL or VirtualBox).

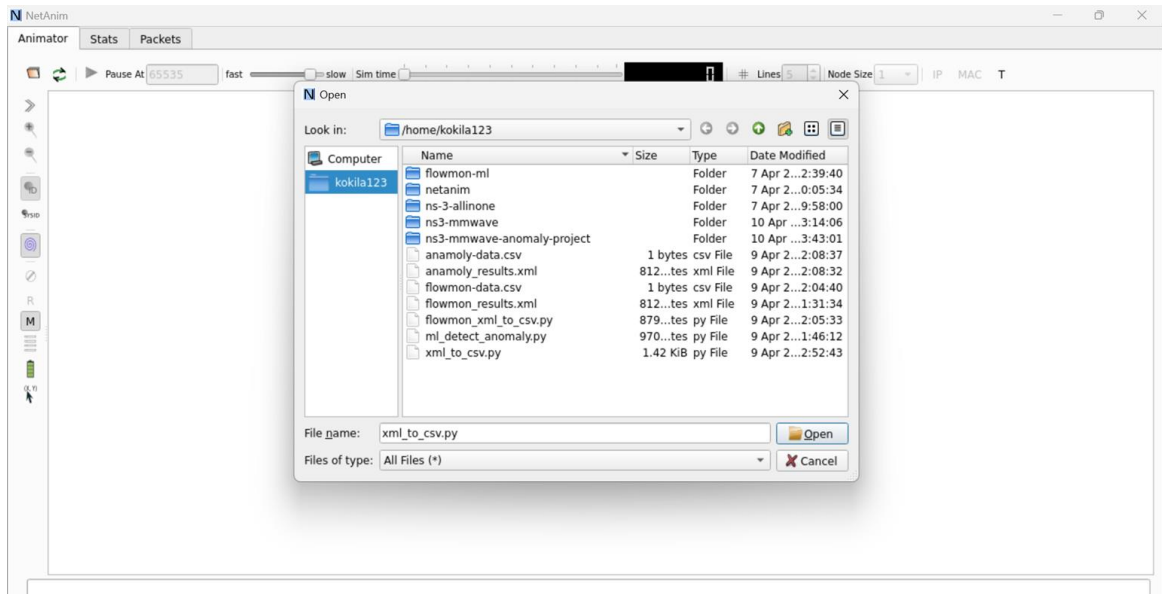


Figure 6.1 Selecting the xml file directory

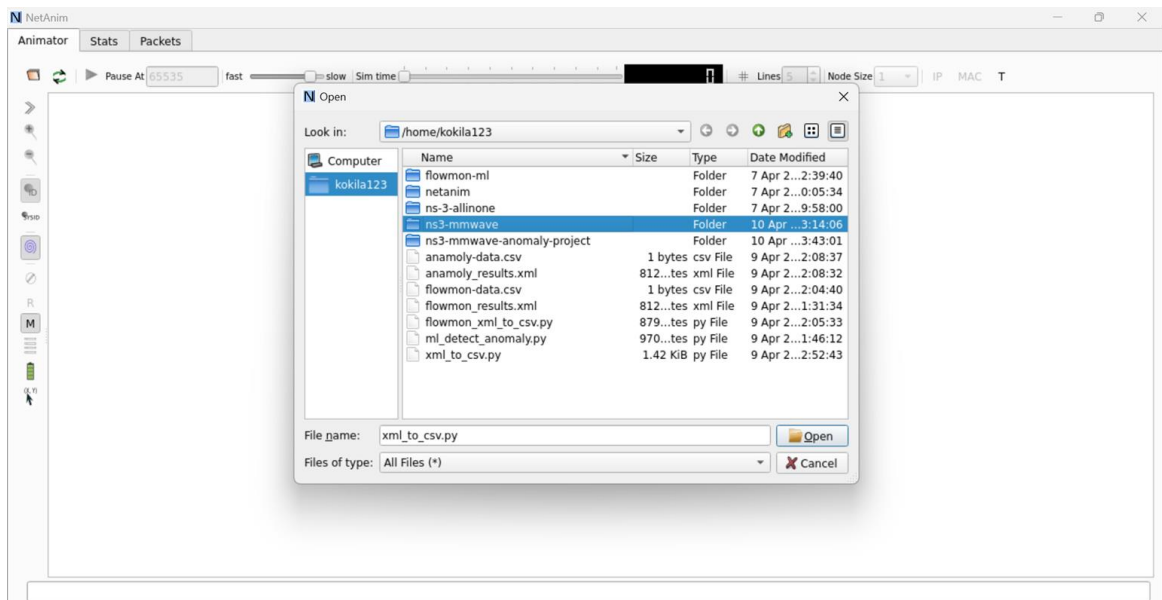


Figure 6.2 Selecting the ns3-mmwave Directory

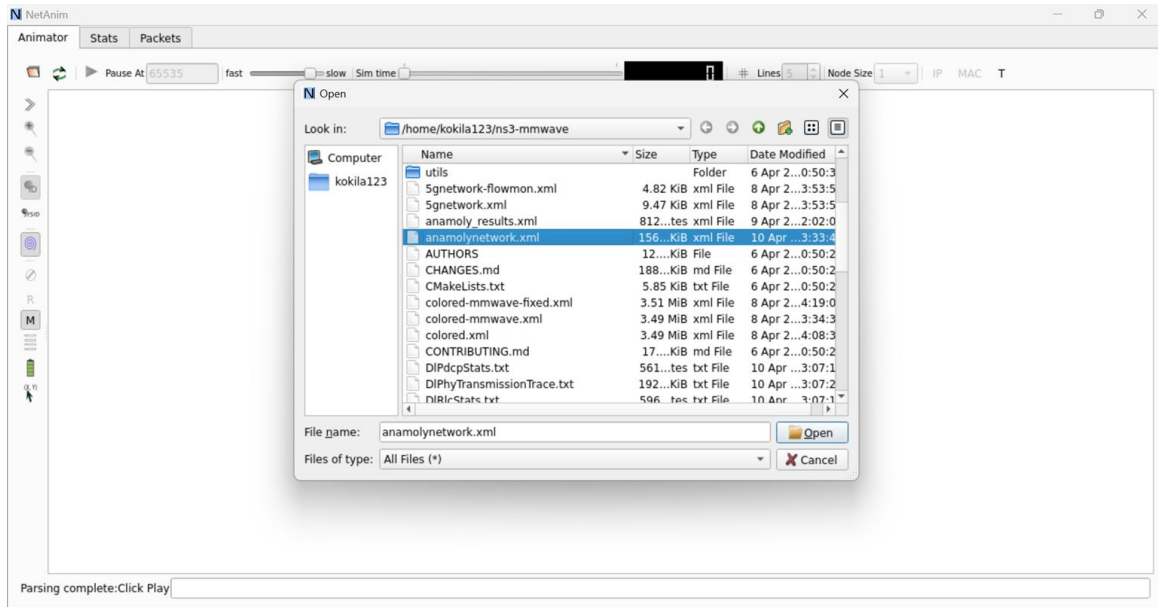


Figure 6.3 Selecting the generated xml file

The Figure 6.4 shows the animated simulation of the 5G mmWave network created using NS-3 and visualized through NetAnim.

Nodes including eNodeBs, UEs, and a remote host are illustrated with their mobility paths and packet flow directions. The green lines indicate active wireless connections and routing paths used during communication.

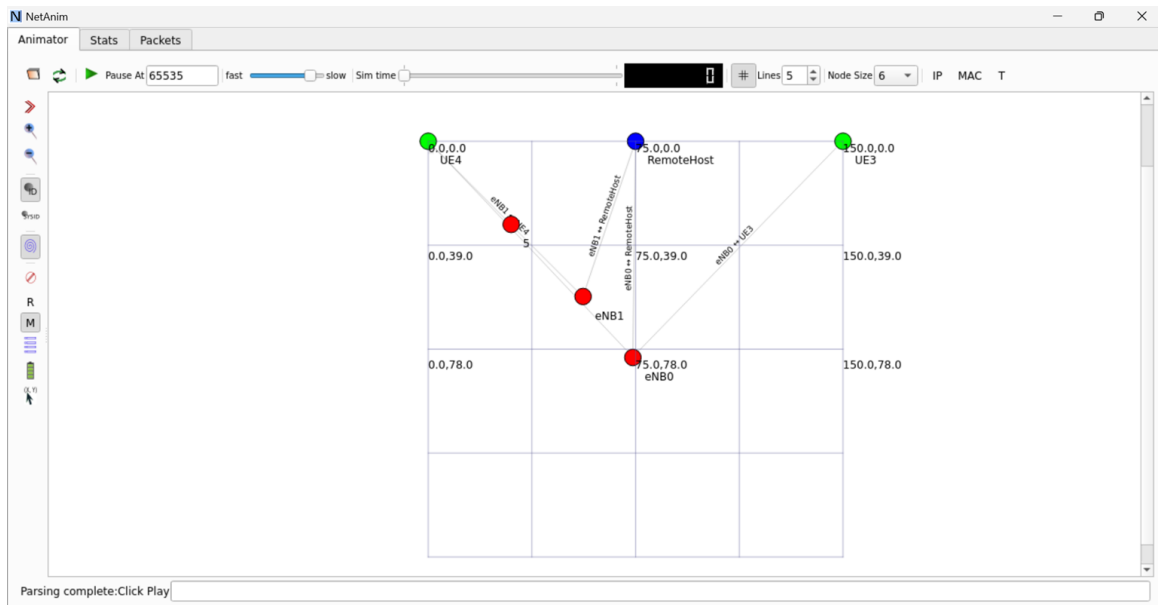


Figure 6.4 Created network

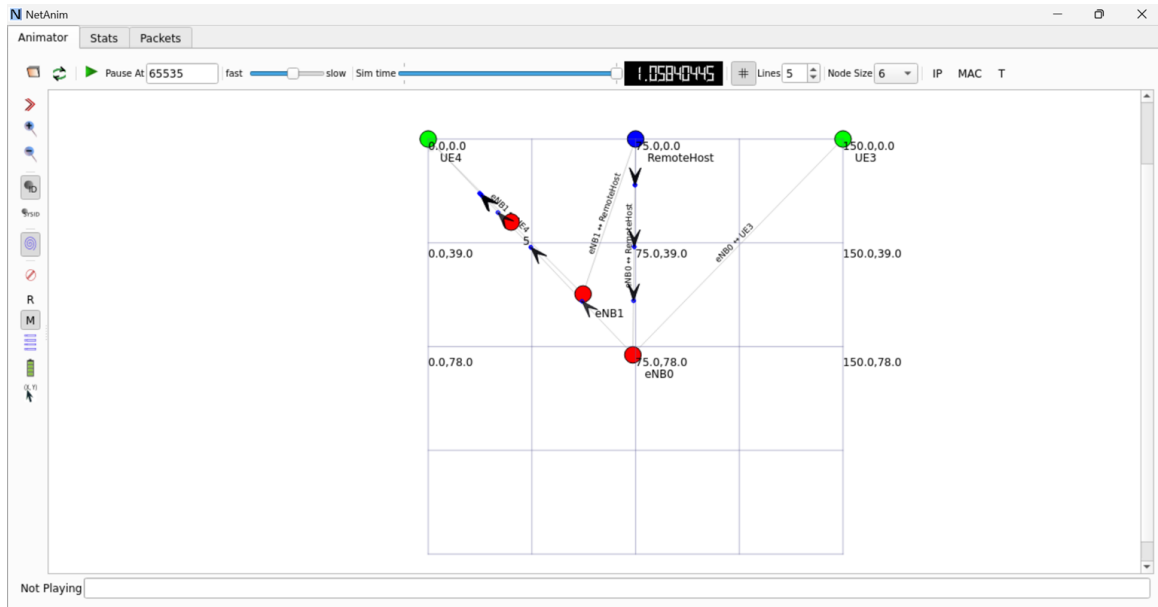


Figure 6.5 Animated network

The Figure 6.5 represents the flow of packets in the created network

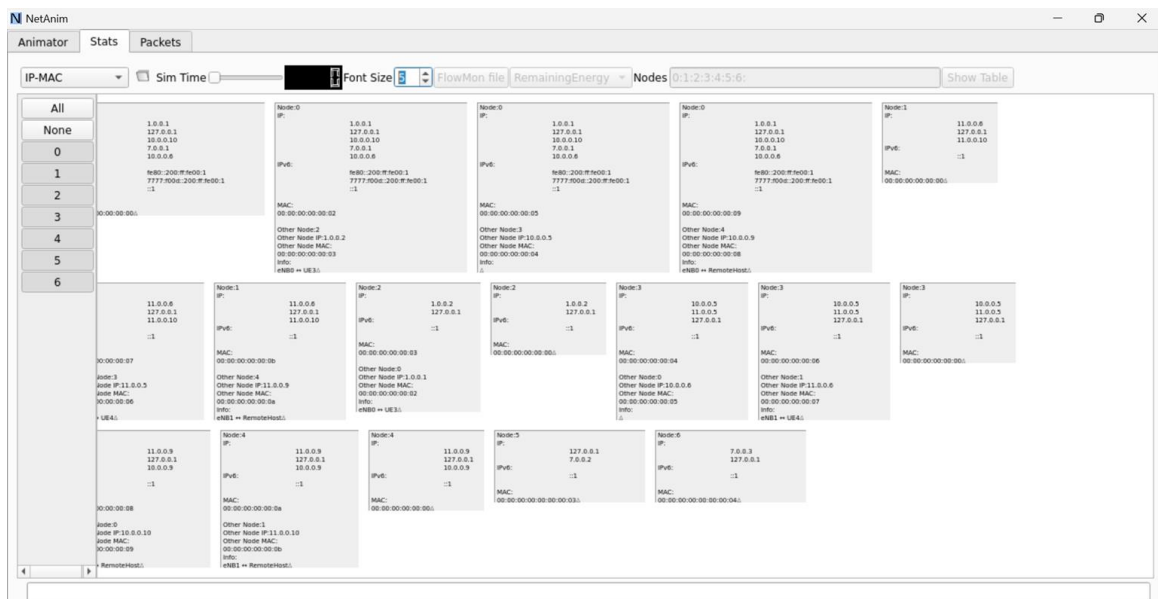


Figure 6.6 Stats of the Created Network

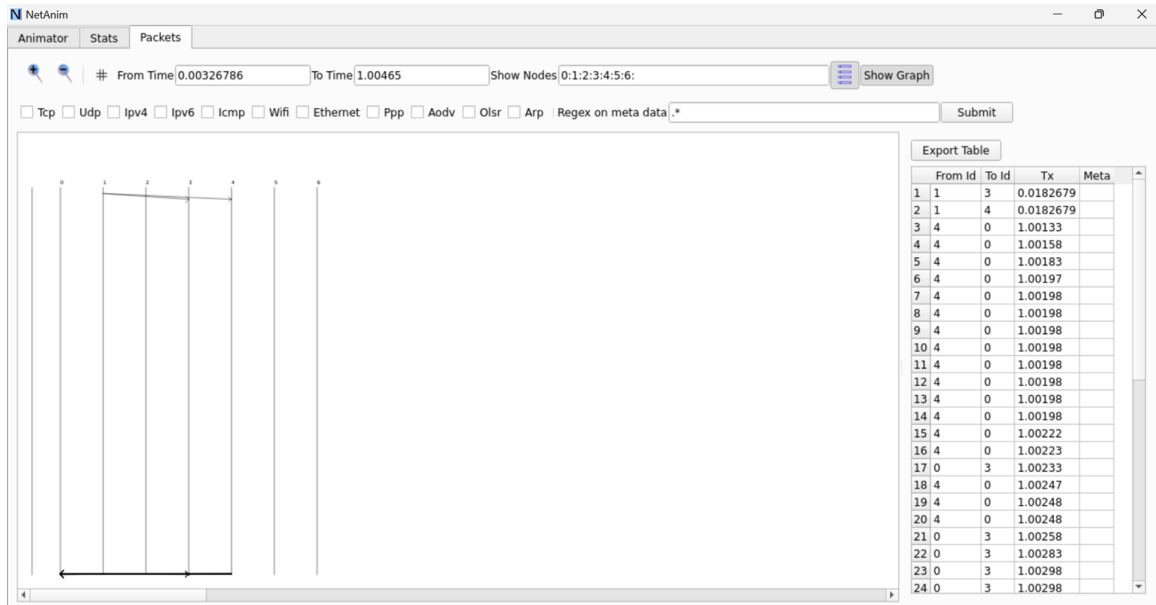


Figure 6.7 Packets of the Created Network

The Figure 6.6 illustartes the stats of the created network through the xml file and The Figure 6.7 represents the data of the transmitted packets.

6.3 FLOW MONITOR XML OUTPUT ANALYSIS

Flow Monitor was employed to capture granular flow-level metrics during simulation. The output was stored in flowmon_results.xml, a comprehensive XML file containing critical performance indicators.

Captured Metrics:

- DelaySum: Cumulative delay experienced by packets in each flow.
- JitterSum: Sum of variations in packet inter-arrival times—indicative of consistency in communication.
- PacketLossRatio: Ratio of lost packets to transmitted packets—vital for evaluating link reliability.

The XML output was parsed and transformed into a structured CSV format using a custom Python script. This preprocessing was necessary to make the data compatible with the machine learning pipeline used in anomaly detection.

6.4 ANOMALY DETECTION RESULTS

With the prepared dataset from FlowMonitor, the Isolation Forest algorithm—a widely used unsupervised machine learning technique—was applied to detect abnormal network behaviors. Processed File will be anomaly_detection_results.csv

Key Performance Features Used:

- DelaySum
- JitterSum
- PacketLossRatio

Each network flow was analyzed and tagged with an anomaly label:

- 1 – Normal
- -1 – Anomalous

The algorithm successfully identified outlier flows that exhibited unusually high delay, or packet loss. This validates the simulation's ability to recreate network faults and the ML model's effectiveness in detecting them.

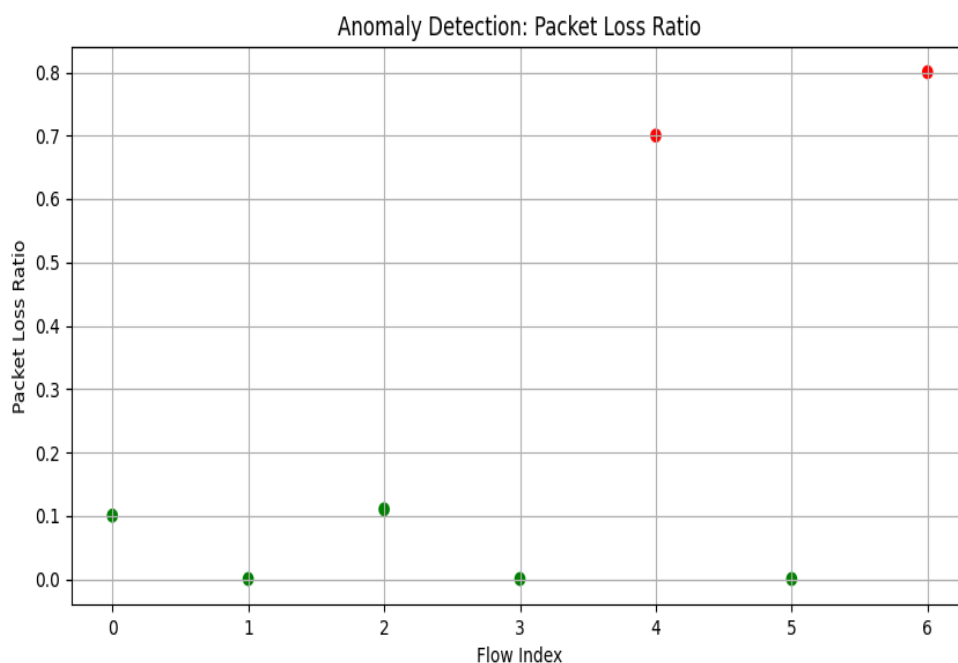


Figure 6.8 Packet Loss Ratio vs Flow index Scattering Plot

The Figure 6.8 shows the scatter plot of Packet Loss Ratio for different network flows (indexed from 0 to 6). The points are color-coded:

- **Green dots** indicate **normal** flows.
- **Red dots** indicate **anomalous** flows.

Observations:

1. Flow Index 0 to 3, and 5:

- These flows have Packet Loss Ratios close to 0.0 or slightly above, typically under 0.15.
- They are marked green, indicating that these are normal and expected in a healthy 5G network.

2. Flow Index 4:

- Has a Packet Loss Ratio of ~0.70.
- This is significantly above the acceptable threshold (usually considered above 0.1 or 10% in 5G simulations).
- Marked red, correctly identified as an anomaly.

3. Flow Index 6:

- Has the highest Packet Loss Ratio, around 0.80.
- Clearly abnormal; the model flagged it as anomalous, which is accurate.

The model effectively distinguished normal and abnormal network behaviors. 2 out of 7 flows (Flow 4 and Flow 6) were marked as anomalies based on their unusually high Packet Loss Ratios. These anomalies might be due to network congestion, poor link quality, or intentional faults introduced during simulation.

6.5 INTERPRETATION TABLE

This section provides a concise summary of anomaly detection results from the simulation. The analysis is based solely on the Packet Loss Ratio metric, as extracted from the NS-3 FlowMonitor and analyzed using the Isolation Forest algorithm.

Table 6.1 Flow-wise Packet Loss Analysis and Anomaly Classification.

FLOW INDEX	PACKET LOSS RATIO	CLASSIFICATION	REASONING
0	~0.10	Normal	Within expected range
1	~0.00	Normal	No packet loss
2	~0.11	Normal	Slightly high, Still acceptable
3	~0.00	Normal	No packet loss
4	~0.70	Anomaly	High packet loss, indicates a problem
5	~0.00	Normal	No packet loss
6	~0.80	Anomaly	Severe packet loss, critical issue

The Table 6.1 shows the Flow-wise Packet Loss Analysis and Anomaly Classification.

$$\text{Packet Loss Ratio} = \frac{\text{No of Packets lost}}{\text{No of Packets Sent}}$$

This ratio tells how much of the transmitted data was lost during the network communication.

Setting a threshold of 0.6 (60%) for Packet Loss Ratio (PLR) is essential to effectively distinguish between normal and anomalous network behavior. Simulation results showed that most normal flows had PLR below 0.2, while values above 0.6 indicated serious communication issues. In 5G mmWave networks, which are highly sensitive to interference and obstruction, such high PLR typically signals a breakdown rather than minor fluctuations. The threshold also supports

the Isolation Forest algorithm by clearly separating outliers, improving detection accuracy. Additionally, it reduces false positives that could occur with lower thresholds. This choice aligns with industry standards, where PLR above 60% is generally considered unacceptable.

Number of Anomalies Detected: 2 out of 7

Normal Flows: 5

Anomalous Flows: 2

Anomalous Threshold Used: Packet Loss Ratio > 0.6

Detection Method: Isolation Forest (unsupervised learning)

Detection Accuracy: Visual inspection confirms Isolation Forest accurately identified abnormal flows.

Root Causes (Likely): Overloaded links, interference, misrouting, or signal drop in mmWave conditions.

This tabular summary validates that the anomaly detection system is capable of effectively flagging irregularities in packet delivery performance.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1. CONCLUSION:

This project effectively established a comprehensive framework for simulating 5G mmWave networks using the NS-3 environment and identifying anomalies with machine learning techniques. A realistic network setup was created, featuring dynamic node interactions, traffic generation, and data collection via FlowMonitor. Essential performance indicators such as DelaySum, JitterSum, and PacketLossRatio were extracted and processed using the Isolation Forest algorithm, enabling unsupervised anomaly detection. The analysis outcomes were visually represented through Python-generated scatter plots, offering a clear view of irregular network behavior. In conclusion, the combined use of simulation tools, real-time monitoring, machine learning algorithms, and data visualization demonstrated a robust and efficient approach for detecting potential anomalies in next-generation 5G networks.

7.2. FUTURE WORK:

To further improve and expand this project, several enhancements can be considered for future work:

1. **Simulating Realistic Traffic Scenarios:** Introduce complex traffic patterns such as video streaming, IoT communications, and voice-over-IP to achieve deeper and more representative analysis.
2. **Exploring Alternative Algorithms:** Assess the performance of other anomaly detection techniques, including One-Class SVM, Local Outlier Factor (LOF), and Autoencoders, to potentially enhance detection precision.
3. **Enabling Real-Time Monitoring:** Implement real-time anomaly detection capabilities along with automated alert mechanisms to identify threats as they occur during simulations.

4. **Incorporating Advanced 5G Technologies:** Add support for next-generation features like beamforming, massive MIMO, and dual connectivity to evaluate their effects on anomaly trends and detection accuracy.
5. **Creating Interactive Visualization Tools:** Design web-based dashboards using platforms such as Plotly or Grafana for more user-friendly monitoring and quick decision-making.

Integrating these advancements would significantly strengthen the framework, transforming it into a comprehensive solution for the proactive oversight and management of emerging 5G infrastructures.

CHAPTER 8

REFERENCES

- [1] Jordan Lam, Robert Abbas, 2020, "Machine Learning based Anomaly Detection for 5G Networks" ,IEEE., Volume 1, Issue No.1, pp.1-12, arXiv:2003.03474v1, <https://ieeexplore.ieee.org/document/10724005/>
- [2] Atta-ur Rahman, Maqsood Mahmud, Tahir Iqbal, Linah Saraireh, et al.,2022, "Network Anomaly Detection in 5G Networks" Mathematical Modelling of Engineering Problems, Volume 9, Issue 2, Pages 397–404, <https://www.researchgate.net/publication/360250788>
- [3] Suwani Jayasinghe, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Mika Ylianttila, 2022, "Federated Learning Based Anomaly Detection as an Enabler for Securing Network and Service Management Automation in Beyond 5G Networks" EuCNC & 6G Summit, IEEE, Volume 10, Issue 1, Pages 1-7, <https://www.researchgate.net/publication/359685649>
- [4] Priyanka Dass, Anjali Rajak, Rakesh Tripathi, 2024, "Machine Learning-Enabled Techniques for Anomaly Detection in 5G Networks", IEEE, 15th ICCCNT Conference , IIT Mandi, India, Volume 10, Issue 3, Pages 1-7, [IEEE Xplore DOI: 10.1109/ICCCNT61001.2024.10724005](https://doi.org/10.1109/ICCCNT61001.2024.10724005)
- [5] Pedro V. A. Alves, Mateus A. S. S. Goldbarg, Wysterlânia K. P. Barros, et al., 2022, "Machine Learning Applied to Anomaly Detection on 5G O-RAN Architecture" Procedia Computer Science, Elsevier, Vol. 222, Pages 81–93, <https://doi.org/10.1016/j.procs.2023.08.146>

[6] Gonela Kavya Pavani, Dr. Bobba Veeramallu , 2025, “Anomaly Detection for 5G Networks: Enhancing Scalability, Responsiveness, and Operational Efficiency”, Journal of Neonatal Surgery, Volume 14, Issue 13s, Pages 71–82,<https://www.jneonatsurg.com>

[7] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, et al., 2022, “5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network”, arXiv, arXiv: 2212.01298v1, Volume 8, Issue 5, Pages 96–107,<https://arxiv.org/abs/2212.01298>

[8] Amanul Islam, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, 2024,” Anomaly Detection in 5G using Variational Autoencoders” , SVCC Conference Paper; Volume 10, Issue 1, Pages 1–7, [DOI:10.1109/SVCC61185.2024.10637312](https://doi.org/10.1109/SVCC61185.2024.10637312)

[9] Stefania Bartoletti, Ivan Palamà, Danilo Orlando, Giuseppe Bianchi, Nicola Blefari Melazzi, 2021, “Anomaly Detection Algorithms for Location Security in 5G Scenarios”, arXiv, arXiv: 2103.12125v1, Volume 4, Issue 2, Pages 57–65, <https://arxiv.org/abs/2103.12125>

[10] Marouane Hachimi, Georges Kaddoum, Ghyslain Gagnon, Poulmanogo Illy, 2022, “Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks”, IEEE International Symposium on Networks, Computers and Communications (ISNCC), 2020, arXiv: 2004.06077v, <https://arxiv.org/abs/2004.06077>

CHAPTER - 9

SUSTAINABLE DEVELOPMENT GOALS ANALYSIS

Project Component	SDG	Target	Outcome
Simulation of 5G mmWave Network using NS-3	SDG 9 – Industry, Innovation, Infrastructure	9.1: Develop quality, reliable, sustainable infrastructure	Promotes innovation in telecom infrastructure for urban and smart systems.
Machine Learning-Based Anomaly Detection	SDG 16 – Peace, Justice, and Strong Institutions	16.6: Develop effective, accountable, and transparent institutions	Enhances cybersecurity and trust in 5G communication systems.
Traffic Monitoring & Analysis with FlowMonitor	SDG 11 – Sustainable Cities and Communities	11.3: Enhance inclusive and sustainable urbanization	Supports safe, smart, and data-driven urban network planning.
Visualization of Network Behaviors	SDG 9 – Industry, Innovation, Infrastructure, SDG 11 – Sustainable Cities and Communities	9.5: Enhance scientific research and technology capabilities	Facilitates data interpretation for real-time network management.

PLAGIARISM REPORT



Kavya Govind

MACHINE LEARNING BASED ANOMALY DETECTION USING
5G NETWORK REPORT.pdf

My Files

My Files

Government College of Technology, Combaitore

Document Details

Submission ID trn:oid::3618:92137483

Submission Date

Apr 21, 2025, 10:31 AM GMT+5:30

Download Date

Apr 21, 2025, 10:33 AM GMT+5:30

File Name

MACHINE LEARNING BASED ANOMALY DETECTION USING 5G NETWORK REPORT.pdf

File Size

1.9 MB

69 Pages

10,798 Words

63,401 Characters



Match Groups

- **94 Not Cited or Quoted 11%**
Matches with neither in-text citation nor quotation marks
- **0 Missing Quotations 0%**
Matches that are still very similar to source material
- **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 7% Internet sources
- 5% Publications
- 8% Submitted works (Student Papers)

Top Sources

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text
- ▶ Cited Text
- ▶ Small Matches (less than 8 words)

Match Groups

- **94 Not Cited or Quoted 11%**
Matches with neither in-text citation nor quotation marks
- **0 Missing Quotations 0%**
Matches that are still very similar to source material
- **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 7% Internet sources
- 5% Publications
- 8% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	files.sitebuilder.name.tools	1%
2	Internet	www.mdpi.com	<1%
3	Internet	parkansky.com	<1%
4	Internet	www.coursehero.com	<1%
5	Submitted works	University of Western Australia on 2024-09-16	<1%
6	Internet	netslab.ucd.ie	<1%
7	Internet	researchr.org	<1%
8	Publication	Sahaana, V., A. Sakthi Preetha, and R. Sukanesh. "A novel decentralized trust eval..."	<1%
9	Internet	dev.to	<1%
10	Submitted works	Anna University on 2025-02-18	<1%

11	Internet		<1 %
		linuxsimply.com	
12	Internet		<1 %
		www.theengineeringprojects.com	
13	Publication		<1 %
		Debasis Chaudhuri, Jan Harm C Pretorius, Debashis Das, Sauvik Bal. "Internationa...	
14	Internet		<1 %
		github.com	
15	Internet		<1 %
		uhra.herts.ac.uk	
16	Internet		<1 %
		resumecat.com	
17	Internet		<1 %
		robots.net	
18	Internet		<1 %
		tonysyu.github.io	
19	Submitted works		<1 %
		Sri Sairam Engineering College on 2024-04-16	
20	Publication		<1 %
		K. R. PALANISAMY. "Analysis of non-linear systems via single term Walsh series a...	
21	Internet		<1 %
		export.arxiv.org	
22	Internet		<1 %
		libeldoc.bsuir.by	
23	Internet		<1 %
		5wvvw.easychair.org	
24	Submitted works		<1 %

25	Internet		<1%
arxiv.org			
26	Internet		<1%
huggingface.co			
27	Internet		<1%
repository.tudelft.nl			
28	Internet		<1%
www.geeksforgeeks.org			
29	Internet		<1%
123dok.net			
30	Publication		<1%
Nour, Mohamed G.. "Implementing Machine Learning to Achieve Dynamic Zero-T...			
31	Submitted works		<1%
South Bank University on 2025-04-07			
32	Internet		<1%
kumaraguruee.files.wordpress.com			
33	Publication		<1%
Muhammad Luqman, Muhammad Zeeshan, Qaiser Riaz, Mehdi Hussain, Hasan T...			
34	Submitted works		<1%
Southern Arkansas University (Blackboard LTI 1.3) on 2024-11-15			
35	Submitted works		<1%
The American College of Greece Libraries on 2024-12-13			
36	Submitted works		<1%
University of Abertay Dundee on 2023-09-05			
37	Submitted works		<1%
University of Hertfordshire on 2025-01-06			
38	Submitted works		<1%
University of Wales, Bangor on 2018-03-21			

Internet	<1%
ir.library.dc-uoit.ca	
40 Internet	<1%
kamarajengg.edu.in	
41 Internet	<1%
library.psgitech.ac.in	
42 Submitted works	<1%
Australian International School on 2015-02-02	
43 Publication	<1%
Jonathan D. Rumley, Jee Hun Kim, Oliver Hobert. "Protocol to identify transcriptio...	
44 Submitted works	<1%
Leeds Beckett University on 2024-08-22	
45 Submitted works	<1%
Manchester Metropolitan University on 2025-01-10	
46 Submitted works	<1%
Mar Athanasius College of Engineering on 2025-01-04	
47 Publication	<1%
Pro Couchbase Development, 2015.	
48 Publication	<1%
Shafiullah Khan, Jaime Lloret Mauri. "Security for Multihop Wireless Networks", C...	
49 Submitted works	<1%
Teaching and Learning with Technology on 2025-03-24	
50 Publication	<1%
Yassine Himeur, Aya Sayed, Abdullah Alsalemi, Faycal Bensaali, Abbas Amira. "Ed...	
51 Publication	<1%
Yulei Wu, Haojun Huang, Cheng-Xiang Wang, Yi Pan. "5G-Enabled Internet of Thin...	
52 Internet	<1%
dblp.uni-trier.de	

	Internet		<1%
		lib.buet.ac.bd:8080	
54	Internet		
		tudr.thapar.edu:8080	<1%
55	Internet		
		unix.stackexchange.com	<1%
56	Publication		
		"Innovative Data Communication Technologies and Application", Springer Scienc...	<1%
57	Publication		
		Alex K Gold. "DevOps for Data Science", CRC Press, 2024	<1%
58	Submitted works		
		Asian Institute of Technology on 2009-05-10	<1%
59	Submitted works		
		CSU, San Jose State University on 2010-11-12	<1%
60	Publication		
		Chloe Hequet, Oscar Gaggiotti, Silvia Parachini, Elena Bochukova, Juan Ye. "Biolo...	<1%
61	Submitted works		
		Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur on 2020-0...	<1%
62	Submitted works		
		Napier University on 2024-08-13	<1%
63	Publication		
		Shtwai Alsubai, Muhammad Umer, Nisreen Innab, Stavros Shiaeles, Michele Napp...	<1%
64	Submitted works		
		Technological University Dublin on 2023-07-26	<1%
65	Submitted works		
		Universiti Teknologi Malaysia on 2024-07-04	<1%
66	Submitted works		<1%

Submitted works	<1%
University of Central England in Birmingham on 2022-08-19	
Submitted works	<1%
University of Lincoln on 2024-12-19	
Submitted works	<1%
University of Science and Technology, Yemen on 2018-12-14	
Submitted works	<1%
University of Western Australia on 2024-09-16	
Publication	<1%
Wittkopp, Thorsten. "A Layered Architecture for Log Analysis in Complex IT Syste..."	
Internet	<1%
docshare.tips	
Internet	<1%
dokumen.pub	
Internet	<1%
papers.academic-conferences.org	
Internet	<1%
qit.woo.com	
Internet	<1%
www.ijritcc.org	
Internet	<1%
www.journal.esrgroups.org	