Scientific Research Publishing

Journal of Information Security, 2025, 16(4), 472-499
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

5G Network Vulnerabilities: A Security Mechanism for Detecting and Blocking DDoS Threats at the Network Edge

Sakib Mahmud, Ahsan Ullah, Shakhawat Hossain Shipon, Mahedi Hassan, Md Nazmus Sakib

Department of Computer Science and Engineering, World University of Bangladesh, Dhaka, Bangladesh
Email: sakib.jpeg@gmail.com, ahsan.ullah@cse.wub.edu.bd, shakhawathossainshipon@gmail.com, mahedi7171@gmail.com, nazmus_sakib70@yahoo.com

How to cite this paper: Mahmud, S., Ullah, A., Shipon, S.H., Hassan, M. and Sakib, M.N. (2025) 5G Network Vulnerabilities: A Security Mechanism for Detecting and Blocking DDoS Threats at the Network Edge. *Journal of Information Security*, **16**, 472-499.
https://doi.org/10.4236/jis.2025.164024

Received: June 3, 2025
Accepted: August 23, 2025
Published: August 26, 2025

Abstract

The rapid expansion of 5G networks has revolutionized global connectivity, enabling billions of devices to communicate seamlessly across various industries. However, this advancement has also increased the vulnerability to Distributed Denial of Service (DDoS) attacks, posing significant threats to network reliability. This research presents a novel machine learning-based approach for detecting and mitigating DDoS attacks at the Multi-Access Edge Computing (MEC) layer, with the objective of enhancing the security and efficiency of 5G ecosystems. The proposed system integrates Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost algorithms with the Zeek Intrusion Detection System (IDS) to enable real-time traffic classification and mitigation at the network edge. Models were trained using the CIC-DDoS2019 dataset to identify realistic attack patterns. Python was used for implementation, and Zeek IDS dynamically extracted traffic features. Simulated traffic streams, blending both benign and malicious behaviors, were employed to evaluate system performance under realistic conditions. The architecture leverages separate inbound and outbound switches to isolate traffic flows, enabling immediate blocking of malicious packets and blacklisting of source IPs while maintaining uninterrupted service for legitimate traffic. The proof-of-concept demonstrated the ability to detect and mitigate 40% of malicious traffic effectively. Despite its success, the system currently relies on simulated data and lacks a complete deployment-ready software package. Future work will focus on building a robust, scalable implementation suitable for real-world MEC environments. This research provides a promising foundation for protecting 5G networks from evolving DDoS threats without compromising le-

gitimate network operations.

Keywords

5G Networks, DDoS Attacks, Network Security, Multi-Access Edge Computing (MEC), Intrusion Detection System (IDS), Machine Learning, Network Edge, Threat Mitigation

1. Introduction

The fifth generation of mobile networks, commonly known as 5G, is revolutionizing the way devices communicate. Unlike earlier generations that primarily connected mobile phones, 5G extends connectivity to a vast array of devices, including smart home appliances, healthcare systems, financial platforms, and transportation networks (Techopedia, n.d.; IBM, n.d.). While this level of interconnectivity enables innovative use cases and enhances efficiency, it also introduces significant security vulnerabilities. Devices on 5G networks are often always online, making them susceptible to exploitation if robust security measures are not in place. This pervasive connectivity creates a new attack surface for malicious activities, including Distributed Denial of Service (DDoS) attacks [1].

A DDoS attack involves hundreds to thousands of compromised devices, often orchestrated as part of a botnet, overwhelming a target server or network with excessive traffic. These attacks can slow down or completely disrupt services, posing severe risks in highly connected 5G environments where uptime and performance are critical. As 5G adoption accelerates, addressing these threats becomes essential to safeguarding sensitive data and ensuring uninterrupted network functionality [2].

This research investigates vulnerabilities in the 5G ecosystem that enable attackers to exploit devices for botnets and launch DDoS attacks. Unlike conventional methods that focus on rate limiting or IP filtering, this study introduces a novel security mechanism that leverages Multi-Access Edge Computing (MEC) and strategically placed switches to manage inbound and outbound traffic. The proposed approach targets malicious traffic originating from compromised end-user devices, detecting and blocking threats closer to their source.

To enhance detection accuracy, machine learning algorithms like Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost models are utilized to analyze and identify DDoS attack patterns. The CIC-DDoS2019 dataset, a comprehensive and publicly available benchmark dataset, is used to train and validate the model. By integrating ML-powered analysis with MEC and switch-based traffic management, this mechanism continuously monitors data streams and network activity, detecting anomalies or traffic surges indicative of DDoS attacks. This innovative solution not only ensures real-time protection for endpoints across the 5G network but also strengthens the overall resilience of the communication infrastructure against evolving threats.

1.1. Objectives

The objectives of this study are as follows:

1) To identify and analyze weaknesses within the 5G ecosystem that allow devices to be misused as botnets for Distributed Denial of Service (DDoS) attacks at Edge Network.

2) To propose a stronger security system, focused on real-time monitoring and blocking, to reduce risks for devices connected to 5G.

1.2. Justification

As 5G becomes a key part of daily life, protecting it from cyber threats is essential for public safety and data security. Devices, due to their constant connection and limited security features, are especially at risk of being used in harmful botnets. Such botnets can cause serious harm, like large-scale DDoS attacks that overwhelm systems and disrupt services or gain unauthorized access to sensitive data (Allot, n.d.). This research is timely, as many existing 5G security measures focus on general security issues but may overlook specific threats like DDoS attacks. By enhancing security at the network's edge, we can better prevent unauthorized access and maintain trust in 5G technology, which is essential as it supports critical applications. Securing the network will bring significant positive impacts [3].

The Common Vulnerability Scoring System (CVSS) helps cybersecurity experts rate how dangerous vulnerabilities are. It gives a score from 0.0 (not severe) to 10.0 (very severe) based on how much damage it can cause and how easy it is to exploit. Using these scores helps experts focus on the most serious problems first.

The recent Common Vulnerability and Exposure (CVE) score of DDoS attacks (Table 1):

Table 1. Severe CVE scores of DDoS vulnerabilities (2024).

CVE ID	Score
CVE-2024-34567	9.9
CVE-2024-23456	8.7

The increasing frequency and sophistication of Distributed Denial-of-Service (DDoS) attacks present significant challenges to protecting critical infrastructure, highlighting the need for advanced and targeted defense mechanisms. This research, titled "***5G Network Vulnerabilities: A Security Mechanism for Detecting and Blocking DDoS Threats at the Network Edge***", introduces a novel approach to addressing these challenges. Unlike traditional methods such as rate limiting or IP-based filtering, which are insufficient for handling the evolving nature of modern DDoS attacks, this study focuses on detecting and blocking malicious traffic originating from end users. By leveraging Multi-Access Edge Computing (MEC) and strategically managing inbound and outbound traffic through different switches, the proposed solution enhances the ability to mitigate DDoS threats

closer to their source. This approach not only ensures a more efficient and accurate detection mechanism but also strengthens the overall security and resilience of the network infrastructure against such threats.

1.3. Scope of Study

This research looks into how to find and stop Distributed Denial of Service (DDoS) attacks in 5G networks. It focuses on protecting the edge of the network. The goal is to create a new security system that uses Multi-Access Edge Computing (MEC) and switches placed in important spots to check and manage internet traffic coming in and out of the network. The research specifically looks at traffic from compromised end-user devices to deal with a major weakness in 5G networks: the misuse of interconnected devices to launch DDoS attacks.

A key part of this research is the use of Zeek, a powerful tool for analyzing network traffic. Zeek is combined with machine learning algorithms at the MEC layer. This helps to find and analyze bad traffic in real-time. The setup makes sure that incoming and outgoing traffic are kept separate using different switches. This stops traffic from mixing and makes it easier to find and handle unusual things. This separation is expected to make stopping DDoS attacks more efficient and accurate.

The research uses a few different machine learning models, including Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost, to find attacks. These models are taught and tested using something called the CIC-DDoS2019 dataset. This dataset is good for copying real-world DDoS attack situations and testing the suggested solution.

Key Areas of Focus

1) 5G Network Vulnerabilities: Identifying architectural and operational weaknesses in 5G that make it prone to DDoS attacks.

2) Edge-Based Traffic Management: Implementing MEC with Zeek to monitor and manage traffic at the network edge.

3) ML-Powered Detection: Utilizing Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost models to classify anomalous traffic patterns indicative of DDoS attacks.

4) Traffic Segregation: Ensuring inbound and outbound traffic is handled on separate switches to prevent cross-contamination and improve detection precision.

5) Dataset Utilization: Using the CIC-DDoS2019 dataset to develop and validate the detection mechanism.

6) Real-Time Protection: Providing continuous monitoring, anomaly detection, and rapid mitigation of threats in a dynamic 5G environment.

Exclusions

The study does not include:

- Broader 5G security aspects such as encryption, privacy, or user authentication.
- The machine learning models were trained and tested independently using Py-

thon code to ensure their accuracy in identifying DDoS traffic. These models were not directly integrated into Zeek.

- General optimization of network performance unrelated to DDoS detection or prevention.

This research contributes to the field of 5G security by introducing a scalable, efficient, and ML-driven edge-based solution that effectively counters DDoS threats, ensuring the resilience and safety of critical network infrastructure.

2. Literature Review

Recent research on Distributed Denial of Service (DDoS) threats within 5G networks and existing solutions will be reviewed. This review aims to highlight the limitations in current methods and set the foundation for the novel security mechanism proposed in this study. By addressing gaps in scalability, real-time traffic management, and endpoint-specific security, this research introduces a tailored mechanism that enhances 5G network defense against DDoS threats, particularly by focusing on malicious traffic originating from end-user devices.

2.1. Review of Existing Solutions

Onoja *et al.* [4], in their study DDoS Threats and Solutions for 5G Networks, present an overview of various DDoS protection strategies. Their work emphasizes Software-Defined Networking (SDN) for centralized traffic management and edge computing for localized detection. While these methods isolate some DDoS patterns, they are limited in their ability to scale across diverse 5G ecosystems, where a wide range of endpoints, from IoT devices to smart appliances, are continuously online. The study also lacks emphasis on long-term botnet monitoring and adaptive attack detection, leaving gaps in addressing threats from botnets that evolve over time or remain dormant before initiating coordinated attacks.

Similarly, studies such as Ghorbani *et al.* [5] and Serrano Mamolar *et al.* [6] rely on static rule-based mechanisms to identify DDoS patterns. These approaches are effective for predictable attack behaviors but struggle to detect advanced threats that leverage stealth or adaptive methods. Such limitations underscore the need for a more dynamic, endpoint-focused solution that addresses the vulnerabilities posed by the always-on connectivity inherent in 5G networks.

Sanmorino and Yazid [7] proposed the use of flow patterns to detect and block malicious traffic. While their methodology demonstrated success in detecting certain DDoS traffic characteristics, it heavily relied on predefined traffic patterns, which are inadequate for detecting highly adaptive and evolving attack vectors. Furthermore, cluster-based analysis methods for DDoS detection are time-consuming and often less precise, making them unsuitable for real-time applications in dense 5G deployments.

2.2. Proposed Improvements

To address these limitations, this study focuses on a novel security mechanism

that leverages Multi-Access Edge Computing (MEC) and the strategic use of separate switches for inbound and outbound traffic. Unlike existing solutions that often mix traffic or rely solely on centralized analysis, this approach ensures better segregation and management of network flows. By preventing the mixing of inbound and outbound traffic at the switch level, the proposed mechanism enables more precise detection of malicious traffic.

This study also incorporates Zeek, a sophisticated tool for analyzing network traffic, to improve real-time detection capabilities at the MEC layer. The proposed solution uses machine learning algorithms like Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost, trained on the CIC-DDoS2019 dataset. This dataset provides a variety of traffic features, such as "*Flow Duration*", "*Total Fwd Packets*", "*Total Backward Packets*", "*Fwd Packet Length Max*", "*Bwd Packet Length Max*", "*Flow IAT Mean*", "*Flow IAT Std*", "*Flow IAT Max*", "*Fwd Packets/s*", *and* "*Bwd Packets/s*", which are used to identify unusual patterns in traffic flows. Although Zeek is not directly integrated with the machine learning models in this study, future work should aim to integrate these components for enhanced real-time detection. This integration will ensure that threats are detected proactively, even as attack methods change.

2.3. The Novelty of the Proposed Solution

This study's novelty lies in its focus on endpoint-specific DDoS mitigation within 5G networks by combining MEC-based traffic analysis with ML-driven detection mechanisms. Unlike traditional methods, which often treat endpoints as secondary to overall network security, this mechanism positions endpoints as primary points of analysis. The always-on nature of 5G-connected devices makes them prime targets for botnet formation and subsequent DDoS attacks, and this solution is specifically designed to address that vulnerability.

The use of separate switches for inbound and outbound traffic further enhances the precision of detection, as it prevents the mixing of legitimate and malicious traffic, enabling faster response times. By deploying this mechanism at the MEC layer, threats can be mitigated closer to their source, reducing the strain on the central network and improving scalability in dense 5G environments.

This research introduces a robust framework for detecting and blocking DDoS attacks through continuous monitoring, malicious traffic eradication and containment ensuring both adaptability and scalability. By leveraging the CIC-DDoS2019 dataset and integrating Zeek at the edge, the study establishes a new benchmark for real-time 5G network defense, addressing gaps in prior research and paving the way for more resilient and secure 5G infrastructures.

3. Methodology

This methodology was adopted to design, implement, and evaluate the proposed security mechanism for detecting and mitigating Distributed Denial of Service (DDoS) attacks within 5G networks. The approach follows a structured incident

response framework inspired by the SANS Incident Response Process, consisting of six phases: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned (Figure 1). The methodology integrates Multi-Access Edge Computing (MEC) with strategically isolated switches for inbound and outbound traffic management to ensure precise threat detection.

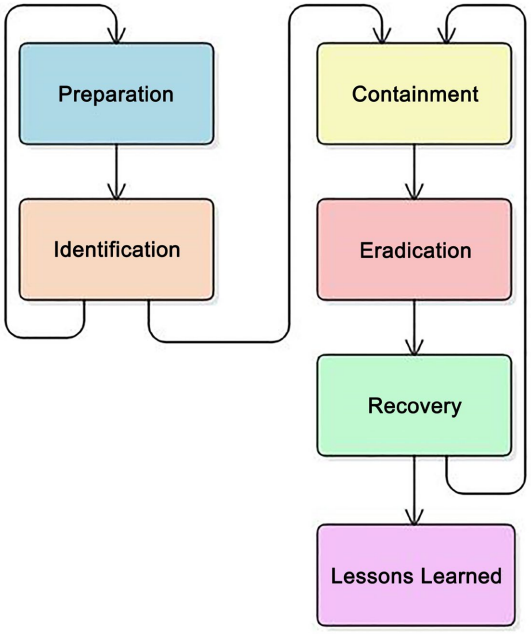

Figure 1. The SANS institute incident response cycle.

Additionally, the proposed system employs Zeek for traffic analysis and an AI-based Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost model trained on the CIC-DDoS2019 dataset to identify abnormal patterns and block malicious traffic. Traffic flows are continuously monitored at the network edge, ensuring real-time identification of threats originating from compromised end-user devices. The incident response process allows for systematic detection, containment, and mitigation of DDoS attacks, ensuring minimal disruption to the network. By leveraging edge computing and AI, this methodology provides a scalable, efficient, and proactive solution for strengthening 5G network security.

Description of Methodology

The proposed methodology is justified as it addresses the challenges of 5G networks by detecting and mitigating Distributed Denial of Service (DDoS) attacks at the network edge. This approach leverages Multi-Access Edge Computing (MEC), which allows for real-time traffic monitoring and analysis closer to the source of malicious activity. By incorporating tools such as Zeek for traffic analysis and machine learning models—Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost—the system is capable of identifying malicious traffic patterns with high accuracy. The methodology further enhances detection efficiency

by isolating inbound and outbound traffic using separate switches, ensuring precise control over network flows. Following the structured SANS Incident Response Process, the proposed system ensures a systematic approach to threat detection, containment, and mitigation, minimizing disruption to the 5G network infrastructure.

The following covers the details about the methodology chosen:

- **Preparation:** The preparation phase lays the foundation for the security mechanism by designing a robust strategy and implementing proactive measures. Multi-Access Edge Computing (MEC) is used to process traffic closer to its origin, enhancing responsiveness. Traffic flows are segregated using separate switches for inbound and outbound traffic, preventing mixing and ensuring better anomaly detection. Zeek is integrated to monitor and analyze traffic in real-time, while machine learning models (RF, KNN, XGBoost) are used to classify traffic patterns. The selected features for analysis are as follows:

1) Flow Duration: Measures the time duration of a traffic flow. Malicious traffic often exhibits longer or irregular durations compared to normal traffic, making this feature critical for identifying anomalies.

2) Total Forward Packets: Counts the total number of packets sent in the forward direction during a flow. DDoS traffic often includes an unusually high number of packets, which can help distinguish malicious flows.

3) Total Backward Packets: Counts the total number of packets sent in the reverse direction. This feature complements the forward packet count to provide a complete picture of traffic flow behavior.

4) Forward Packet Length Max: Captures the maximum length of packets in the forward direction. DDoS attacks may include large packets to overwhelm the target, making this feature valuable.

5) Backward Packet Length Max: Captures the maximum length of packets in the reverse direction. Similar to the forward packet length, it aids in identifying abnormal traffic patterns.

6) Flow Inter-Arrival Time (IAT) Mean: Measures the average time between packets in a flow. Malicious traffic often exhibits irregular inter-arrival times compared to legitimate traffic.

7) Flow Inter-Arrival Time (IAT) Standard Deviation: Quantifies the variability in inter-arrival times. High variability can indicate irregular traffic patterns typical of DDoS attacks.

8) Flow Inter-Arrival Time (IAT) Max: Captures the maximum inter-arrival time within a flow. This feature provides additional insight into traffic irregularities.

9) Forward Packets per Second (Fwd Packets/s): Calculates the rate of packets sent in the forward direction. DDoS traffic often involves a high packet rate, making this feature crucial.

10) Backward Packets per Second (Bwd Packets/s): Calculates the rate of packets sent in the reverse direction. It complements the forward packet rate to identify

anomalous patterns.

These features were chosen because they collectively capture the characteristics of network traffic, including volume, timing, and flow behavior. Such metrics are critical for differentiating between legitimate and malicious traffic in real-time. This comprehensive preparation ensures the network is equipped to handle DDoS threats effectively.

- **Identification:** In this phase, continuous monitoring of network traffic is conducted to detect anomalies. Zeek is deployed at the MEC layer to analyze traffic in real-time. The selected features are extracted from the traffic flows and fed into the machine learning models. The models—trained on the CIC-DDoS2019 dataset—identify patterns indicative of DDoS attacks, such as sudden increases in packet rates or irregular inter-arrival times. The segregation of inbound and outbound traffic using separate switches further enhances detection accuracy, enabling the system to pinpoint malicious traffic swiftly.
- **Containment:** In the containment phase, the focus is on stopping the spread of harmful traffic while ensuring that normal network operations remain unaffected. The key approach here is to separate good and bad traffic using different switches for better control. Traffic coming from the internet (inbound) and traffic generated by end-user devices toward the core network (outbound) are handled through separate switches, ensuring there is no mixing of traffic. For cases where a single switch is used for a base station (BTS), separate VLANs are configured for inbound and outbound traffic, further isolating data flows.

At the Multi-Access Edge Computing (MEC) layer, the AI-trained detection program runs continuously to analyze and classify the traffic. The Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost model detects anomalies and determine whether the traffic is good or bad. Once harmful traffic is identified, it is contained using the outbound switch, preventing it from reaching the core network. This approach ensures that malicious traffic is blocked at the edge, minimizing the risk of disruption to the 5G network while allowing legitimate traffic to flow smoothly.

- **Eradication:** The eradication phase focuses on removing the source of harmful traffic and ensuring that the network is clean and free from DDoS threats. Once the malicious traffic is detected by the Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost models at the MEC layer, it is blocked and isolated using the outbound switch. This ensures that bad traffic cannot travel further into the core network. At the same time, additional steps are taken to identify the source IP addresses or compromised end-user devices generating the attack.

By leveraging Zeek for deep traffic analysis, detailed logs are generated to pinpoint the root cause of the DDoS attack. If a network provider uses a single switch with VLANs for traffic separation, the compromised VLAN segment can be isolated, and the bad traffic can be purged. Once the source is identified, corrective

measures are applied to remove or neutralize the malicious devices, such as shutting down affected endpoints, updating security rules, or blacklisting the source IPs.

This process ensures that the network is not only protected from immediate threats but also cleaned thoroughly, reducing the chances of future attack recurrence.

- **Recovery:** The recovery phase focuses on restoring normal operations in a controlled and secure manner. A specific time and date are defined for resuming services, and the final decision is made by system owners based on recommendations from the Cybersecurity Incident Response Team (CSIRT). Before systems go live, extensive testing and verification are carried out to ensure that they are clean, fully functional, and free of any lingering threats. Once services are restored, continuous monitoring is performed to detect any abnormal behaviors or signs of residual issues. This monitoring phase ensures that the network remains stable and secure. Additionally, proactive measures are implemented on the restored systems to prevent a recurrence of the same incident. By applying these steps, the recovery process ensures the 5G network returns to optimal operation while maintaining resilience against future DDoS attacks.
- **Lessons Learned:** The lessons learned phase is crucial for improving future incident response processes and strengthening the overall security framework. No later than two weeks after the incident, all relevant information is gathered and reviewed by the Cybersecurity Incident Response Team (CSIRT). The primary goal is to document the incident thoroughly and extract key lessons that will help prevent similar issues in the future.

Comprehensive documentation is completed, ensuring that all aspects of the incident, including events, actions taken, and challenges faced, are clearly recorded. An incident report is prepared, offering a detailed, step-by-step analysis of the incident while answering critical questions: Who, What, Where, Why, and How. This report helps identify areas where the response was effective and where improvements are needed.

Additionally, Zeek's capability to log details of the attack can be analyzed to identify patterns of malicious traffic. These logs provide insights into the nature of the attack and enable the development of preemptive measures to mitigate similar threats in the future. By applying these lessons, the incident response process is refined, ensuring better preparedness and stronger defense against future DDoS threats within the 5G network.

The CSIRT uses the report to identify specific ways to improve team performance by highlighting any issues that were not handled efficiently. Metrics derived from the incident, such as response time or detection accuracy, are established as benchmarks for future comparisons. Finally, a lessons learned meeting is conducted with the CSIRT team and key stakeholders to discuss findings and implement improvements immediately. By applying these lessons, the incident

response process is refined, ensuring better preparedness and stronger defense against future DDoS threats within the 5G network.

4. Requirement Analysis, Design & Developments

4.1. Requirement Gathering Technique

This chapter presents the key requirements, design considerations, and development details of the proposed system to detect and mitigate DDoS threats in 5G networks using edge computing. It defines the functional and non-functional requirements necessary for implementing the solution, emphasizing the capabilities and limitations of the proposed methodology.

4.2. Functional Requirements

Functional requirements define the essential tasks and behaviors that the system must perform to achieve its objectives. In this research, the focus is on detecting and mitigating DDoS attacks using Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost as the machine learning model, while leveraging Multi-Access Edge Computing (MEC) and network traffic separation strategies.

The functional requirements are as follows:

- **Dataset:** A reliable dataset is a primary requirement for this research. The CIC-DDoS2019 dataset, curated by Dr. Iman Sharafaldin, Dr. Saqib Hakak, Dr. Arash Habibi Lashkari, and Dr. Ali Ghorbani, has been used. This dataset, obtained from Kaggle, serves as a foundation for the analysis of real-world DDoS attack traffic, enabling accurate training and validation of our machine learning models.
- **Python Programming Language**: Python serves as the core programming language for this research. Its flexibility and versatility allow for efficient pre-processing of data, development of machine learning models, and analysis of results. The simplicity and broad support of Python make it ideal for implementing an intrusion detection system (IDS).
- **Computing Platform:** A secure and high-performance computing environment is required to execute the machine learning tasks efficiently. Experiments are conducted in platforms such as Google Colab to ensure collaborative development and seamless resource allocation. The computing system must support large-scale data analysis and machine learning model training.
- **Python Libraries**: The research leverages several Python libraries to streamline data processing and model development. Libraries like Pandas are used for data manipulation, NumPy for numerical computations, and TensorFlow for deep learning model implementation. These libraries support pre-processing, feature extraction, model training, and evaluation, ensuring the system operates efficiently in detecting DDoS attacks.
- **Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost Classifiers:** These machine learning models form the core of the detection mechanism. Trained on the CIC-DDoS2019 dataset, they analyze traffic flow features to

identify patterns indicative of DDoS attacks. By accurately classifying data as either benign or malicious, these models ensure reliable detection of threats.

- **MEC-Based Network Traffic Management:** Traffic analysis and DDoS mitigation are performed at the Multi-Access Edge Computing (MEC) layer. Inbound and outbound traffic are strictly separated using dedicated switches to avoid mixing good and bad traffic. For single-switch deployments, VLANs are configured to isolate traffic flows. This design improves the precision of traffic monitoring and ensures quicker detection and containment of malicious traffic.
- **Zeek IDS Integration**: Zeek IDS is deployed at the MEC layer to monitor and analyze real-time network traffic. It detects suspicious traffic patterns, anomalies, and potential DDoS indicators, enabling proactive identification of threats before they reach the core network.

4.3. Non-Functional Requirements

Non-functional requirements focus on the overall quality, performance, and usability of a system rather than its core functions. These attributes are vital to ensure the Intrusion Detection System (IDS) for DDoS attacks operates effectively and remains practical in real-world 5G environments.

The non-functional requirements for this research include:

- **Responsiveness:** The system must detect and react to DDoS threats swiftly, minimizing delays in identifying and mitigating harmful traffic.
- **Precision:** The solution should deliver reliable results with minimal errors, ensuring that legitimate traffic is not mistakenly flagged as malicious.
- **Adaptability:** The system must adjust to varying network loads and dynamic traffic patterns without losing efficiency, especially as 5G networks expand.
- **Data Integrity:** It is crucial to ensure the protection and accuracy of network data, preventing unauthorized alterations or compromises during monitoring and analysis.
- **Resilience:** The system must maintain stability and functionality under high-pressure conditions, such as during heavy traffic loads or sustained attacks.
- **Accessibility:** The solution must be designed with simplicity in mind, making it easy for network administrators to deploy, manage, and maintain without requiring specialized skills.

By addressing these non-functional requirements, the system not only fulfills its purpose but also ensures seamless integration into real-world scenarios, offering a reliable and scalable defense against DDoS attacks in the 5G network environment.

4.4. Design

The design of the proposed system focuses on detecting and mitigating DDoS attacks originating from compromised end-user devices within the 5G network. The system leverages Multi-Access Edge Computing (MEC), Zeek Intrusion Detection

System (IDS), and a Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost-based detection model to analyze and filter malicious traffic. The core principle of the design ensures that legitimate and malicious traffic are separated, with a clear focus on precision and efficiency at the network edge.

The design components and flow of the system are as follows:

1) Malicious Traffic from End-User Devices: The traffic originates from end-user devices, which may be compromised and used to generate DDoS attacks. This malicious traffic is transmitted through the 5G Gateway and 5G Base Station (BTS) to the network.

2) Traffic Segregation at Switch IN: All traffic, both legitimate and malicious, is directed to Switch IN. This switch serves as the entry point for analyzing traffic flows.

3) Traffic Analysis at MEC Layer: At the Multi-Access Edge Computing (MEC) layer, the traffic is sent for analysis. The MEC is equipped with the Zeek IDS and the Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost detection models trained on the CIC-DDoS2019 dataset.

- Zeek IDS monitors traffic in real-time and extracts relevant flow features (e.g., packet rates, source IPs, protocol types) for analysis.
- The Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost model processes these features to identify traffic patterns and classify them as either legitimate or malicious.

4) Traffic Filtering and Containment:

- Traffic identified as legitimate is passed to Switch OUT. This switch ensures that the clean, legitimate traffic is forwarded to the core network and eventually to the internet.
- Malicious traffic is contained and blocked at the MEC layer, preventing it from reaching the core network. This containment reduces the risk of service disruption and safeguards the network's performance.

5) Traffic Isolation: To ensure precision, inbound and outbound traffic are managed using separate switches (Switch IN and Switch OUT). This prevents the mixing of good and bad traffic, enhancing detection accuracy and improving containment efficiency. In cases where a single switch is used for a Base Station (BTS), VLANs are configured to separate inbound and outbound traffic flows.

Design Benefits:

- The use of MEC ensures real-time analysis and containment of DDoS traffic closer to its source, reducing latency and improving efficiency.
- Zeek IDS combined with the Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost model allows for accurate detection of malicious traffic.
- The separation of traffic through dedicated switches or VLANs prevents congestion and ensures that legitimate traffic flows smoothly to the core network.

This design (**Figure 2**) ensures a robust, scalable, and efficient system for mitigating DDoS attacks, protecting the 5G network from disruptions caused by malicious end-user devices.

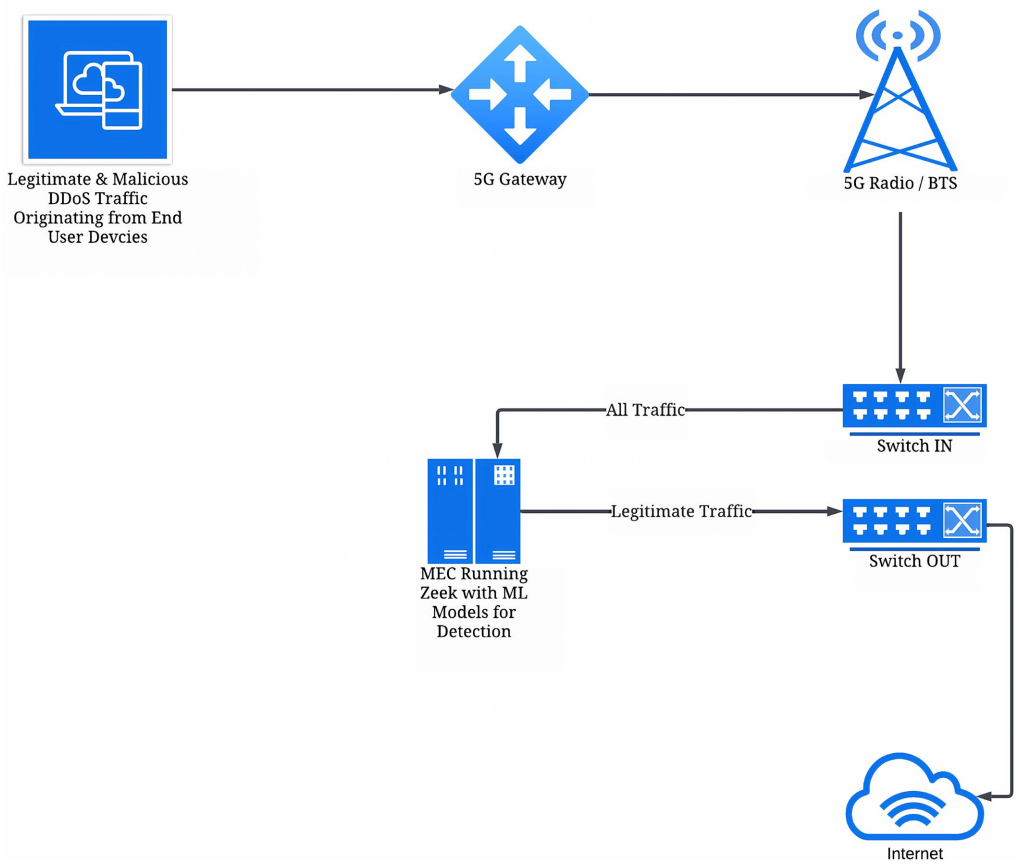

Figure 2. System design for DDoS detection and mitigation in a 5G network.

4.4.1. Use Case Analysis of DDoS Attacks

A DDoS attack using a botnet is a coordinated cyber assault that aims to disrupt the normal functioning of a target server or network. In this type of attack, a botnet a network of compromised devices, often called "bots" is controlled by a central entity, such as a hacker or malware operator. Once an attack is initiated, these infected devices generate and send an overwhelming amount of traffic toward the target system, as illustrated in Figure 3.

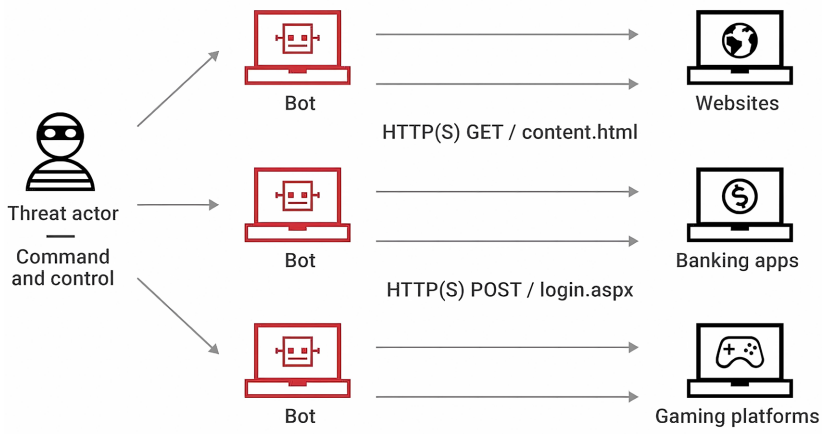

Figure 3. DDoS attack (Source: Akamai).

The primary objective of a DDoS attack is to overwhelm the target's resources, such as bandwidth, processing power, or memory, ultimately rendering services inaccessible to legitimate users. Botnets can include thousands or even millions of infected devices, amplifying the scale and severity of the attack. The consequences of such attacks are significant, often causing financial losses, operational disruptions, and reputational harm. These impacts make DDoS attacks a serious challenge for network administrators and security professionals, particularly within highly connected environments like 5G networks.

4.4.2. Detection and Mitigation Mechanism at MEC

The Detection and Mitigation Mechanism at the Multi-Access Edge Computing (MEC) layer is designed to handle incoming network traffic in real time. This system is created to detect Distributed Denial of Service (DDoS) attacks and block malicious traffic while ensuring legitimate traffic is forwarded seamlessly. The process is carried out step by step as shown in Figure 4.

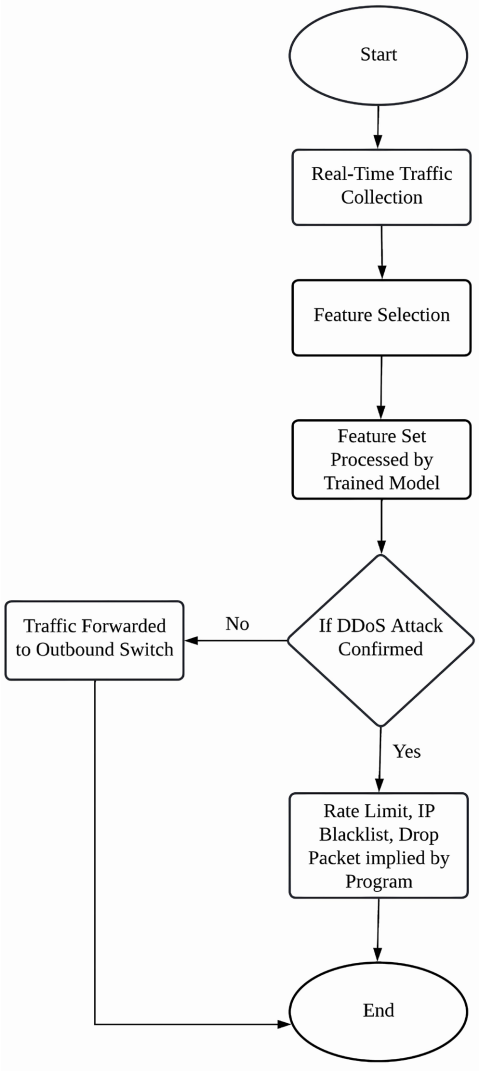

Figure 4. Detection and mitigation process at MEC.

Step 1: Start and Real-Time Traffic Collection

All incoming traffic is collected continuously by the MEC layer from Inbound Switch. A Traffic Collector is used to monitor and capture live network traffic. This ensures that no packet bypasses the system. The traffic is prepared for further analysis.

Step 2: Feature Extraction

After the traffic is collected, important features are extracted from the packets. Attributes like packet size, protocol type, flow duration, and packet rate are analyzed and structured. This step is carried out in real-time to prepare the data for classification. The selected features are critical for distinguishing between legitimate and malicious traffic by providing insights into the flow characteristics of the network. By focusing on these key attributes, the model can efficiently identify patterns indicative of DDoS attacks. Accurate and timely feature extraction ensures the system's responsiveness and effectiveness in detecting malicious traffic in real-world scenarios.

Step 3: Feature Set Processing by the Trained Machine Learning Algorithm

The extracted feature set is processed using the pre-trained Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost Machine Learning algorithm. This model was trained earlier on the CIC-DDoS2019 dataset. The features are analyzed, and the traffic is classified into two categories:

- Legitimate Traffic, which follows normal patterns.
- Malicious Traffic, which shows unusual behavior and is likely to be part of a DDoS attack.

The classification is performed quickly and efficiently using decision trees in the Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost models.

Step 4: Confirmation of DDoS Attack

If the traffic is classified as malicious, the system checks whether the patterns confirm a DDoS attack. If no attack is detected, the traffic is forwarded as legitimate. However, if a DDoS attack is confirmed, the system immediately moves to mitigation.

Step 5: Mitigation of Malicious Traffic

Mitigation measures are applied to block or reduce the impact of malicious traffic. Several actions are taken:

- **Rate Limiting**: The flow of packets from the malicious source is reduced to minimize the damage.
- **IP Blacklisting**: Malicious IP addresses are blocked, preventing them from sending further traffic.
- **Packet Dropping**: Malicious packets are immediately discarded, stopping them from reaching their destination.

These actions are carried out in real-time, ensuring that the network is protected without delays.

Step 6: Forwarding Legitimate Traffic

Traffic classified as legitimate is forwarded through the Outbound Switch to its

intended destination or the core network. This ensures that normal users experience no disruptions and that valid traffic continues uninterrupted.

5. Project Description

5.1. Real-Time Detection of DDoS Attacks Using CIC-DDoS2019 and ML Classifiers

This project focuses on designing an Intrusion Detection System (IDS) for detecting and mitigating DDoS attacks using machine learning techniques, specifically the Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost Classifier. The project utilizes the CIC-DDoS2019 dataset, a real-world dataset that contains diverse traffic patterns, including malicious and legitimate flows. The dataset is preprocessed to remove irrelevant columns, handle missing values, and prepare features for model training. The input features are analyzed, and the target labels (attack or normal) are defined for classification purposes. The data is split into training and testing sets, with 80% used for model learning and 20% for evaluation. A Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost model is trained to classify network traffic accurately, with performance assessed using metrics such as accuracy and a confusion matrix. Hyperparameter optimization is performed using GridSearchCV to improve the model's performance. This project demonstrates how machine learning can be leveraged to detect malicious traffic patterns in real time, contributing to 5G network security. The structured approach ensures that the system is robust, scalable, and capable of effectively differentiating between normal and DDoS attack traffic.

5.1.1. Library Import

The first step involves importing all the necessary Python libraries (Figure 5) required for data loading, preprocessing, feature selection, model building, and evaluation. Libraries such as Pandas and NumPy form the backbone of data handling. Pandas is utilized to work with tabular data, enabling easy reading, cleaning, and manipulation of datasets. NumPy supports numerical operations, such as array handling and mathematical computations, which are essential for machine learning tasks.

```
]: # Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder, label_binarize
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
import xgboost as xgb
import joblib
from imblearn.over_sampling import SMOTE
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_curve, auc
```

Figure 5. Importing python libraries.

For visualization, Matplotlib and Seaborn are imported. These libraries are used to create graphical representations, such as correlation matrices and confusion matrices, making it easier to analyze and interpret data patterns.

From Scikit-learn, a comprehensive machine learning library, tools like train_test_split are used for splitting datasets into training and testing subsets. Preprocessing utilities such as StandardScaler and LabelEncoder standardize and encode data, ensuring it is suitable for model training. Feature selection is achieved using SelectKBest and mutual_info_classif, which identify the most relevant features for classification tasks.

The machine learning models include RandomForestClassifier, KNeighborsClassifier, and XGBoost (from the XGBoost library). These algorithms are the core of the detection mechanism, trained to classify traffic as benign or malicious. The SMOTE library (Synthetic Minority Over-sampling Technique) is employed to handle class imbalances in the training data, ensuring the models perform effectively even with skewed datasets.

Additional utilities like joblib enable saving and loading trained models, while evaluation metrics such as classification_report, confusion_matrix, roc_curve, and auc assess model performance. By importing these libraries at the start, a comprehensive toolkit is established for building a robust DDoS detection system.

5.1.2. Loading Dataset

The dataset, CIC-DDoS2019, is loaded into the Python environment using the pd.read_csv() function from the Pandas library (Figure 6). This function reads the data from a CSV file and converts it into a DataFrame, a table-like structure that allows for easy access, analysis, and manipulation of the data.

After loading, the script displays the column names using the columns attribute of the DataFrame, providing an overview of the dataset's features. The shape of the dataset, representing the number of rows and columns, is displayed using the .shape attribute, giving an idea of the dataset's size.

```
# Load the dataset
dataset = pd.read_csv("cicddos2019_dataset.csv")
```

Figure 6. Dataset path.

To explore the dataset (Figure 7) more effectively, the Pandas option display.max_columns is set to None, ensuring that all columns are visible when viewing the DataFrame. The dataset itself is printed, which allows a complete view of its structure, column names, and sample values.

This preliminary exploration is essential for understanding the dataset and preparing for further analysis. Key details such as column names, data types, and the presence of missing values will be examined in subsequent steps to address any issues and prepare the data for modeling.

```
print("\nDataset Columns:")
print(dataset.columns)

# Display the shape of the dataset
print("\nShape of the dataset:")
print(dataset.shape)

# set the max columns to none
pd.set_option('display.max_columns', None)
dataset
```

```
       'Bwd Packets Length Total', 'Fwd Packet Length Max',
       'Fwd Packet Length Min', 'Fwd Packet Length Mean',
       'Fwd Packet Length Std', 'Bwd Packet Length Max',
       'Bwd Packet Length Min', 'Bwd Packet Length Mean',
       'Bwd Packet Length Std', 'Flow Bytes/s', 'Flow Packets/s',
       'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', 'Flow IAT Min',
       'Fwd IAT Total', 'Fwd IAT Mean', 'Fwd IAT Std', 'Fwd IAT Max',
       'Fwd IAT Min', 'Bwd IAT Total', 'Bwd IAT Mean', 'Bwd IAT Std',
       'Bwd IAT Max', 'Bwd IAT Min', 'Fwd PSH Flags', 'Bwd PSH Flags',
       'Fwd URG Flags', 'Bwd URG Flags', 'Fwd Header Length',
       'Bwd Header Length', 'Fwd Packets/s', 'Bwd Packets/s',
       'Packet Length Min', 'Packet Length Max', 'Packet Length Mean',
       'Packet Length Std', 'Packet Length Variance', 'FIN Flag Count',
       'SYN Flag Count', 'RST Flag Count', 'PSH Flag Count', 'ACK Flag Count',
       'URG Flag Count', 'CWE Flag Count', 'ECE Flag Count', 'Down/Up Ratio',
       'Avg Packet Size', 'Avg Fwd Segment Size', 'Avg Bwd Segment Size',
       'Fwd Avg Bytes/Bulk', 'Fwd Avg Packets/Bulk', 'Fwd Avg Bulk Rate',
       'Bwd Avg Bytes/Bulk', 'Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate',
       'Subflow Fwd Packets', 'Subflow Fwd Bytes', 'Subflow Bwd Packets',
       'Subflow Bwd Bytes', 'Init Fwd Win Bytes', 'Init Bwd Win Bytes',
       'Fwd Act Data Packets', 'Fwd Seg Size Min', 'Active Mean', 'Active Std',
       'Active Max', 'Active Min', 'Idle Mean', 'Idle Std', 'Idle Max',
       'Idle Min', 'Label', 'Class'],
      dtype='object')

Shape of the dataset:
(431371, 80)
```

3]:

	Unnamed: 0	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max	Bwd Packet Length Min	Bwd Packet Length Mean	Bwd Packet Length Std	Flow
0	0	17	216631	6	0	2088.0	0.0	393.0	321.0	348.00000	35.088460	0.0	0.0	0.000000	0.000000	9.638
1	1	17	2	2	0	802.0	0.0	401.0	401.0	401.00000	0.000000	0.0	0.0	0.000000	0.000000	4.010
2	2	17	48	2	0	766.0	0.0	383.0	383.0	383.00000	0.000000	0.0	0.0	0.000000	0.000000	1.595
3	3	17	107319	4	0	1398.0	0.0	369.0	330.0	349.50000	22.516660	0.0	0.0	0.000000	0.000000	1.302
4	4	17	107271	4	0	1438.0	0.0	389.0	330.0	359.50000	34.063667	0.0	0.0	0.000000	0.000000	1.340
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
431366	306196	6	550740	2	1	0.0	31.0	0.0	0.0	0.00000	0.000000	31.0	31.0	31.000000	0.000000	5.628
431367	306197	6	337766	1	3	31.0	62.0	31.0	31.0	31.00000	0.000000	31.0	0.0	20.666666	17.897858	2.753
431368	306198	6	228	1	2	0.0	0.0	0.0	0.0	0.00000	0.000000	0.0	0.0	0.000000	0.000000	0.000
431369	306199	17	47709	2	2	86.0	194.0	43.0	43.0	43.00000	0.000000	97.0	97.0	97.000000	0.000000	5.868
431370	306200	6	322010	30	33	5902.0	7444.0	1073.0	0.0	196.73334	353.039300	2525.0	0.0	225.575760	515.547240	4.144

431371 rows × 80 columns

Figure 7. Basic information about the dataset.

5.1.3. Handling Missing Values

Missing or infinite values can cause problems during model training. To address this, infinite values were first replaced with NaN using the replace() function. Then, rows containing NaN values were removed using the dropna() function. This ensured that the dataset was clean and ready for analysis. Handling these values also helped prevent errors during the training process. The code below was used:

```
# Replace infinite values with NaN and drop rows with NaN values
dataset.replace([np.inf, -np.inf], np.nan, inplace=True)
dataset.dropna(inplace=True)
```

5.1.4. Selection of Relevant Classes

The selected classes encompass a range of benign and malicious traffic, including denial-of-service attacks (DrDoS) targeting various protocols (DNS, LDAP, MSSQL, NTP, NetBIOS, SNMP, UDP), as well as other attack types such as LDAP, MSSQL, NetBIOS, Portmap, Syn, TFTP, UDP, UDP-lag, and WebDDoS. This selection enables the model to learn distinct patterns associated with these prevalent attacks, contributing to a more accurate and robust intrusion detection system. By concentrating on these specific classes, the research aims to provide valuable insights into their characteristics and improve mitigation strategies against them.

selected_classes = ['Benign', 'DrDoS_DNS', 'DrDoS_LDAP', 'DrDoS_MSSQL', 'DrDoS_NTP', 'DrDoS_NetBIOS', 'DrDoS_SNMP', 'DrDoS_UDP', 'LDAP', 'MSSQL', 'NetBIOS', 'Portmap', 'Syn', 'TFTP', 'UDP', 'UDP-lag', 'UDPLag', 'WebDDoS']

dataset = dataset [dataset ['Label'].isin(selected_classes)]

5.1.5. Feature Selection

The dataset included many features, but not all of them were equally useful for detecting DDoS attacks. A set of ten important features was selected based on domain knowledge and their relevance to network traffic. These features included Flow Duration, Total Fwd Packets, and Flow IAT Mean. By selecting only these features, the complexity of the dataset was reduced, and the model's performance improved. The selected features are shown below:

selected_features = ['Flow Duration', 'Total Fwd Packets', 'Total Backward Packets', 'Fwd Packet Length Max', 'Bwd Packet Length Max', 'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', 'Fwd Packets/s', 'Bwd Packets/s'] X = dataset [selected_features] y = dataset ['Label']

5.1.6. Correlation Analysis

Understanding the relationships between features is essential for improving the model's accuracy. A correlation matrix was created to identify how features were related to one another. The matrix was visualized using a heatmap, which made it easier to see which features had strong correlations. Features with high correlation values often carry similar information, so this step also guided further feature selection. The correlation heatmap was created with the following code (**Figure 8**).

5.1.7. Splitting the Dataset

To train and test the machine learning models, the dataset was divided into two parts: training data (80%) and testing data (20%). This split ensured that the models were trained on one set of data and evaluated on another. Stratified sampling was used to maintain a balanced distribution of classes in both subsets. The following code was executed for this step:

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

5.1.8. Handling Imbalanced Data

The dataset was imbalanced, meaning that some classes had significantly fewer

samples than others. To address this, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training data. SMOTE created synthetic samples for the minority classes, making the dataset more balanced. A balanced dataset improves the model's ability to detect all classes effectively.

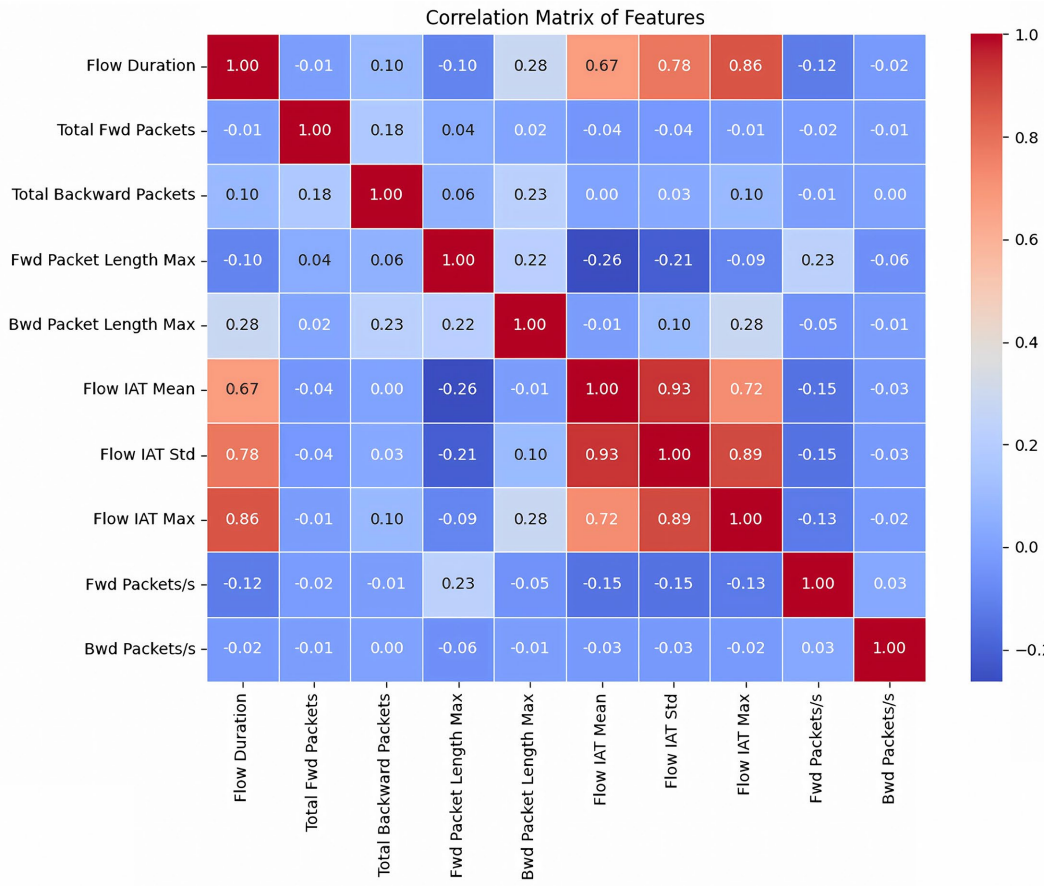

Figure 8. Correlation matrix of features.

5.1.9. Normalizing Features

Machine learning models perform better when features are on a similar scale. To achieve this, the StandardScaler was used to normalize the data. After normalization, all features had a mean of 0 and a standard deviation of 1. This step ensured that no feature dominated others due to differences in scale.

5.1.10. Confusion Matrix

The confusion matrices provide a visual assessment of the classification performance for three different machine learning models: Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost (XGB). Each matrix reveals the models' strengths and weaknesses in accurately identifying various network traffic categories, including both benign traffic and different types of attacks.

- **Random Forest:** The Random Forest model (**Figure 9**) demonstrates strong overall performance with high accuracy in classifying several attack types, notably DrDoS_NTP, Syn, and TFTP. This success can be attributed to its ability

to learn complex patterns in the data and handle high dimensionality. However, some confusion arises between classes like DrDoS_LDAP and DrDoS_DNS, suggesting potential similarities in their network characteristics. Areas for improvement include enhancing the model's ability to distinguish between these closely related attacks and increasing accuracy for classes like WebDDoS, potentially through the addition of more training data or refined feature engineering.

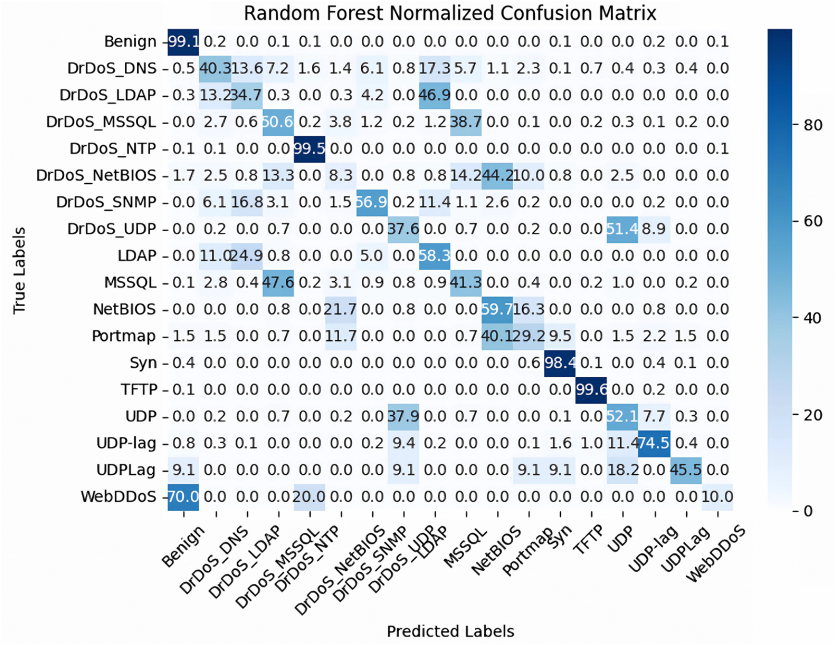

Figure 9. Random forest confusion matrix.

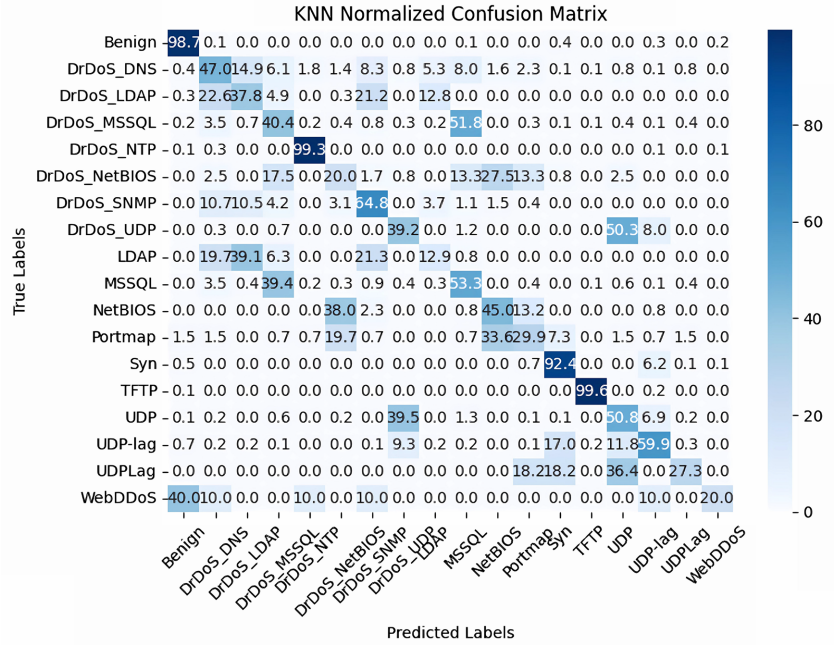

Figure 10. KNN confusion matrix.

- **K-Nearest Neighbors**: The KNN model (Figure 10) shares similar strengths with Random Forest, exhibiting high accuracy in classifying DrDoS_NTP and Syn attacks. This efficacy likely stems from KNN's ability to capture local relationships in the data and its relative simplicity in implementation. However, KNN also faces challenges in differentiating between DrDoS_LDAP and DrDoS_DNS, indicating a need for improved feature selection or distance metrics to better separate these classes. Notably, KNN shows improved performance compared to Random Forest in classifying WebDDoS traffic, suggesting its potential suitability for this specific attack category.
- **XGBoost:** XGBoost (Figure 11) demonstrates high accuracy in classifying DrDoS_MSSQL, DrDoS_NTP, and Syn attacks, likely due to its ensemble learning approach and ability to handle complex relationships within the data. While XGBoost also exhibits some confusion between DrDoS_LDAP and DrDoS_DNS, it shows comparable performance to KNN in classifying WebDDoS traffic, highlighting its potential for detecting this attack type. Further improvements could focus on addressing the confusion between specific attack classes and optimizing hyperparameters to enhance overall performance.

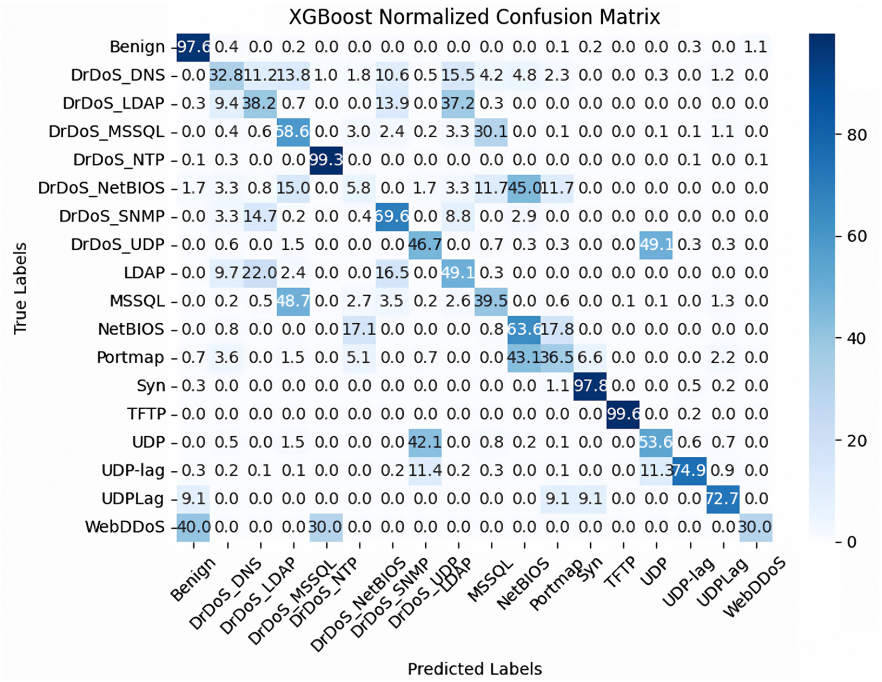

Figure 11. XGBoost confusion matrix.

5.1.11. Implementation of the Random Forest (RF), K-Nearest Neighbor (KNN) and XGBoost Model with Zeek at MEC

In this phase, the implementation of the Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost models was tested for their accuracy before deploying them in a Multi-Access Edge Computing (MEC) environment. MEC allows data to be processed closer to the network's edge, significantly reducing latency and

improving response times for real-time applications. This setup is particularly beneficial for applications that require fast detection and mitigation of network threats, such as Distributed Denial-of-Service (DDoS) attacks.

Zeek, an open-source Intrusion Detection System (IDS), plays a key role in monitoring network traffic in real time. Zeek captures traffic flow data and extracts relevant features, such as Flow Duration, Total Fwd Packets, Fwd Packet Length Max, Flow IAT Mean, and Bwd Packets/s. These features are then sent to the pre-trained machine learning models (RF, KNN, and XGBoost) for classification. Based on the input features, the models predict whether the traffic is benign or malicious.

In this phase, the models were tested by simulating DDoS attacks using Python code. When the models classify traffic, legitimate traffic is forwarded to the core network, while malicious traffic is blocked and contained at the MEC layer. This process is dynamic, occurring in real-time without the need for intermediate storage like CSV files, which allows immediate action against threats.

The trained models and Python code were uploaded to a Virtual Private Server (VPS) for testing. The models were saved as .pkl files and loaded into the testing server, where they were used to simulate and analyze traffic. This setup ensures that the models can be tested under live conditions before full deployment.

For future work, the goal is to integrate the trained machine learning models directly with Zeek in the MEC environment. This integration will enable seamless, real-time detection and mitigation of DDoS attacks. By leveraging the low-latency characteristics of MEC, this system aims to block malicious traffic as soon as it is detected, improving overall network security without relying on cloud resources or manual intervention.

5.2. Testing the Models Using Python Script

On an Ubuntu 22.04 LTS machine (**Figure 12**), the Python script mec_traffic_monitor.py was created and executed to simulate real-time traffic monitoring and classification. The pre-trained .pkl files for the Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost models, the models, along with the StandardScaler, were previously downloaded from Google Colab, were uploaded and placed in the same directory as the script. Using the python3 command, the script initiated a dynamic process to simulate traffic flows, with a portion of the traffic mimicking malicious DDoS attacks. Legitimate traffic was identified and forwarded, while malicious traffic was blocked, and the associated IP addresses were recorded in a file named blacklist.txt. A one-second delay (time.sleep(1)) was introduced to regulate the speed of classification, making the results easy to observe in the terminal. It is important to note that this is only a demonstration of the proposed idea. To fully implement this system in a real-world scenario, a complete software solution would need to be developed to run on the MEC layer, capable of handling live traffic streams with scalability and robustness.

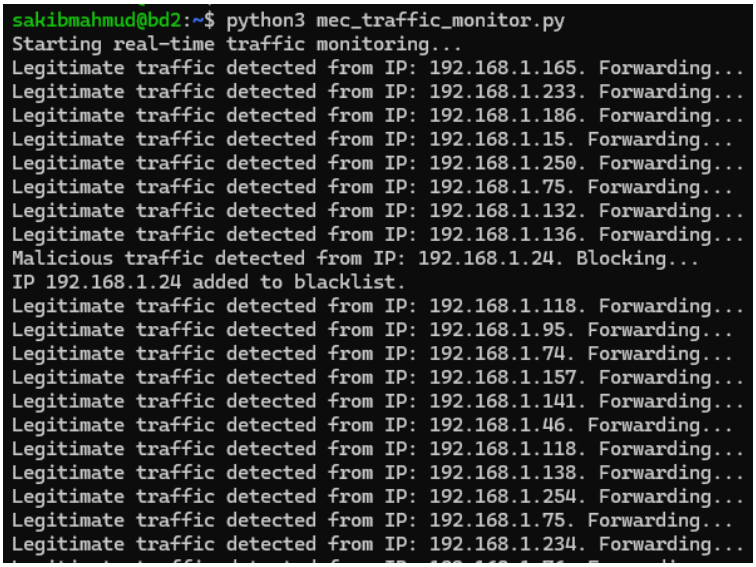

Figure 12. Result of demonstration.

5.3. Evaluation and Results

In Table 2, we present the evaluation results for the Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost models, which were trained on the CIC-DDoS2019 dataset. The models were evaluated using key classification metrics, including precision, recall, F1-score, and accuracy. These metrics allow for a comprehensive understanding of the models' effectiveness in classifying network traffic into benign and malicious categories.

Table 2. Classification report.

Class	Precision (RF)	Recall (RF)	F1-Score (RF)	Precision (KNN)	Recall (KNN)	F1-Score (KNN)	Precision (XGBoost)	Recall (XGBoost)	F1-Score (XGBoost)
Benign	0.99	0.99	0.99	0.99	0.99	0.99	1.00	0.98	0.99
DrDoS_DNS	0.51	0.40	0.45	0.45	0.47	0.46	0.46	0.33	0.38
DrDoS_LDAP	0.25	0.35	0.29	0.24	0.38	0.29	0.29	0.38	0.33
DrDoS_MSSQL	0.39	0.51	0.44	0.37	0.40	0.39	0.40	0.59	0.48
DrDoS_NTP	1.00	1.00	1.00	1.00	0.99	1.00	1.00	0.99	1.00
DrDoS_NetBIOS	0.05	0.08	0.06	0.16	0.20	0.18	0.05	0.06	0.05
DrDoS_SNMP	0.72	0.57	0.64	0.59	0.65	0.62	0.57	0.70	0.63
DrDoS_UDP	0.33	0.38	0.35	0.34	0.39	0.36	0.36	0.47	0.41
LDAP	0.38	0.58	0.46	0.30	0.13	0.18	0.34	0.49	0.40
MSSQL	0.54	0.41	0.47	0.52	0.53	0.53	0.58	0.40	0.47
NetBIOS	0.36	0.60	0.45	0.35	0.45	0.40	0.31	0.64	0.41
Portmap	0.23	0.29	0.25	0.22	0.30	0.25	0.19	0.36	0.25
Syn	0.99	0.98	0.99	0.96	0.92	0.94	0.99	0.98	0.99
TFTP	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Continued

UDP	0.59	0.52	0.55	0.59	0.51	0.54	0.61	0.54	0.57
UDP-lag	0.69	0.74	0.72	0.48	0.60	0.53	0.87	0.75	0.81
UDPLag	0.12	0.45	0.20	0.06	0.27	0.10	0.06	0.73	0.12
WebDDoS	0.03	0.10	0.05	0.03	0.20	0.05	0.01	0.30	0.02
Accuracy	**0.92**	**0.92**	**0.92**	**0.91**	**0.91**	**0.91**	**0.92**	**0.92**	**0.92**
Macro avg	0.51	0.55	0.52	0.48	0.52	0.49	0.50	0.59	0.52
Weighted avg	0.92	0.92	0.92	0.91	0.91	0.91	0.93	0.92	0.92

5.3.1. Model Performance Overview

The evaluation of the three models revealed the following results:

- **Accuracy**: All three models showed high accuracy in detecting network traffic. The Random Forest (RF) and XGBoost models achieved an accuracy of 92%, while the K-Nearest Neighbor (KNN) model was slightly lower at 91%. These accuracy scores indicate that all three models are fairly effective in distinguishing benign traffic from various types of DDoS attacks.
- **Precision, Recall, and F1-Score**: The RF and XGBoost models performed exceptionally well in detecting benign traffic, with precision and recall values close to 1.00. The KNN model showed comparable performance for benign traffic but struggled more with certain attack types.

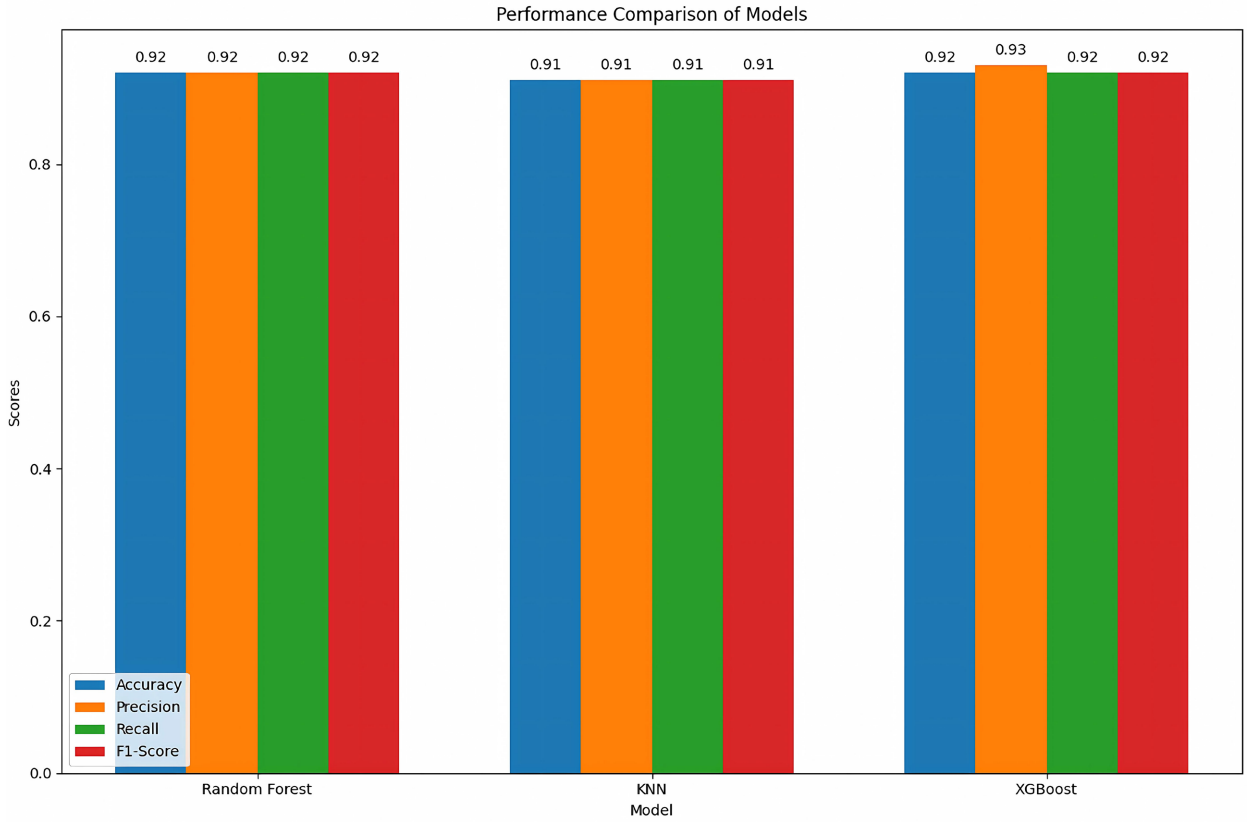

Figure 13. Performance comparison of models.

- The DrDoS_NTP class had high precision and recall across all models, demonstrating the models' strength in detecting this attack type.
- Attack classes such as DrDoS_LDAP and DrDoS_NetBIOS were more challenging for all models, with lower precision and recall values, indicating these attacks were harder to detect consistently.
- WebDDoS presented a particular challenge, with KNN and XGBoost having very low precision and recall for this class. Random Forest also struggled with this class, although it performed slightly better.

5.3.2. Performance Comparison

The following performance comparison bar chart (Figure 13) illustrates the accuracy of the models. From the chart, we observe that Random Forest (RF) and XGBoost consistently outperformed KNN in terms of overall accuracy. Despite KNN performing well on certain traffic types, the more complex nature of some DDoS attacks required the more robust models, RF and XGBoost, to yield better results.

The models performed well in distinguishing benign traffic from attacks such as DrDoS_NTP, Syn, and TFTP. However, the challenges were evident for attacks with low frequencies, such as DrDoS_LDAP and WebDDoS, where recall values were notably lower.

6. Conclusions

This study focused on the detection and mitigation of Distributed Denial of Service (DDoS) attacks in 5G networks by leveraging machine learning algorithms. Specifically, Random Forest (RF), K-Nearest Neighbor (KNN), and XGBoost were used to classify traffic as benign or malicious. The approach aimed to address the vulnerabilities of 5G networks, particularly those arising from compromised end-user devices. While the project was conceptual, a Python script was used to demonstrate how these models can detect attacks, mimicking a real-world scenario. The CIC-DDoS2019 dataset served as a solid foundation for training and testing the models, allowing the system to identify and mitigate malicious traffic in real time. The results indicated that the proposed solution could improve network security and minimize service disruptions by detecting threats at the edge. Although the core components of the framework were validated, the research did not implement a fully operational system at the MEC layer, marking a clear area for future development.

6.1. Limitations

Despite the positive results, the study faced several limitations. The most significant challenge is the cost of deploying Multi-Access Edge Computing (MEC) infrastructure at the edge of the network, which could impede its scalability in real-world applications. Moreover, the project did not result in a fully deployable solution that could run autonomously at the MEC layer. Although the machine learning models were validated, the implementation was limited to simulations

and requires substantial development to transition into a practical deployment. Additionally, the research focused only on network-layer DDoS attacks, overlooking application-layer threats, which also pose considerable risks in 5G environments.

6.2. Future Works

We successfully integrated the trained RF, KNN, and XGBoost models with Zeek IDS for real-time traffic monitoring and DDoS detection. Future work should focus on fully implementing this system at the MEC layer, ensuring it can autonomously process live traffic and mitigate attacks in real time. Additionally, research should aim to reduce the cost of MEC deployment by exploring shared infrastructure or optimizing resource allocation. Expanding the system to address application-layer DDoS attacks and incorporating other machine learning algorithms could enhance its accuracy and adaptability. This will make the system more robust against evolving threats, enabling it to effectively secure 5G networks at scale.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] GSMA (2019) 5G and the Future: How 5G Research Is Transforming Our World. https://www.gsma.com/get-involved/gsma-membership/wp-content/uploads/2019/11/5G-Research_A4.pdf

[2] Rinf.tech (2024) The IoT Security in the 5G Era. https://www.rinf.tech/the-iot-security-in-the-5g-era/

[3] Robustel (2024) How 5G Autonomous Vehicles Will Benefit from 5G. https://www.robustel.com/iot-technology-solutions-blog/how-5g-autonomous-vehicles-will-benefit-from-5g-1/

[4] Onoja, D., Hitchens, M. and Shankaran, R. (2022) DDoS Threats and Solutions for 5G-Enabled IoT Networks. In: Pal, S., Jadidi, Z. and Foo, E., Eds., *Secure and Trusted Cyber Physical Systems*, Springer, 115-133. https://doi.org/10.1007/978-3-031-08270-2_5

[5] Ghorbani, H., Mohammadzadeh, M.S. and Ahmadzadegan, M.H. (2020) DDoS Attacks on the IoT Network with the Emergence of 5G. 2020 *International Conference on Technology and Entrepreneurship— Virtual* (*ICTE- V*), San Jose, 20-21 April 2020, 1-5. https://doi.org/10.1109/icte-v50708.2020.9113779

[6] Serrano Mamolar, A., Salvá-García, P., Chirivella-Perez, E., Pervez, Z., Alcaraz Calero, J.M. and Wang, Q. (2019) Autonomic Protection of Multi-Tenant 5G Mobile Networks against UDP Flooding DDoS Attacks. *Journal of Network and Computer Applications*, **145**, Article ID: 102416. https://doi.org/10.1016/j.jnca.2019.102416

[7] Sanmorino, A. and Yazid, S. (2013) DDoS Attack Detection Method and Mitigation Using Pattern of the Flow. 2013 *International Conference of Information and Communication Technology* (*ICoICT*), Bandung, 20-22 March 2013, 12-16. https://doi.org/10.1109/icoict.2013.6574541