Learning at the Edge: Simulated DDoS Detection in 5G Networks

Karim Khalil*, Lars Breum Hansen†, Elham Tayebi‡, Christian Gehrmann
Department of EIT, Lund University, Lund, Sweden
{karim.khalil, christian.gehrmann}@eit.lth.se,
{†la8177br-s, ‡el1461ta-s}@student.lu.se

***Abstract*—The growing use of 5G networks for critical services makes them vulnerable to Distributed Denial of Service (DDoS) attacks. While numerous Machine Learning (ML)-based approaches have been proposed, the real-world deployability of these models remains understudied. This work presents what is, based on existing literature, the first simulation-driven methodology to evaluate both the transferability and the operational feasibility of ML-driven DDoS detection in realistic 5G Multi-Access Edge Computing (MEC) settings. The study assess the cross-scenario performance of two state-of-the-art Convolutional Neural Network (CNN) DDoS detection models using three diverse datasets, including synthetic traffic representative of 5G environments. Leveraging the full 5G network simulator Simu5G, the study integrate the better-performing model into an MEC application to demonstrate a functional end-to-end pipeline from offline training to live attack mitigation. This approach delivers a reproducible framework for testing ML-based network defenses under realistic yet controllable conditions, enabling systematic evaluation beyond static benchmarks. The results confirm the feasibility of assessing the practical resilience of ML-driven DDoS defenses in 5G networks, with several areas identified for further optimization, including expansion of attack scenarios, enhancement of model robustness across datasets, and refinement of deployment strategies within the simulation environment.**

***Index Terms*—5G network simulation, DDoS, dataset generation, ML-based anomaly detection, Multi-Access Edge Computing (MEC).**

I. Introduction

Distributed Denial of Service (DDoS) attacks remain a growing threat to digital infrastructure. In Q4 2024, Cloudflare reported mitigating the largest recorded botnet-based DDoS attack, involving 13 million devices and peaking at 5.6 Tbps [1]. These attacks are increasingly driven by compromised Internet of Things (IoT) devices, whose global count is projected to surpass 18.8 billion in 2024 [2]. 9% of cybersecurity incidents in European hospitals in 2023 were DDoS attacks [3].

With 5G enabling smart cities, remote healthcare, and autonomous systems, ensuring its availability is critical [4], [5]. 5G's high-speed bandwidth and low-latency connections also facilitate botnet-driven DDoS attacks. Such volumetric attacks (e.g., UDP-flooding) aim to exhaust network capacity [6], affecting not only targeted services but also collateral users on the same 5G infrastructure [7].

This work was supported in part by the WASP Foundation and the research project funded by the ELLIIT Foundation.
*Corresponding author

Significant research efforts have targeted efficient DDoS detection models [8]–[12], and the impact of DDoS on mobile network performance is also well studied [7], [8], [13], [14]. In contrast to works aimed at producing higher-accuracy models, this study's focus is on proposing and demonstrating a methodology to evaluate an often overlooked aspect of Machine Learning (ML)-based DDoS detection: how well such models perform when deployed in realistic 5G environments and within the resource constraints of an operational setting. For this initial investigation, this study apply the approach to a single representative DDoS scenario, high-rate volumetric UDP flooding, chosen for its impact on both 5G infrastructure and downstream services, and for its suitability to controlled simulation. The study focuses on Convolutional Neural Network (CNN) models due to their proven effectiveness in capturing spatial and statistical features from network traffic while maintaining computational efficiency [15]. Other DDoS types, such as low-rate or application-layer attacks, are recognized as important but are outside the scope of this study.

This work explores a practical evaluation pipeline for ML-based DDoS anomaly detection in 5G networks, focusing on how well existing high-performing CNN-based detection models perform when exposed to a synthetic serialized DDoS dataset with similar construction and features. Using the Simu5G framework, synthetic 5G data traffic is generated, a selected detection model is integrated into a Multi-Access Edge Computing (MEC) application, and both detection performance and network-level effects are analyzed during simulated attacks. Rather than claiming a definitive detection solution, this study aim is to illustrate a methodology that bridges offline model evaluation with operational considerations, including processing overhead, responsiveness, and feasibility of real-time deployment. Additionally, the study discusses current limitations in dataset diversity, DDoS attack profiles, and mitigation design, and outline how these constraints will be addressed in future work.

The main contributions of the paper are the following:

1) Develop and demonstrate a simulation-based methodology to assess how existing DDoS detection models, trained initially on benchmark datasets, perform when evaluated on a synthetically generated 5G traffic dataset.
2) Integrate the higher-performing of the evaluated models into a full MEC-enabled 5G simulation, allowing joint

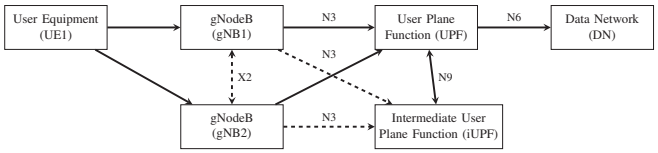


Fig. 1. 5G user-plane topology with multiple gNodeBs and key interfaces

assessment of detection accuracy, resource use, and impact on network traffic dynamics during live attack scenarios.

3) Highlight key findings on synthetic datasets generation, detection feasibility, and operational overhead, as well as the current limitations in the defined attack coverage and mitigation mechanisms, which collectively define a roadmap for future works.

The remainder of the paper is structured as follows: Section II provides background, followed by the DDoS attack assumptions, and related work in Sections III and IV. Section V describes the system model, with results and discussion in Sections VI and VII. Conclusions and future work are given in Section VIII.

II. Background

A. 5G User-Plane Network Topology

The 5G user plane is responsible for forwarding data packets between UEs and external data networks. As specified in 3GPP TS 23.501, this involves key components such as the gNodeB (gNB), which provides radio access and handles both control and user data, and the User Plane Function (UPF), which enforces traffic policies and manages QoS. In larger deployments, an Intermediate UPF (iUPF) may be used to aggregate traffic from multiple gNBs before forwarding to the core network. Figure 1 illustrates the main user-plane components and relevant interfaces.

B. Available Datasets

A major challenge in DDoS detection research is the availability of relevant and high-quality datasets. While many datasets exist, few focus solely on volumetric DDoS attacks, which are most relevant for this work. Three primary datasets were evaluated and the drawbacks and advantages of each dataset are summarized in Table I. The CICDoS2019 dataset [13] was used as the performance benchmark in this work.

III. DDoS Assumptions

For this study, a threat scenario is considered in which no dedicated DDoS defenses or detection mechanisms are present within the 5G network infrastructure. The adversary is assumed to operate a botnet of compromised cellular IoT devices, using the 5G network solely as a medium to transport DDoS packets towards targeted backend Internet services. The attacker does not have access to, or control over, any part of the 5G infrastructure itself, such as gNodeBs, User Plane Functions, or core network components, nor does the scenario include any physical-layer attacks or internal compromise. The adversary's goal is to maximize attack impact on the external service while minimizing resource expenditure and avoiding detection.

This initial focus allows a clear evaluation of ML-based DDoS detection deployed at the network edge, with the assumption that the 5G operator is a neutral party until defensive measures are introduced in the simulation. Other attack types, such as low-rate or application-layer DDoS, as well as scenarios involving infrastructure compromise, remain outside the scope of this study to ensure controlled assessment of the selected volumetric UDP flooding scenario.

Table II summarizes the key characteristics of the DDoS scenario.

IV. Related Work

Recent work [17] has reviewed the security challenges and defense techniques for DDoS attacks in MEC networks. MEC is increasingly recognized as an advantageous environment for deploying DDoS detection mechanisms due to its proximity to end devices and ability to perform early traffic analysis.

In terms of datasets used for DDoS detection research in MEC contexts, most studies have historically relied upon established datasets like KDDCUP99, UNSW-NB15, CISIDS2017, and CICDDoS2019. While these are valuable, they lack the diversity and edge-specific attack patterns, limiting their efficacy for realistic MEC deployments. Zeeshan et al. [18] demonstrated the benefits of synthesizing more comprehensive datasets by merging features from UNSW-NB15 and Bot-IoT, thereby improving model generalization. However, the focus remained on experimental improvements rather than practical edge deployment. The necessity for improved datasets aligns with the trend of leveraging MEC servers as platforms for real-time DDoS detection.

Recent studies have demonstrated the effectiveness of ML models in processing and analyzing large-scale network datasets for DDoS detection. These studies have employed various approaches for preprocessing and data preparation, utilizing different model architectures and different datasets. In [9], the CSE-CIC-IDS2018 AWS was used to train neural network-based models, while [8] trained Artificial Neural Networks (ANNs) and LSTMs to capture complex DDoS traffic patterns. In [10], a deep neural network DNN-based method was proposed to detect DDoS attacks in Software-Defined Networking (SDN) environments, demonstrating high detection accuracy across several datasets.

The LUCID framework, introduced in [12], proposes a CNN-based solution tailored for resource-constrained environments. It achieves a 40x speedup over traditional DL models while maintaining high detection accuracy. LUCID uses a dataset-agnostic preprocessing method that converts raw network traffic into spatial representations, thereby reducing the need for manual feature engineering. It extracts packet-level attributes to enable flexible, real-time classification and has shown high performance across varied network conditions.

TABLE I. COMPARISON OF DDOS DATASETS

Dataset	Generation Method	Advantages	Drawbacks
KDDCup1999 [14]	Live capture with simulated attacks	Historically important, large volume	Outdated; contains statistical and structural flaws
UNSW-NB15 [16]	Live capture with testbed-based attack injection	Better balance of benign and attack traffic	Benign traffic may lack real-world representativeness
CICDoS2019 [13]	High-quality traffic captured with realistic attack scenarios	Modern, well-documented	Labeling strategy can be inconsistent

TABLE II. DDOS SCENARIO

Aspect	Description
Capabilities	Control over a set of cellular IoT devices.
Goals	Launch volumetric UDP-based DDoS on public-facing services.
Constraints	No compromise of infrastructure (e.g., gNBs, UPFs); physical-layer attacks excluded.

The study, [11], introduces a CNN-based model to detect and categorize DDoS attacks. Through the analysis of processed data from the CICDDoS-2019 dataset, the CNN effectively learns to distinguish between malicious traffic and normal activity by identifying important features. Although XGBoost's accuracy was higher, CNN still has considerable potential for automated and flexible attack detection.

While [11] evaluated a CNN-based model trained on 34 selected network features, achieving an accuracy of 83.89% and an F1 score of 0.723. However, it struggled to detect certain attack types (e.g., UDPLag, Portmap), highlighting the need for careful feature selection and dataset balancing to improve generalization.

In total, five studies, utilizing eight datasets and implementing twelve DL or ML models, were evaluated for comparison with this work. A summary of prior DDoS detection research is provided in Table III, which highlights key datasets, techniques, and performance outcomes.

V. METHOD

A simulation-based approach was chosen to safely evaluate the impact of DDoS attacks on 5G infrastructure and validate ML-based mitigation in a controlled environment. Public datasets serve as baselines, while new datasets generated with BoNeSi and Simu5G provide varied traffic for robustness testing.

The investigation was structured into three main stages:

1) **Dataset Generation**: Creating synthetic network traffic using both BoNeSi [19] and Simu5G [20].
2) **Model Training and Evaluation**: Adapting and benchmarking ML models ([12], [11]) using both public and custom datasets.
3) **Deployment in MEC**: Testing the models in an MEC application to assess real-time detection.

Figure 2 shows an architectural overview of this setup. The goal is to test whether ML models can detect and mitigate volumetric DDoS attacks in a simulated 5G network, using only packet-level features and limited compute resources.

A. Dataset Generation Method

Three datasets were utilized to train and evaluate the detection models in this study. The first dataset, **CICDDoS2019-UDP**, is a filtered subset of the publicly available CICDDoS2019 dataset, containing only UDP flood traffic. It serves as a reference dataset due to its frequent use in DDoS detection literature. Traffic was filtered by attack intervals, timestamps, and IP addresses using the LUCID parser.

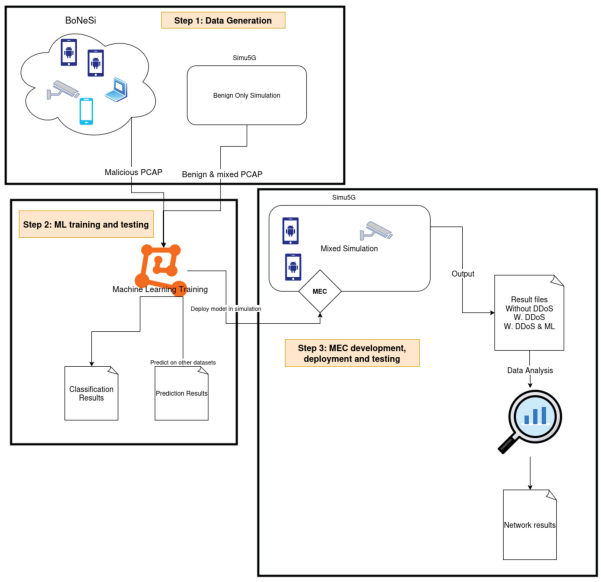


Fig. 2. The architectural overview of the project

To generate the second dataset, named **BoNeSi+Simu5G**, malicious traffic was created using BoNeSi running within a virtual Kali Linux environment. Each attack simulation involved five bots transmitting UDP packets at data rates between 25 and 45 Mbps, with varying packet sizes and sampling rates (see Table IV). Captured packets were parsed into feature data using the LUCID PyShark-based parser. Benign traffic for this dataset was separately generated using Simu5G.

The third dataset, **Mixed Simu5G**, was entirely generated within the Simu5G simulator in OMNeT++ [21]. Benign user equipment (UEs) employed low-rate UDP applications (`UdpBasicApp`, `VideoStreamClient`), whereas malicious UEs generated higher-rate UDP traffic by increasing packet sizes and reducing transmission intervals. Traffic was captured at the network's central router. Simulations ran for 6000 seconds, producing several gigabytes of raw data.

All data processing and simulations were performed on a standard Linux laptop (Lenovo IdeaPad Yoga, 16 GB RAM, Intel i5). Dataset generation scripts and network configurations are publicly accessible on GitHub for reproducibility[1].

[1]https://github.com/karkha-0/LearingAtTheEdge-DDoSimu5G

TABLE III. SUMMARY OF RELATED WORK ON DDOS DETECTION

Study	Dataset Used	Techniques Evaluated	Best Performing Model	Accuracy
[9]	CSE-CIC-IDS2018 (AWS)	Neural Networks (Keras)	Neural Network	99.98%
[10]	InSDN	Deep Neural Networks (DNNs) in SDN	DNN	99.98%
[10]	CICIDS2018	Deep Neural Networks (DNNs) in SDN	DNN	100%
[10]	Kaggle DDoS	Deep Neural Networks (DNNs) in SDN	DNN	99.99%
[8]	CICDDoS-2019	ANNs, LSTMs, SVM, Logistic Regression, Random Forests, Decision Trees	ANN	85.1%
[8]	CICDDoS-2019	ANNs, LSTMs, SVM, Logistic Regression, Random Forests, Decision Trees	LSTM	98%
[11]	CICDDoS-2019	KNN, CNN, XGBoost, SGD, Naive Bayes	XGBoost	89.29%
[12]	ISCX2012, CIC2017, CSECIC2018	CNN-Based Lightweight Detection	CNN (Lucid)	98.88%

TABLE IV. BONESI MALICIOUS DATA GENERATION SETTING

BoNeSi Configuration	Attack 1	Attack 2	Attack 3
# Bots	5	5	5
Goal Total Data rate (Mbps / bot)	25	35	45
Goal Total Data rate (B/s) / bot	3125000	4375000	5625000
Packet Size [B]	512	1024	1400
Sampling Rate	6104	4272	4018
Total Data Rate [Mbps]	125	175	225

B. Experimental Setup

1) DDoS Detection Model: Two open-source DDoS detection frameworks, the LUCID framework [12] and a 1D-CNN model from [11] (referred to as the Mohak model), were selected due to their proven performance and publicly available implementations. Their implementations were obtained from official GitHub repositories[2] [3] and adapted for binary classification (benign vs. malicious).

LUCID's parser (`lucid_dataset_parser.py`) was used to preprocess both CICDDoS2019 and the newly generated datasets. Only UDP-flood packets from known attacker IPs during the labeled intervals (10:53–11:03, 12:45–13:09) [13] were retained. PCAP files were batch-processed into balanced and normalized HDF5 datasets.

The LUCID CNN model applies Conv2D and GlobalMaxPooling2D layers with ReLU activation, followed by a sigmoid-activated dense layer. Minor modifications were made to configure the `DOS2019_FLOWS` dictionary for experiment-specific IPs.

The Mohak model was refactored from its original Jupyter notebook format into a modular `Conv1dModel` function for compatibility with the LUCID preprocessing pipeline. It uses two Conv1D layers followed by dense layers, with the final layer adapted for binary output.

Each model was independently trained, validated, and tested on three different datasets: CICDDoS2019-UDP, BoNeSi+Simu5G, and Mixed Simu5G. The data was split into training, validation, and test sets using the LUCID parser's default configuration, following an 80-10-10 split.

2) MEC Application: To enable real-time evaluation, models were integrated into a custom MEC application (`PredictMec`) in Simu5G. The application predictions were triggered at fixed intervals (30s, 45s, 60s). This choice was driven by two factors: (i) the current Simu5G MEC framework triggers Python-based inference synchronously, making it more practical to schedule predictions at fixed simulation times, and (ii) fixed intervals ensured consistent and comparable measurements across runs. More advanced triggering mechanisms are possible, such as dynamic detection windows that adapt to sudden spikes or anomalous trends in network load. While such an approach would be more responsive in operational settings, it was left for future work in order to focus on demonstrating the end-to-end feasibility of the testing framework.

Three 60-second simulation scenarios were run:

1) **Baseline (no attack)**: Validates normal performance.
2) **Attack only**: Measures network stress under DDoS without defense.
3) **Attack + MEC prediction**: Evaluates detection and mitigation impact.

All scenarios used the same network setup, with minor variations. For the MEC implementation, the `MecAppBase` class in the Simu5G source was extended to include `PredictMec`, which handles scheduled predictions and UE communication. MEC ran the model as a Python script at the edge. Simulation parameters are shown in Table V.

Due to simulator limitations, packets exceeding specific data rates were fragmented and dropped during the parsing process. To avoid this, attack traffic was capped. Prediction windows were implemented to limit analysis to recent packets. Since traffic was captured at the router, manual filtering was required to exclude irrelevant flows (e.g., remote server traffic). A more robust setup would capture only inbound packets.

The attack mitigation mechanism was implemented as an increase in the packet send interval for suspected attackers rather than full packet blocking. This approach was selected to minimize invasive modifications to the Simu5G forwarding pipeline while still enabling a measurable reduction in attack traffic; it also allowed for testing the model's full detection loop without risking simulator instability from mid-flow packet drops. This simplified mitigation does not clear queued or in-flight packets. Future work will implement more robust, dynamic, and load-responsive blocking strategies at the gNodeB or UPF level to better reflect real-world operational defenses.

Resource usage (CPU, memory, runtime) was logged for each prediction. These metrics provide insight into the efficiency of different models and MEC configurations.

[2]https://github.com/doriguzzi/lucid-ddos

[3]https://github.com/mohak1/Detection-and-Classification-of-Distributed-DoS-Attacks-using-Machine-Learning

TABLE V. SIMULATION PARAMETERS USED FOR EACH CONFIGURATION

Simulation Parameter	Baseline – No Attack	Baseline – With Attack	Experiment – With Prediction and Blocking
# UEs	8	8	8
# VidUEs	2	2	2
# malUEs	0	5	5
# malVidUEs	0	5	5
UE messageLength	uniform(16B, 512B)	uniform(16B, 512B)	uniform(16B, 512B)
UE sendInterval	exponential(1s)	exponential(1s)	exponential(1s)
videoServer sendInterval	0.1s	0.1s	0.1s
videoServer PacketLen	1024 B	1024 B	1024 B
videoServer videoSize	200 MiB	200 MiB	200 MiB
malUE messageLength	1024 B	1024 B	1024 B
malUE sendInterval	exponential(0.09s)	exponential(0.09s)	exponential(0.09s)
malVidUE messageLength	2048 B	2048 B	2048 B
malVidUE sendInterval	exponential(0.09s)	exponential(0.09s)	exponential(0.09s)
Prediction Times	60s	60s	30s, 45s, 60s
Prediction Time Window	10s	10s	10s

VI. RESULTS

This section presents results on model performance during training, generalization to new datasets, and the impact of MEC-based deployment on network behavior.

A. Model Performance on CICDDoS2019-UDP

Both the LUCID and Mohak CNN models were first evaluated on the CICDDoS2019 UDP-flood test set:

- LUCID CNN achieved a validation accuracy of 96.98% and an F1-score of 0.9686. The confusion matrix confirms balanced performance with most benign and malicious packets correctly classified.
- Mohak CNN reached 94.99% validation accuracy with an F1-score of 0.9470.

Results confirmed the models' effectiveness on the UDP-flood traffic. Table VI summarizes the results and shows that the Mohak CNN model achieved a higher F1-score and accuracy when classifying the UDP traffic from the CICDDoS2019 dataset compared to the entire mixed dataset, likely due to simpler features and clearer patterns. In contrast, the LUCID CNN did not improve on the UDP data, suggesting it was less able to leverage simpler patterns or fewer features.

B. Generalization to New Datasets

Models were trained, validated, and tested separately on the BoNeSi+Simu5G and Mixed Simu5G datasets. The models exhibited varying levels of generalization performance on these new datasets.

LUCID CNN classified all traffic as benign in both new datasets (F1-score: 0), showing poor generalization. This indicates that the model tuning was overly specialized to the original CICDDoS2019 dataset features, limiting its ability to generalize to the different traffic characteristics and attack patterns in the simulated datasets. Consequently, LUCID CNN was not considered further for MEC deployment.

The Mohak CNN also performed poorly on the BoNeSi+Simu5G dataset (F1-score: 0.0055, accuracy: 49.2%), but in contrast, it demonstrated strong generalization capability when evaluated on the Mixed Simu5G dataset (F1-score: 0.8976, accuracy: 88.5%). The balanced confusion matrix further supports its effectiveness. This suggests that the features learned from CICDDoS2019-UDP were sufficiently general to detect attacks in a dataset generated by Simu5G, with high accuracy. While the BoNeSi+Simu5G dataset, specifically Bonsei DDoS traffic, was significantly different in features from the CICDDoS2019 dataset, this caused the model to perform poorly.

A summary of these findings is provided in Table VII.

C. MEC Impact on Network Behavior

This section present the measured network metrics for each simulation run, as well as the performance of the MEC application hardware.

1) Network Load Results: The average incoming data rate at the UPF varied slightly between each simulation run, as shown in Table VIII. However, this measurement was the mean over 60s, and doesn't reflect MEC's performance.

Figure 3a shows the expected baseline reading of the incoming data rate to the UPF with no attack. Figure 3b shows the same reading at the UPF, but with the incoming attack starting to ramp up at about 8s. The Mohak model setup (Figure 4) shows a slight decrease in network load. By comparing the incoming data rate in the first 30 seconds and the last 30 seconds, the data rate decreased from 623675.2 bps to 601692 bps.

2) MEC Behavior, Prediction Times and HW Resource Usage: As shown in Table VIII, all runs resulted in some false positives. Even in the baseline scenario with no DDoS attack traffic, benign UEs were incorrectly identified as malicious at the 60-second prediction. Additionally, during experimental runs, some malicious UEs were repeatedly flagged across multiple prediction intervals, meaning they were detected as malicious more than once. This double detection explains why, for example, the Mohak experiment shows a total of 15 predictions despite the combined count of true positives and false positives being only 11.

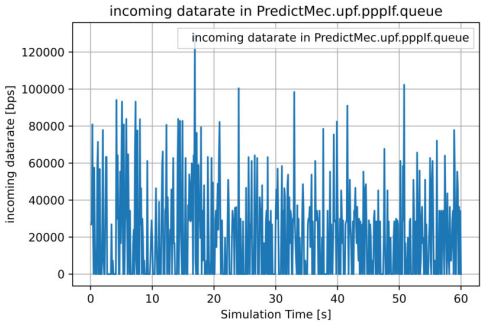
Table IX presents analysis time and hardware usage for each ML model. These results serve as a useful reference for future researchers comparing MEC implementations and ML models.

TABLE VI. VALIDATION PERFORMANCE OF MOHAK AND LUCID MODELS ON CICDDOS2019-UDP

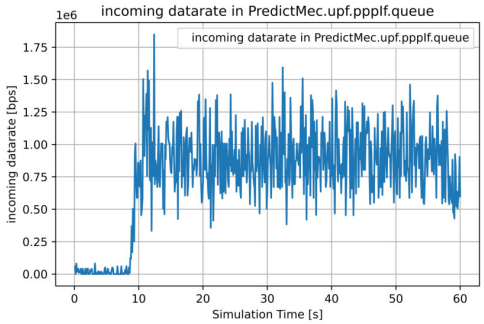
Model	Evaluation Scenario	Dataset	F1-score	Accuracy
Mohak	Expected Performance	CICDDoS2019	0.7131	0.8392
Mohak	Actual Performance	CICDDoS2019-UDP	0.9470	0.9499
Lucid	Expected Performance	CICDDoS2019	0.9939	0.9947
Lucid	Actual Performance	CICDDoS2019-UDP	0.9686	0.9698

TABLE VII. PREDICTION PERFORMANCE OF TRAINED MODELS ON NEWLY GENERATED SIMULATED DATASETS

Model	Dataset	F1-score	Accuracy
Mohak	bonesi + Simu5G	0.0055	0.4922
Mohak	Mixed Simu5G	0.8976	0.8846
Lucid	bonesi + Simu5G	0	0.5123
Lucid	Mixed Simu5G	0	0.3790



(a) The baseline reading on the incoming data rate on the PPP interface on the UPF, with no attack ongoing.



(b) The baseline reading, with attacking UEs, on the incoming data rate on the ppp-interface on the UPF.

Fig. 3. The incoming data rate at the UPF with no predictions during the simulation

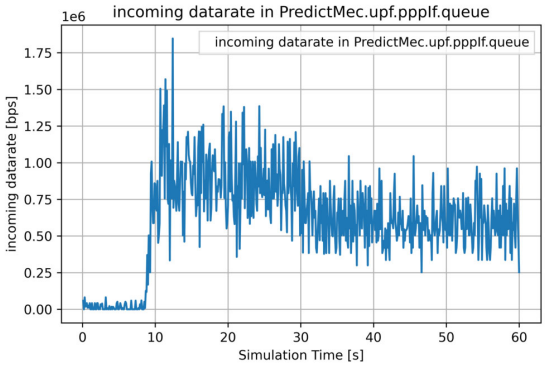


Fig. 4. The incoming data rate at the UPF with predictions at 30, 45 and 60 seconds. The Mohak model is used to predict

TABLE VIII. MEC PREDICTION AND NETWORK RESULTS

Result	Baseline (No Attack)	Baseline (With Attack)	Experiment (Mohak)
UE Throughput [Mean]	1513.3 bps	2286.6 bps	929 bps
VidUE End-to-End Delay [Mean]	123 ms	121 ms	120 ms
UPF Incoming Data rate [Mean]	20.39 kbps	766.42 kbps	612.63 kbps
Total Predictions	5	8	15
True Positives	0	4	6
False Positives	5	4	5
False Positive Rate (%)	100	50	45

TABLE IX. THE MEC'S AND ML MODEL'S PREDICTION TIMES AND HARDWARE RESOURCE USAGE

Result	Experiment (Mohak)
1st Prediction Time [s]	10
2nd Prediction Time [s]	13
3rd Prediction Time [s]	14
CPU Usage	95–100%
Memory Usage [GB]	0.640

VII. DISCUSSION

A. Dataset Generation and Utility

Two datasets were generated to evaluate whether synthetic traffic created in simulation can realistically be used not only for offline model training, but also for operational deployment testing in a MEC-enabled 5G environment. The first dataset combined benign Simu5G traffic with BoNeSi-generated high-rate UDP flood attacks. The second, "Mixed Simu5G", was created entirely within Simu5G, with both benign and malicious traffic originating from simulated UEs.

An important objective of this stage was to investigate whether synthetic datasets generated within a 5G simulator could closely replicate the key traffic characteristics of the benchmark CICDOS2019 dataset (filtered to UDP traffic), such that the resulting detector performance would be comparable. In this sense, the dataset creation process itself is an integral part of the proposed methodology, as it enables a unified framework where datasets for both training and realistic deployment evaluation can be produced under controlled, reproducible conditions.

In the presented experiments, the Mixed Simu5G dataset achieved behavior broadly consistent with CICDDoS2019-UDP for one of the models tested (Mohak CNN), indicating that the synthetic generation process can yield usable approximations of benchmark datasets. In contrast, the BoNeSi+Simu5G dataset led to poor detection results, suggesting that differences in packet timing, structure, or statistical feature distributions can significantly affect transferability.

These outcomes both validate the feasibility of the approach and highlight the need for more advanced synthetic traffic generation methods to consistently match benchmark datasets.

B. Generalization of Detection Models

The LUCID CNN performed well on the CICDDoS2019-UDP dataset but failed on the new synthetic traffic, indicating model overfitting to the original training dataset. Mohak's CNN also failed on BoNeSi+Simu5G but succeeded on the Mixed Simu5G dataset. This highlights a key point: models trained on one dataset may not perform very well even when trained on a similar dataset.

It is important to clarify that the generalization experiments here are not simply a conventional cross-dataset test, but rather an assessment of how well benchmark-trained models respond to datasets synthetically generated within the simulation environment to resemble the benchmark's statistical and structural properties. In other words, the goal was to test whether the synthetic dataset generation process could create realistic traffic patterns that both reflect benchmark characteristics and can be used for operational deployment evaluation. While the Mixed Simu5G dataset showed such alignment for the Mohak model, the BoNeSi+Simu5G dataset did not, underlining the variability in reproducing feature distributions and motivating future optimization of the dataset generation strategies.

C. MEC Prediction and Implementation Gaps

MEC integration showed potential but suffered from issues:

- Predictions were based on static time intervals rather than dynamic traffic features.
- Some UEs were misclassified due to overly simplistic criteria (e.g., only packet length).
- Repeated predictions occurred due to queued messages in INET's UdpBasicApp.

Comparing the ML results to the MEC predictions reveals a significant disparity. The ML models performed much worse when deployed in the MEC, due to several underlying factors. First, the prediction accuracy improved, and the UPF bitrate decreased when the packet sizes of benign UEs were reduced or those of malicious UEs were increased. This indicates that the model primarily relied on packet length as its distinguishing feature. Second, including source–destination bitrate in the model would likely improve detection, due to the fragmentation behavior of IPv4. Finally, as noted in Section VI, the MEC repeatedly flagged double-flagged malicious UEs, likely due to the INET `UdpBasicApp`, which queues messages. Even after a UE was silenced, pending messages continued to be sent. This issue could be addressed by blocking UEs at the base station or UPF level rather than relying on silencing.

D. Simulation Constraints

The simulation itself has significant limitations as the number of devices, benign or otherwise, had to be kept fairly low due to hardware constraints. Especially the malicious UEs required a lot of CPU to run effectively. In this work, the botnet consisted of only ten devices, whereas real botnets are typically much larger.

Furthermore, rather than outright blocking malicious devices, the MEC implementation mitigated their impact by increasing the UEs' send intervals to very high values, effectively suppressing attack traffic within the current framework. This practical simplification was chosen to demonstrate the method's feasibility. However, it reduces realism since real MEC-based defenses would typically rely on adaptive prediction intervals and more robust mitigation strategies, such as selective blocking or traffic filtering at the gNB/UPF. Exploring these alternatives represents an important direction for future work.

VIII. Conclusions and Future Work

This work presented a simulation-driven approach for evaluating the transferability and operational feasibility of ML-based DDoS detection models in realistic 5G MEC environments. In the defined experiments, the dataset produced with Simu5G aligned reasonably well with one of the evaluated CNN models, supporting effective detection, whereas the BoNeSi-generated dataset differed significantly from the CICDDoS2019 benchmark used for model training, leading to reduced performance. This outcome illustrates both the potential and the challenges of generating synthetic traffic that accurately reflects established benchmark datasets.

Future optimization will focus on creating synthetic datasets that are suitable for evaluating models under deployment conditions while closely resembling benchmark datasets in their key statistical and structural properties. To achieve this, future work shall explore advanced traffic-generation and feature-modeling techniques, complemented by comparison metrics such as statistical similarity indices, and feature-distribution matching to ensure equivalence to benchmark datasets.

In parallel, refinements to the current implementation, particularly in attack mitigation mechanisms, will be pursued to enable dynamic and fine-grained responses. Such enhancements will allow a more precise assessment of mitigation effects on critical 5G performance metrics.

While this study focused on high-rate UDP flooding as a test case, the methodology developed here can be applied to other attack types. Future extensions will include, a more comprehensive attack scenario coverage and the generation of synthetic dataset variants for low-rate and application-layer DDoS traffic, enabling the evaluation of ML-based detection models across a broader spectrum of threat scenarios in realistic 5G deployments.

Overall, the study confirms the feasibility of using a controlled simulation-based framework to assess not only the performance of ML-based DDoS defenses but also their readiness for operational deployment, providing a reproducible method for evaluating robust, adaptive, and resource-aware network protection strategies.

References

[1] "Record-breaking 5.6 tbps ddos attack and global ddos trends for 2024 q4," *https://blog.cloudflare.com/ddos-threat-report-for-2024-q4/*, 2025.

[2] S. Sinha, "Number of connected iot devices growing 13% to 18.8 billion," *IoT Analytics*, Sep. 2024.
[3] E. Commision, "European action plan on the cybersecurity of hospitals and healthcare providers," Jan 2025.
[4] IBM, "5g use cases that are transforming the world," *IBM*, Mar. 2024.
[5] Taoglas, "What is 5g? exploring its features, benefits, and applications," *Taoglas*, Dec. 2024.
[6] Akamai Technologies, "What is a volumetric attack?" *Akamai*, Feb. 2025.
[7] H. Djuitcheu, T. Shah, M. Tammen, and H. D. Schotten, "Ddos impact assessment on 5g system performance," in *2023 IEEE Future Networks World Forum (FNWF)*, 2023, pp. 1–6.
[8] S. S and S. S, "Ddos detection using ml and deep learning approaches," in *8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 2024, pp. 1–6.
[9] S. Chauhan, P. Byahatti, and S. K. Patel, "Securing networks: Leveraging machine learning for enhanced ddos detection," in *First International Conference on Software, Systems and Information Technology (SSITCON)*, 2024, pp. 1–8.
[10] V. Hnamte, A. A. Najar, H. Nhung-Nguyen, J. Hussain, and M. N. Sugali, "Ddos attack detection and mitigation using deep neural network in sdn environment," *Computers & Security*, vol. 138, 2024.
[11] G. Usha, M. Narang, and A. Kumar, "Detection and classification of distributed dos attacks using machine learning," in *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020*. Springer, 2021, pp. 985–1000.
[12] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for ddos attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, 2020.
[13] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, 2019, pp. 1–8.
[14] S. Hettich and S. D. . Bay, "Kdd cup 1999 data," *The UCI KDD Archive*, 2000.
[15] A. A. Najar and S. M. Naik, "Cyber-secure sdn: A cnn-based approach for efficient detection and mitigation of ddos attacks," *Computers & Security*, vol. 139, p. 103716, 2024.
[16] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.
[17] Y. Ma, L. Liu, Z. Liu, F. Li, Q. Xie, K. Chen, C. Lv, Y. He, and F. Li, "A survey of ddos attack and defense technologies in multi-access edge computing," *IEEE Internet of Things Journal*, 2024.
[18] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, "Protocol-based deep intrusion detection for dos and ddos attacks using unsw-nb15 and bot-iot data-sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2021.
[19] "Netcentric security project," *Deutsches Forschungszentrum fuer Kuenstliche Intelligenz*, 2006.
[20] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, 2020.
[21] G. Nardini, A. Noferi, P. Ducange, and G. Stea, "Exploiting simu5g for generating datasets for training and testing ai models for 5g/6g network applications," *SoftwareX*, vol. 21, p. 101320, 2023.