5GDAD: A Deep Learning Approach for DDoS Attack Detection in 5G P4-based UPF

Rana Abu Bakar, Faris Alhamed, Piero Castoldi, Andrea Sgambelluri
Scuola Superiore Sant'Anna
Pisa, Italy
rana.abubakar@santannapisa.it

Juan Jose Vegas Olmos
NVIDIA
Israel
juanj@nvidia.com

Filippo Cugini, Francesco Paolucci
CNIT
Pisa, Italy
francesco.paolucci@cnit.it

***Abstract*—The fast-paced growth of 5G networks, along with the emergence of 6G technology, has emphasized the crucial importance of strong security measures to safeguard communication infrastructures. A key security issue in 5G data networks is Distributed Denial-of-Service (DDoS) attacks, which specifically target the GTP-based protocol which is a significant threat. However, network telemetry data provides a rich source of information about the nature of network traffic, which can be used to detect and predict DDoS attacks. We propose a novel framework for collecting and processing large amounts of telemetry data in 5G networks leveraging state-of-the-art technologies, including data-plane programmability in P4-based User-Plane Function (UPF) and Data Processing Unit (DPU). Furthermore, we propose an anomaly-detection method for performing live deep learning analysis on network traffic using a Convolutional Neural Network (CNN) to detect DDoS attacks. Our results demonstrate the effectiveness of our framework, achieving an impressive 98.6 % accuracy and 98% F1-score.**

***Index Terms*—Network, DDoS, Detection, P4Lang, Telemetry, Traffic Analysis**

I. Introduction

With the rapid growth and deployment of telecommunication networks, there is a pressing need to address the escalating network security attacks [1]. DDoS attacks are a major threat to the security of 5G networks. These malicious attacks aim to disrupt network services by overwhelming network resources with excessive traffic. Developing effective defense mechanisms is crucial to mitigate the risk of DDoS attacks and ensure accessibility of network services [2]. The increasing prevalence and severity of DDoS attacks pose significant challenges to the security and stability of 5G networks [3]. These attacks exploit the vulnerabilities of network infrastructures, causing service disruptions and financial losses for businesses and organizations [4]. To effectively counter these threats, developing robust and proactive defense mechanisms that can detect and mitigate DDoS attacks in real-time is crucial [5].

Before talking about our contributions **??**, we would like to highlight some of the interesting works that are relevant to our research topic. Paolucci et al. [6] explored integrating the P4 language into SDN/NFV networks to address their complexity and heterogeneity. They explored the Programmable Data Plane (PDP) with P4 language in the context of 5G networks to enhance the functionality of many networking aspects, including monitoring, security, network slicing, and traffic engineering.

On the other hand, the authors in [7] proposed a vEPC-vSDP framework to secure communication within virtualized mobile core networks. By virtualizing Software Defined Perimeter (SDP) components, it is possible to establish a zero-trust environment, allowing only authenticated and authorized network elements to access each other. The framework demonstrated resilience against various attacks and effectively protected core network traffic. Furthermore, the authors in [8] proposed a framework for trustworthy Self-Driving Networks (SelfDNs) across multiple domains. They leverage programmable data planes, P4 language, AI, blockchain, and federated learning to enable real-time telemetry collection and automatic translation of policy intents into executable actions. In their work [9], the authors extensively compared attack detection methods in Software Defined Networks (SDN), employing a Mininet test bed to simulate real-world SDN environments. They evaluated various Artificial Intelligence techniques for detecting common attacks on transport and application layers, utilizing an architecture comprising an OpenFlow-based flow collector, an ONOS SDN controller, and pre-processing/detection modules. However, their approach involved extracting a limited set of features from flow data, which may restrict the capacity of analysis and detection abilities. Finally, the DDoS Attack Detection (DAD) described in our previous paper [10] focused on leveraging the capabilities offered by the programmable data plane to build a comprehensive framework for DAD in various cases. This work is also particularly interesting to our research topic due to its use of PDP with the P4 language for collecting and mitigating DDoS attacks. In this paper, we aim to improve the data collection process by introducing preprocessing of telemetry data and using a CNN for the flow analysis. In this paper, we extend our prior work [10] on DDoS attack detection frameworks, particularly in the context of 5G networks, leading to the development of our proposed framework named 5G DDoS Attack Detection (5GDAD). While it is acknowledged that prior research has explored ideas surrounding P4 telemetry data for traffic analysis, our work distinguishes itself through several key contributions. Our framework 5GDAD offers the following contributions:

1) We propose a novel framework for predicting DDoS attacks in 5G networks using P4 telemetry data and CNNs.

2) We develop a CNN model specifically designed to detect and predict DDoS attacks on 5G networks to handle the unique characteristics of P4 telemetry data.
3) We evaluate our framework using real-world 5G traffic data and demonstrate its effectiveness in predicting and mitigating DDoS attacks.

The rest of this paper covers the background, methodology, experimental setup, and analysis techniques employed for DDoS attack detection using P4 telemetry data in 5G networks. The findings, conclusions, and future research directions are also presented.

II. Proposed Architecture

In this section, we present the architecture of our proposed framework and topology in Figure 1. The framework is designed to leverage 5G telemetry data for predicting DDoS attacks using a CNN approach. Our architecture for DDoS attack detection in 5G networks integrates various components to effectively mitigate and respond to DDoS attacks. The detailed flow of the 5GDAD is presented in Figure 2, while the deep learning-based DDoS attack detection algorithmic details are outlined in Algorithm 1.

A. P4-based UPF

The framework implements the UPF within a programmable P4 switch [6], endowing the system with the flexibility of PDP. SmartNiCs integrated into the gNodeB enhance performance by offloading resource-intensive networking functions from the CPU and contribute to improved monitoring capabilities.

B. Telemetry Collector

Telemetry is collected using a Two-Stage P4 telemetry collector [11]. The first stage collects and compresses network telemetry data to reduce bandwidth usage, while the second stage performs rapid packet processing on a dedicated server that runs on a Linux Operating System, utilizing the Fastcapa framework [12]. This stage examines telemetry packets, extracts relevant information, and efficiently stores it in InfluxDB.

C. Telemetry Dataset

The transformed telemetry dataset is used to train a CNN-based detector, which learns to identify patterns indicative of DDoS attacks. Once trained, the detector can analyze telemetry flows in real-time and trigger alerts upon detecting potential attacks.

D. Database

The framework includes a database for storing flow information and visualization tools for presenting performance metrics to human operators. Including these components gives the framework additional functionality.

Our proposed framework incorporates programmable switches, SmartNICs, and a two-stage telemetry collection process, culminating in a CNN-based predictive system for identifying DDoS attacks in a 5G network. The system's analytical capabilities are enhanced by an integrated database and visualization tools, which allow for in-depth network analysis.

III. Implementation of 5GDAD

Our experimental setup comprises four physical servers, each equipped with two Intel(R) Xeon(R) Gold 6238R processors, 256 GB of RAM, and 100 Gbps Nvidia DPU network interface cards. One server hosts a group of 4 Docker containers representing network hosts, including 2 malicious nodes. A second server runs two 5G UPFs in a P4 software switch for traffic forwarding, telemetry generation, and exportation, while a third server handles telemetry aggregation. We conducted experiments using Docker containers to evaluate the effectiveness of our proposed DDoS detection system. Each container was configured with a P4 switch, enabling us to control the behavior of malicious nodes responsible for DDoS attacks. The P4 switches were programmed with flow rules to manage network traffic, leveraging existing GTP tunnels within the simulated 5G network for communication channels between the malicious nodes and the gNodeB. Our experiments created various attack scenarios by configuring the malicious nodes to generate DDoS attacks and adjusting the intensity and duration to simulate real-world scenarios. DDoS flooding traffic was directed toward the gNodeB using GTP tunnels. Our DDoS detection system, based on CNN, analyzes telemetry data from P4 switches to identify attacks by monitoring network traffic patterns, packet flow, rates, and other metrics. We evaluated the performance of our DDoS detection system with another state-of-the-art system, 5GAD2022 [13] dataset.

A. Telemetry Collection

Telemetry collection uses Nvidia Bluefield-2 DPU network interface cards. DPU can access the stream of telemetry packets generated by the network's P4 switches and perform unrestricted processing and modifications. In the initial version of the system, a telemetry packet carrying a telemetry record is generated by the p4 switch for each observed packet. These telemetry packets can be forwarded through the control plane by incorporating valid network layer headers. Given the average packet size at ESnet [14] of approximately 1512B, it is crucial to note that generating 72B telemetry digest packets instead of capturing and processing the entire packets significantly reduces the data rate of the original traffic. With a packet length of 1512 bytes (1440 payload), 100 Gbps of traffic was generated. The telemetry packets were produced at a rate of 8.8 million per second, corresponding to a telemetry data rate of 8 Gbps.

B. DPDK-based telemetry packet process

The packets are initially received from the wire and then passed to the open-source software of the Fastcapa [12]. Fastcapa, implemented using the DPDK (Data Plane Development Kit), facilitates reading telemetry from the wire-speed and directing it to the appropriate Kafka topic based on each flow. Before reaching the Kafka topics, the packets undergo preprocessing, which may include sampling and histogramming, to achieve fast and scalable telemetry packet processing while maintaining packet ordering at 100 Gbps. For example, for 100 Gbps traffic

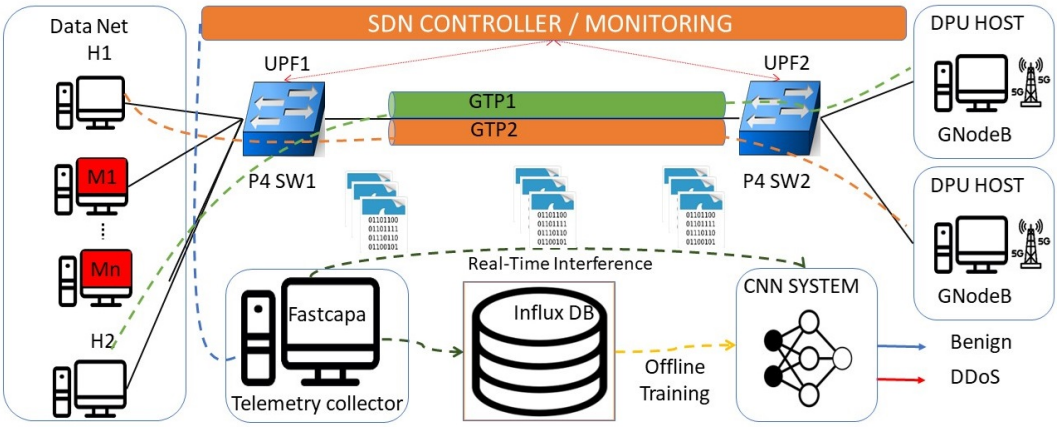


Fig. 1: Proposed Framework Network Topology

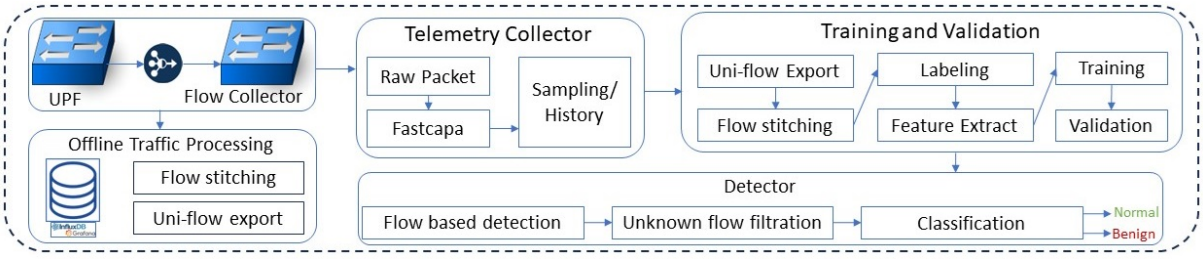


Fig. 2: Detailed 5GDAD Architecture for DDoS Detection

with 1512B MTU, more than 8 million telemetry packets will be generated per second by each P4 switch. Our telemetry processing system takes 120 ns to process each packet. With a DPDK-based telemetry worker system, we use up to 5 CPU cores to process telemetry packets within a single pipeline designed to handle 100 Gbps. A modern multi-core server with Smart Network Interface Card (SmartNiC) can also handle telemetry packets from multiple 100 Gbps links. The Fastcapa server provides five types of DPDK workers on each CPU core to handle tasks such as pulling packets, ACL control, flow handling, Kafka topics handling, TCP connection maintenance, and CSV file generation.

C. Dataset

The dataset for evaluating the CNN-based DDoS attack detection system consists of 420,000 samples of normal traffic and 520,950 samples of malicious traffic. The dataset was collected from a 5G network testbed and included traffic from various devices, including DataNET hosts, GNodeBs, and UPFs. The normal traffic samples were collected from outgoing traffic captures from all nodes connected to the network design. The normal traffic dataset consists of 420,000 samples of 24 hours of traffic. The traffic was collected from four containers acting as DataNET hosts in the 5G data center network. The traffic captures were separated into 1, 2, 6, and 12 hours, resulting in 20,000 samples per duration. The malicious traffic dataset comprises 520,950 samples of 24 hours of traffic. The traffic was generated by launching various DDoS attacks against gNodeB with the four docker containers acting as DataNET hosts. The normal traffic data was collected using a telemetry collector. The malicious traffic data was generated using tools specifically designed for launching DDoS attacks tools [15]. The traffic captures were stored in a CSV for easy processing and analysis. The dataset was split into training, validation, and testing sets. The training set was used to train the CNN-based DDoS attack detection system. The validation set was used to tune the hyperparameters of the system. The testing set evaluated the system's performance on unseen data. The feature selection in the context of CNN-based 5G-based DDoS detection involves various techniques [16]–[20]. We consider the following key features source and destination IP addresses, ports, protocol type, TTL, total length, TCP flags, flow duration, total packets, bytes, packet length mean, flow byte, packet rates, flow inter-arrival time mean, Window size, flags, and retransmissions.

These features, selected through rigorous analysis, contribute to accurately identifying DDoS attacks in 5G networks and serve as the basis of our CNN model training and evaluation.

D. CNN-based DDoS Detection Model

The architecture consists of three convolutional layers for spatial feature extraction, two pooling layers for spatial down-sampling, and three fully connected layers for high-level abstraction, working together to capture intricate patterns within network traffic data. Each layer is crucial in processing and extracting relevant features from the input traffic data. The model experiences rigorous hyperparameter tuning, optimizing parameters by using learning rate, batch size, and optimizer settings using grid search and random search techniques, to ensure superior

Algorithm 1 5GDAD: CNN-based DDoS Attack Detection Algorithm

Require: Telemetry data from UPF and other network devices
Ensure: DDoS attack detection results
Capture telemetry data from UPF and other network devices
Extract raw telemetry data packets from UPF and other network devices
Perform initial preprocessing steps (packet dissection, header extraction)
Reshape packet data into a suitable format for CNN input, converting to a sequence of 8-bit integers
Further preprocess the data to enhance spatial representation into a three-dimensional array
Normalize and standardize the preprocessed data to ensure consistent input ranges (using Min-Max scaling or Z-score normalization)
Padding the preprocessed data to ensure uniform sample lengths and associate flow-level labels with the preprocessed samples
Return the preprocessed telemetry data with relevant features and labels
Train 5GDAD CNN model using labeled telemetry data
Classify telemetry data as normal or malicious based on the 5GDAD CNN model
Detect anomalies in telemetry data by analyzing deviations from expected patterns
if Anomalies are detected **then**
    Mitigation Process is started
    if Anomalies exceed predefined thresholds **then**
        Trigger alerts to network operator
    else
        Log anomalous events for further investigation
    end if
else
    No action required
end if

accuracy, precision, recall, and F1 score compared to conventional approaches. Using binary cross-entropy, the model is trained on a labeled dataset consisting of normal and malicious traffic samples, involving iterative weight and bias updates to minimize a chosen loss function. Validation using a separate set ensures its generalization capability across different network conditions.

1) Network Traffic Preprocessing: Network Traffic Preprocessing Algorithm 2 outlines the preprocessing steps for network traffic data. This algorithm aims to preprocess raw network traffic data, making it suitable for further analysis and classification by our machine learning model. The algorithm takes as input the network traffic trace (NTT), flow-level labels (L), a time window (t), and the maximum number of packets per sample (n). It outputs a list of labeled samples (E), containing preprocessed network traffic data.

Algorithm 2 Network Traffic Preprocessing Algorithm

Require: Network traffic trace (NTT), flow-level labels (L), time window (t), max packets per sample (n)
Ensure: List of labeled samples (E)
procedure PREPROCESSING(NTT, L, t, n)
    Initialize an empty dictionary to store preprocessed samples (E)
    Initialize the current time window start time (τ) as -1
    for each packet pkt in NTT **do**
        Extract flow identifier (id) from the packet
        Calculate the time difference between the current packet's timestamp and the current time window start time (Δt)
        if $\tau == -1$ or $\Delta t > t$ **then**
            Update the current time window start time (τ) to the current packet's timestamp
            Initialize an empty list to store packet features for the current flow within the current time window ($E[\tau, id]$)
        end if
        if Length of $E[\tau, id] < n$ **then**
            Preprocess the packet's data ($pkt.data$) to obtain packet features ($pkt\_features$) using the PREPROCESSING(($pkt.data$)) function
            Append the preprocessed packet features ($pkt\_features$) to the list of packet features for the current flow within the current time window ($E[\tau, id]$)
        end if
    end for
    Normalize and pad the preprocessed samples using the NORMALIZEANDPADSAMPLES((E)) function
    Apply the corresponding flow-level label ($L[id]$) to each sample (e) in the preprocessed samples (E)
    return the preprocessed samples (E)
end procedure

2) CNN Architecture: The CNN architecture described in Table I uses a sequence of layers to extract features and make classifications. The architecture is optimized to handle packets of length 1512 bytes, which are commonly used packet sizes in network traffic analysis. At the core of the CNN architecture are convolutional layers, which play a crucial role in capturing local patterns within the input data. Learnable filters are used in these layers to extract features that distinguish between normal and malicious traffic. By employing three convolutional layers, the network model can automatically learn hierarchical features, progressively capturing more abstract representations of the data. Activation functions, Rectified Linear Unit (ReLU) and Leaky ReLU, introduce non-linearity to the model, enabling it to learn complex relationships between 5G network traffic features.

ReLU is defined as:

$$\text{ReLU}(x) = \max(0, x)$$

Leaky ReLU, which allows a small, non-zero gradient

when the input is negative, is defined as:

$$\text{Leaky ReLU}(x, \alpha) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}$$

The CNN model is optimized using a categorical cross-entropy loss function, which is well-suited for multi-class classification tasks. Mathematically expression for the cross-entropy loss is:

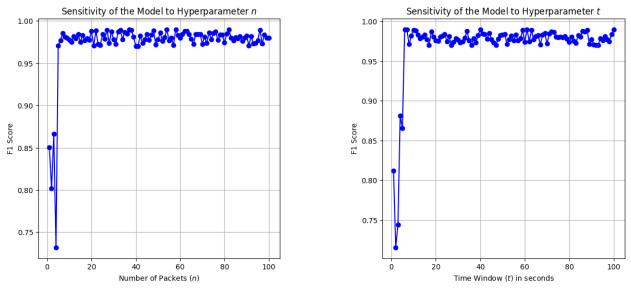
$$\text{Cross-Entropy Loss} = -\sum_{i} \text{True Class}_i \cdot \log(\text{Predicted}_i)$$

where True Class_i represents the true label of the i-th class, and Predicted_i represents the predicted probability of the i-th class.

TABLE I: CNN Model Architecture

Layers	Filters	Kernel Size	Activation Function
Convolutional	32	3×3	ReLU
Max Pooling	-	2×2	-
Convolutional	64	3×3	ReLU
Max Pooling	-	2×2	-
Convolutional	128	3×3	ReLU
Flatten	-	-	-
Fully Connected	128	-	ReLU
Fully Connected	64	-	ReLU
Output	1	-	Sigmoid

3) Model Evaluation: The model's performance is evaluated using standard metrics of accuracy, precision, recall, and F1-score. We performed fine-tuning of hyperparameters by adjusting the learning rate and batch size to achieve optimal performance. The model is also optimized using backpropagation and gradient descent-based optimization algorithms. In this sensitivity analysis, Figure 3 showcases the impact of varying the hyperparameter n, representing the number of packets, on the F1 score of our CNN-based DDoS detection model. The blue line illustrates the model's F1 score, revealing a consistent increase in performance as n grows, leveling off after reaching a saturation point. The analysis suggests that, for our model, higher n values contribute to improved accuracy, with stability achieved beyond a certain threshold. This balance is crucial for optimizing the trade-off between detection accuracy and computational efficiency, a key consideration in real-world applications.



(a) F1 Score vs. Number of Packets

(b) F1 Score vs. Time Windows

Fig. 3: The model's sensitivity to the hyperparameter n for F1 score.

E. Experimental Results

The performance metrics were measured during our experiments, and the results are presented in Figure 4. Figure 4 illustrates the performance metrics of the proposed CNN model for DDoS attack detection in 5G networks. The model achieves high accuracy (98.6%), precision (99.9%), recall (97%), and F1 score (98%). Additional metrics include the normal detection rate (98.6%), attack detection rate (98%), true negative rate (TNR - 98%), true positive rate (TPR - 98.3%), false negative rate (FNR - 0.022), and false positive rate (FPR - 0.018). Figure 4 displays the accuracy and loss of the CNN model throughout training on the P4 telemetry data. The model achieves an accuracy of over 98.6% after nine epochs of training, with a loss of less than 0.001. Figure 5 is a heatmap representation of a confusion matrix depicting the performance of our CNN model. The matrix shows the counts of TN, FP, FN, and TP.

Figure 6 illustrates the start and end instants of the DDoS attack (e.g., with vertical line). The attack is indicated by a sudden increase in traffic, followed by a sharp decrease due to intervention from the control plane. The traffic then goes below its pre-attack level as the controller reconfigures the network to block malicious traffic. Finally, the traffic recovers to its original level as the controller finishes the network configuration. These results demonstrate that our DDoS detection system is highly accurate and effective in mitigating attacks while maintaining low latency and packet loss. The system also utilizes resources efficiently, ensuring optimal performance. This table II compares key performance metrics between our telemetry dataset and another 5G dataset [13]5GAD2022. The telemetry dataset demonstrates superior performance across all metrics, showcasing the robustness and efficacy of our proposed DDoS detection system.

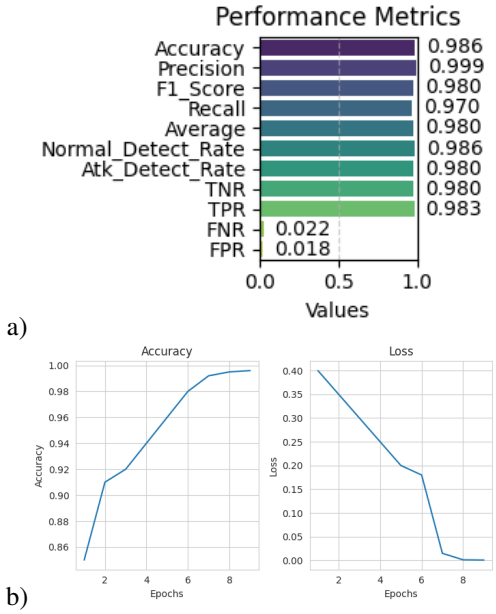


Fig. 4: Performance Metrics and Loss: a) Performance metrics of the CNN model for DDoS attack detection. b) Accuracy and loss of the CNN model throughout training on P4 telemetry data.

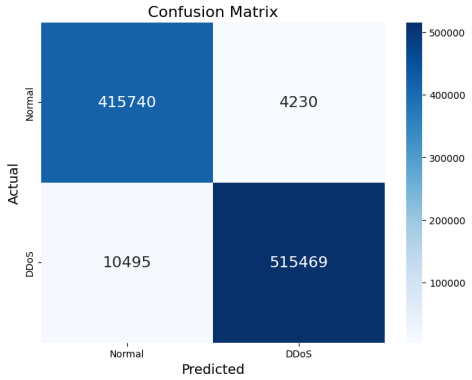


Fig. 5: Confusion matrix showing the classification performance of CNN model for distinguishing between "Normal" and "DDoS" instances.

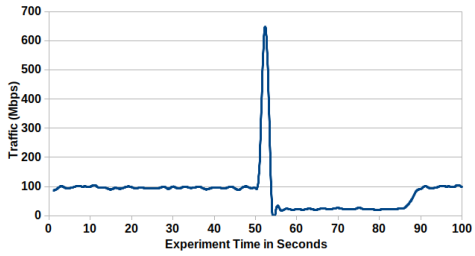


Fig. 6: Traffic pattern captured before, during, and after a DDoS attack.

TABLE II: Performance Metrics Comparison

Metric	Telemetry Dataset	[13]5GAD2022 Dataset
Accuracy	98.60%	98.49%
Precision	99.9%	99.14%
Recall	97.0%	96.72%
F1-score	98.00%	97.90%

IV. Conclusion

This paper presented a novel and effective approach for DDoS attack detection in 5G networks utilizing P4 telemetry data. The proposed method used the P4 programmable data plane and the rich information embedded in P4 telemetry data to identify and mitigate DDoS attacks accurately. The experimental results demonstrated the effectiveness of the proposed method, achieving an accuracy of 98.6%, F1 score 98%, and a precision of 99%. These results highlight the potential of the proposed method for practical DDoS detection in 5G networks. As future work, we aim to develop a hybrid P4-based Native AI 6G UPF with a Bluefield-2 DPU to optimize hardware offloading for AI tasks and enhance security against other types of attacks in 5G and 6G Networks.

Acknowledgment

This work has been funded by the European Commission Horizon Europe SNS JU DESIRE6G (GA 101096466) Project and work carried out within the Department of Excellence in Robotics and Artificial Intelligence of Scuola Superiore Sant'Anna.

References

[1] M. El Rajab, L. Yang, and A. Shami, "Zero-touch networks: Towards next-generation network automation," *Computer Networks*, p. 110294, 2024.

[2] V. A. Shirsath, M. M. Chandane, C. Lal, and M. Conti, "Sparq: Syn protection using acyclic redundancy check and quartile range on p4 switches," *Computer Communications*, vol. 216, pp. 283–294, 2024.

[3] A. Javadpour, F. Ja'fari, T. Taleb, and C. Benzaïd, "Reinforcement learning-based slice isolation against ddos attacks in beyond 5g networks," *IEEE Transactions on Network and Service Management*, 2023.

[4] D. Javaheri, S. Gorgin, J.-A. Lee, and M. Masdari, "Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives," *Information Sciences*, vol. 626, pp. 315–338, 2023.

[5] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," *Computers & Electrical Engineering*, vol. 98, p. 107716, 2022.

[6] F. Paolucci, F. Cugini, P. Castoldi, and T. Osiński, "Enhancing 5G SDN/NFV edge with P4 data plane programmability," *IEEE Network*, vol. 35, no. 3, pp. 154–160, 2021.

[7] Y. Bello, A. R. Hussein, M. Ulema, and J. Koilpillai, "On sustained zero trust conceptualization security for mobile core networks in 5g and beyond," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1876–1889, 2022.

[8] O. Hireche, C. Benzaïd, and T. Taleb, "Deep data plane programming and ai for zero-trust self-driven networking in beyond 5g," *Computer Networks*, vol. 203, p. 108668, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621005442

[9] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "Sdn-based architecture for transport and application layer ddos attack detection by using machine and deep learning," *IEEE Access*, vol. 9, pp. 108 495–108 512, 2021.

[10] F. Musumeci, V. Ionata, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-learning-assisted ddos attack detection with p4 language," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[11] F. Alhamed, D. Scano, P. Castoldi, J. J. Vegas Olmos, I. Vershkov, F. Paolucci, and F. Cugini, "P4 telemetry collector," *Computer Networks*, vol. 227, p. 109727, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912862300172X

[12] A. Metron. (2018) Fastcapa. [Online]. Available: https://github.com/apache/metron/tree/master/metron-sensors/fastcapa

[13] C. Coldwell, D. Conger, E. Goodell, B. Jacobson, B. Petersen, D. Spencer, M. Anderson, and M. Sgambati, "Machine learning 5g attack detection in programmable logic," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 1365–1370.

[14] Z. Liu, B. Mah, Y. Kumar, C. Guok, and R. Cziva, "Programmable per-packet network telemetry: From wire to kafka at scale," in *Proceedings of the 2021 on Systems and Network Telemetry and Analytics*, 2020, pp. 33–36.

[15] 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.

[16] M. Wang, Y. Lu, and J. Qin, "A dynamic mlp-based ddos attack detection method using feature selection and feedback," *Computers & Security*, vol. 88, p. 101645, 2020.

[17] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[18] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, vol. 132, pp. 81–98, 2018.

[19] A. Boualouache, A. A. Jolfaei, and T. Engel, "Multi-process federated learning with stacking for securing 6g-v2x network slicing at cross-borders," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[20] E. Paolini, L. Valcarenghi, L. Maggiani, and N. Andriolli, "Real-time clustering based on deep embeddings for threat detection in 6g networks," *IEEE Access*, 2023.