3D5G-O: DDoS Detection and Defense System of 5G on O-RAN*

Shih-Ting Chiu[1], Fang-Yie Leu[1], Kun-Lin Tsai[2], Heru Susanto†

[1]Tunghai University, Taichung, Taiwan
{g13350005,kltsai,leufy}@thu.edu.tw
[2]Universiti Teknologi Brunei, Center for Innovative Engineering, Brunei
heru.susanto@utb.edu.bn

Abstract

This paper proposes a 5G network slicing intrusion detection scheme, called DDoS Detection and Defense on 5G network for O-RAN (3D5G-O) which integrates machine learning and real-time traffic monitoring techniques to detect and mitigate DDoS attacks within the Open RAN (O-RAN). The 3D5G-O's architecture consists of a Random Forest (RF) classification model, which is deployed within the Service Management and Orchestrator (SMO) of O-RAN to classify packets transmitted from UE to the RAN, and a hybrid detection approach comprising the CuSum algorithm and entropy detection mechanism. Our experimental results show that the classification accuracy of RF classification model achieves 99.98% and the 3D5G-O can effectively mitigate and block DDoS attacks.

Keywords: DDoS, Random Forest, CuSum, Entropy.

1. Introduction

In recent years, people request high-efficiency and high-speed 5G/6G (hereinafter referred to as 5G) network, wishing to enrich their everyday lives. To archive this, network equipment and facilities need to be constantly upgraded. Network Slice [1[2]and Open Radio Access Network (O-RAN) [3[4] have been emerged to this. The former allow network resources to be used more efficiently. Due to its isolation characteristics, users can utilize different network resources at the same time in the same physical network system without interfering with each other. Network slices as denoted by S-NSSAI (Single Network Slice Selection Assistance Information) and slice types can be distinguished by Slice/Service type (SST) [1[2].

* Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 16, December 16-18, 2025, Sapporo , Japan. \space © The copyright of this paper remains with the author(s).
† Corresponding author

In the O-RAN architecture, the Non-Real-Time RIC equipped in the Service Management and Orchestration (SMO) function and the Near-Real-Time RIC enable the deployment of machine learning or artificial intelligence models[3] to autonomously allocate bandwidth and QoS policies for improving the efficiency of base station resource allocation and traffic control capabilities. However, due to the quick advances of network technology, hackers' intrusion techniques and knowledge follow. Therefore, network attacks have never decreased, and DDoS attacks are always around us[5]. In other words, the defense of network systems must be continuously improved. Generally, it is very important to detect attacks early, defend against them early, and mitigate the power of network attacks as soon as possible.

On the other hand, network slices are functionally isolated from each other. If a slice i is attacked and its traffic is heavy, the traffic of other slices, for example $j, i \neq j$, may be still low. The IDS/IPS that monitors the entire network traffic to detect DoS/DDoS attacks on the physical network or components may be unable to explore it. Singh *et al.*[2] mentioned that 5G network slices must be prevented from DoS and DDoS attacks to serve users efficiently. Polese *et al.* [6] found that several limitations related to SMO can be found in literature. First, the defense coordination and deployment of intelligent strategies in SMO has not been fully described, explained and implemented. As a result, when network slices face dynamic attacks, such as DDoS, the solution tends to be with static encryption, lacking intelligent real-time detection and response capabilities.

Therefore, to solve these problems, in this paper, we propose a security mechanism, named DDoS Detection and Defense on 5G network for O-RAN (3D5G-O), that detects DDoS attacks on network slices. The 3D5G-O deploys machine learning techniques, such as the Random Forest (RF) model in the SMO to classify the slice to which a packet entering the base station belongs. Furthermore, it monitors the operational status of all slices in the network in real time by using the CuSum and Entropy algorithms. This allows the rapid identification of the attack sources when an attack occurs, and the attack is then blocked according to QoS rules for each slice to minimize the probability of false positives.

The rent of this paper is structured as follows. Section 2 reviews the relevant literature and background knowledge of this research. Section 3 describes the main system architecture and attack detection methods of the 3D5G-O. Experimental results and functional verification are presented in Section 4. Section 5 concludes this study and addresses our future work.

2. Related Work and Background

In recent years, attack methods and defense strategies against network slicing attacks have become a focus of academic research. This section will review some existing studies, and security mechanisms used in this research.

2.1 Literature Review

To address DDoS attacks in 5G networks, experts and scholars have announced their observational conclusions and research results. Majeed *et al.* [7] used a deep learning model, CNN to identify abnormal network traffic, enabling early detection of attacks and reducing attack traffic to mitigate the attack force. Scholars have also extended defense mechanisms to O-RAN, deploying machine learning and artificial intelligence models on the RIC component of the SMO.

Houda *et al.* [8] proposed a Federated Deep Reinforcement Learning technology for O-RAN to mitigate the risk of jamming attack. This technology also introduces federated learning and machine learning into O-RAN to improve the accuracy of attack detection. It can also autonomously adjust detection strategies in response to different scenarios to increase its detection flexibility. The authors leveraged the local nature of federated learning to enable model training at multiple locations, and then transmitted parameters for aggregation. This increases O-RAN`s capability in modeling validation data, enabling direct verification using real data from each local node while ensuring the privacy of the original data.

Currently, there are still no technique standards and consensus on how to classify network slices based on existing network protocols.

Wu *et al.* [9] classified the traffic generated by applications into three levels by marking the applications and classifying the slices according to different traffic levels required. For example, network services with low traffic, such as HTTP and Messenger, are classified as lightweight slices; applications that experience traffic changes due to different user service usage, such as Facebook and Instagram, are classified as mixed slices; and applications that require high bandwidth, such as Cloudflare and Google Drive, belong to heavyweight slices.

Allaw *et al.* [10] simulated network slicing within the SDN architecture and used neural networks as a means of attack detection. The authors pre-divided traffic into three logical networks based on custom rules: eMBB, audio applications, and web text resources, such as HTTP. Traffic was then divided based on the defined slicing frequency bands using the OpenFlow protocol. Utilizing multiple attack classification models, they categorized the attacks into nine types and compared the performance of each model.

3. 3D5G-O System Architecture

The 3D5G-O classifies packets to network slices based on Differentiated Services Code Point (DSCP) values, enabling more efficient DDoS attack detection within the slice. Our model combines SMO and Central Unit (CU). Once an attack from the UE side is detected, it is blocked at the RAN, preventing malicious packets from passing through or reaching the core network.

If DDoS attack traffic flows through the UPF, it will consume the UPF's resources, thus affecting its service quality. Attack detection at the gNB is a distributed security system, allowing the UPF to maximize its performance and serve all users. However, packets flowing from the Data Network (DN) into the core network still require the UPF and its mirror port before being sent to the IDS/IPS. Of course, if these packets also originate from mobile phones and enter the DN via the base station's SMO, the DDoS attack will be detected at that base station.

Figure 1 shows the 3D5G-O system architecture. Upon system startup, the SMO pushes our AI/ML model M to the Near-RT RIC and then sends relevant parameters to the CU. When a packet p from UE is transmitted to the base station, it is passed from the Radio Unit’s (RU's) physical lower layer to the Distributed Unit (DU's) physical upper layer, where the analog signal is converted into a digital signal. The DU then passes p up to the CU. The CU then mirrors p into the AI/ML model M via the xAPP. The slice classification model developed in this research in the CU classifies the slice to which p belongs based on its DSCP value. The IDS/IPS then accumulates and counts traffic. This method calculates all QoS flows for all PDU sessions passing through a specific slice of the CU, for example, slice i, to detect attacks.

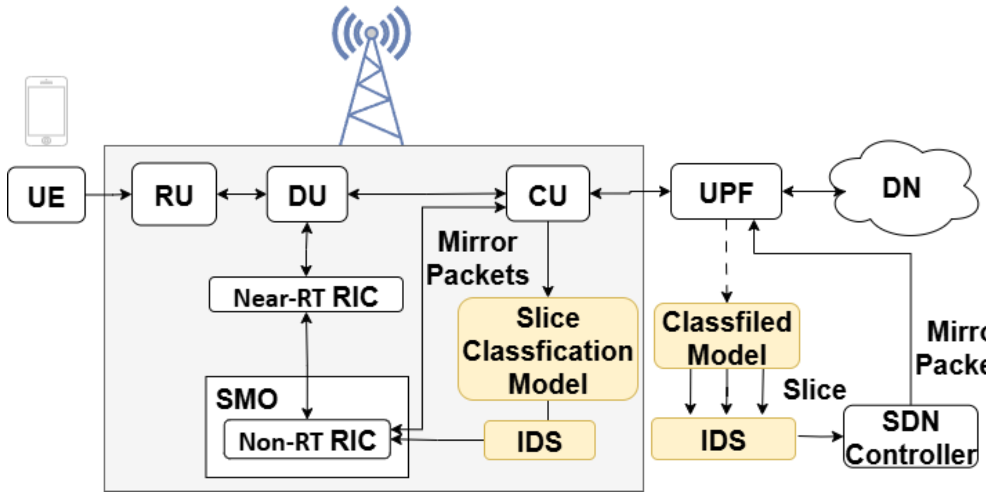

Figure 1: 3D5G-O system architecture.

3.1 Slice Classification

Currently, network standards lack clear definitions on the fact that which slice packet should be mapped to which specific network protocol (e.g., IP or transport layer). In this study, we use a custom-coded DSCP field for traffic classification. We also map existing application protocols to the 5G QCI/5QI, thereby redefining the DSCP values corresponding to those application protocols.

The 6 bits of the DSCP field are defined as follows: the first 3 bits are the NSSAI tag: bit 0 = 1 for eMBB, bit 1 = 1 for mMTC, and bit 2 = 1 for uRLLC (Note that if V2X and HMTC (High performance machine-type communication) need to be considered, the coding approach ought to be enhanced. Currently, we only deal with the first three slices). The last 3 bits represent the priority of the slice; a larger value indicates a higher priority. Table 1 shows the mapping between NSSAI_Type and DSCP definitions. However, due to the large number of protocols and applications in the world, it is hard to fully list all of them here and map them to any of the three types of slices. Of course, some applications require more than two types of slices, such as industrial automation, intelligent transportation systems, and augmented reality. Therefore, Table 1 also defines hybrid slices. For example: 101 represents eMBB + uRLLC, 011 shows mMTC + uRLLC, and so on.

Table 1:The mapping between NSSAI_Type and DSCP Definition.

NSSAI_Type	DSCP's first 3 bits	Description
eMBB	100	Representing eMBB
mMTC	010	Representing mMTC
uRLLC	001	Representing uRLLC
eMBB+ uRLLC	101	For example, transmitting images of patients lying on the bed for telesurgery
mMTC+ uRLLC	011	For example, an emergency shutdown failure in a smart factory triggers immediate control instructions.
. . .	. . .	. . .

Table 2, which shows the QCI/5QI values for each application category involved in this study, lists the DSCP values defined. The priority levels (the last three bits of DSCP) in the table are specified based on the application category's default priority, packet loss rate, and latency. mMTC communications do not have a specific 5QI type. Therefore, we expand the 5QI values to 11-13, corresponding to MQTT, CoAP, and LwM2M, respectively. A 5QI value of 9 categorizes traffic as Best Effort with its DSCP value set to 000000_2. Table 2 is merely an example; users may modify it following this principle and/or define specific DSCP values for more applications.

Table 2: Correspondence between 5QI value, application category and DSCP value.

NSSAI_Type	5QI	DSCP value	GBR Type	Application Category
uRLLC	1	001 010	GBR	Voice conversation
uRLLC	3	001 100	GBR	Real-time games, V2X
uRLLC	65	001 011	GBR	Critical message transmission for Multimedia
uRLLC	67	001 101	GBR	Mission-critical video
mMTC	11	010 011	Non-GBR	MQTT (Message Queuing Telemetry Transport Protocol)
mMTC	12	010 010	Non-GBR	CoAP (Conventional Application Protocol)
mMTC	13	010 000	Non-GBR	LwM2M (Lightweight device communication)
eMBB	2	100 101	GBR	Live video
eMBB	4	100 011	GBR	Non-real-time video streaming
eMBB	6	100 000	Non-GBR	Web browsing, background downloading
Others	9	000 000	Non-GBR	Applied to Best Effort
uRLLC +eMBB	82	101 001	critical GBR	Industrial Automation + Augmented Reality (AR)
uRLLC +eMBB	83	101 010	critical GBR	Discrete Automation (V2X)
uRLLC +eMBB	84	101 011	critical GBR	Intelligent transportation system
. . .	. . .	. . .	. . .	. . .

3.2 RF Model

RF suitable for handling large-scale network traffic will classify packets. In this study, the class_weight='balanced' parameter is set to assign higher weights to application categories with fewer samples during the training phase. RF is deployed in SMO with the following functions.

(1) Network Slice Label Mapping: With Table 2, we established a two-tiered mapping. mechanism.

1). The first tier, Protocol Name to Application Category, categorizes the packet's protocol name (Protocol Name) into a higher-level Application Category. For example, instant messaging applications such as SKYPE and TEAMSPEAK are categorized as Voice Conversation (see the first row of Table 2, 5QI = 1).

2). The second tier, Application Category to Slice and QoS, defines the corresponding NSSAI_Type and DSCP for each Application Category. For example, in the first row, the Voice Conversation category is then mapped to the slice with NSSAI_Type = uRLLC and

DSCP = 001010_2. All uncategorized applications are assigned to the default Best_Effort category with an NSSAI_Type of Others. The NSSAI_Type field is the prediction target (label) for our machine learning model, namely, the network slice.

(2) Data Cleaning and Stratified Sampling: Data cleaning and sampling aim to improve data quality and consistency and optimize the processing efficiency of model training without sacrificing sample representativeness.

1) Rare Class Handling:

From the distribution of the target variable NSSAI_Type, remove rare classes with fewer than r samples (in this study, r=10) to prevent interference with model training and evaluation.

2) Stratified Sampling:

Due to the large size of the original dataset (e.g., a total of Y records), training would be time-consuming and uneconomical. Therefore, we extract a subset from Y to form a representative subset of X records where $X < Y$. Sampling is performed based on the NSSAI_Type label. Let X_s be the number of samples for a given NSSAI_Type S (see Eq. (1)).

$$X_s = \frac{X}{Y} \times u \qquad (1)$$

where u is the number of samples of S in Y, aiming to eliminate the bias that may be introduced by random sampling and ensure that the final sample X can still truly maintain the actual proportion of all NSSAI_Types, like those in Y (see Eq.(2)).

$$X = \sum_{s=1}^{m} X_s, \quad 1 \leq S \leq m \qquad \textbf{(2)}$$

where m is the number of the concerned network slices.

Algorithm 1: RF Network Slicing and Application Category Training

```
Input: Network traffic dataset (CSV format), Application Category to NSSAI_Type/DSCP
       mapping rules
Output: Trained RF model M (ONNX format)

Phase 1: Network Slicing Label Mapping
1. Adding Application Category, NSSAI_Type/DSCP fields based on their mapping rules
2. Preprocessing data on those fields with infinite (inf) and missing values (NaN)
Phase 2: Feature Engineering
3. Merging similar applications to generate MergedProtocolName /* Google, Gmail. ->
   Google_services*/
4. Creating new proportional features $FBPR, FBBR, and\ FHPR$
Phase 3: Data Cleaning and Stratified Sampling
5. Removing those categories with fewer samples
```

```
    6. Selecting feature matrix X and target label Y /*using the merged MergedProtocolName*/
    7. Labeling target label Y
    8. Sampling the data (in this study, 300,000 records were selected from 3.5 million records)
    9. Create an asymmetric cost matrix to penalize NSSAI_Type misclassification
    Phase 4: Training RF model /*Performing hyperparameter search*/
10. Using RandomizedSearchCV with five-fold cross-validation on the training subset to search for
    optimal hyperparameters /*The classification model M is a RF with Bagging bootstrap sampling
    and Gini impurity as the splitting criterion*/
11. Using the optimal hyperparameters and cost matrix to train the final RF classification model M
    on the training set by utilizing five-fold cross-validation.
```

Algorithm 1: The summary of the RF Network Slicing and Application_Category Training algorithm.

(3) Training RF model

1) In the training subset X, M is trained using bagging theory. Sampling uses two key random processes, including Bootstrap sampling of training data and random sampling of features, while Gini Impurity [13] is employed as the evaluation scheme for node splitting.

2) Strategy for Handling Class Imbalance: The number of samples in each NSSAI_Type slice is uneven (e.g., |eMBB traffic| >> |uRLLC traffic|). Therefore, the M built will tend to favor the eMBB. To solve this problem, this study introduces an asymmetric cost matrix. Its penalty weight is inversely proportional to the number of samples in each class. This means that the cost of misclassifying a uRLLC (eMBB) packet is $1/|\text{uRLLC traffic}|$ ($1/|\text{eMBB traffic}|$) where $(1/|\text{uRLLC traffic}|) > (1/|\text{eMBB traffic}|)$.

The RF Network Slicing and Application_Category Training algorithm is summarized and listed in Algorithm 1.

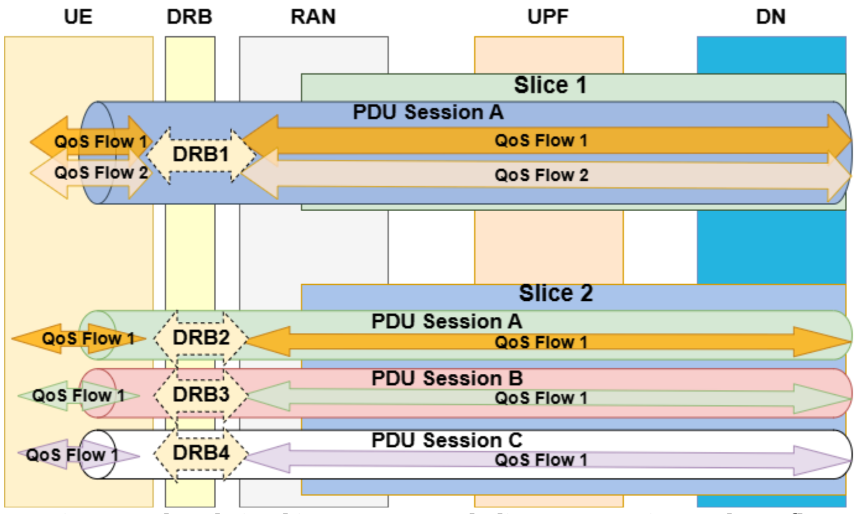

Figure 2: The relationship among network slices, PDU sessions and QoS flows.

3.3 Anomaly Detection and Identification

The 3D5G-O detects the QoS flows of all PDU sessions within a slice (see Figure 2). The packets p transmitted from the DN to the UPF are also duplicated through UPF's mirror port.

(1) Detection on the RAN side: The system uses a cross-layer tracking mechanism based on a quadruple Quad=(Src/Dst IP, PortNum, QFI, TEID) which integrates key identifiers from the application layer to the tunnel layer.

1) Src/Dst IP and PortNum: At the CU-UP layer, they are used to accurately identify the source and destination IP addresses of an application flow.

2) QFI (QoS Flow Identifier): Mapping directly to a packet p's QoS attributes, such as S-NSSAI and 5QI, to identify its Application_Category (the 5th column of Table 2) and priority (the last three bits of DSCP in the 3rd column).

3) TEID (Tunnel Endpoint Identifier): In the GTP-U protocol, it is used to associate. p with its corresponding radio bearer (DRB).

(2) When the IDS/IPS on the UPF side detects an attack, it directly modifies the routing (Flow) table to block the source connection of the attack packet. For suspicious connections that have not yet been confirmed to be attacks, the Meter table will be set to limit their transmission bandwidth or lower their transmission priority in Flow table to mitigate the seriousness of attack.

On the RAN side, when the CU detects abnormal traffic being sent to a Dst IP, it analyzes the Src IP of the QoS flows that transmit packets to the attacked Dst IP. The CU then sets packet filtering rules (such as access control lists (ACLs) based on TEID, QFI, IP, and/or port) to allow 80% of the Src IP's traffic passing through. Once it is determined to be an attack, the CU directly discards GTP-U packets that match the attack signature using the filtering methods mentioned above, and releases the UE's data bearer (DRB), interrupting its data transmission connection.

3.3.1. Detection techniques

Our detection techniques include the CuSum algorithm and information entropy. If hackers utilize many small traffic flows to attack a target or targets, but the detection based on total traffic volume does not reach the threshold to trigger an alarm, 3D5G-O can still detect the threat.

Under normal circumstances, the basic threshold Th1 of each NSSAI_Type slice is set to $\mu + 2\sigma$, and the alarm threshold Th2 is set to $\mu + 2\sigma$. If the upper limit of the TH2 is set too low, the detection system will be too sensitive. For example, when detecting eMBB slices, a massive number of packets often suddenly enter the slice, which can easily trigger an alarm. If Th2 is set too high, the attack will not be easily detected. On the other hand, CuSum's DDoS detection cannot effectively distinguish the QoS flows of an attack. In a DDoS attack, the frequency of sending packets to the target Dst IP in a certain time duration will be relatively higher than that of normal traffic. Therefore, a certain Dst IP can be used to detect the attack. For example, the entropy value of Dst IP$_i$ can determine whether traffic is abnormal or not. This is done by observing the traffic over a sliding time window W. When the entropy value decreases or decreases rapidly, the probability of the traffic being an attack source is high. Conversely, when the entropy value increases smoothly, the traffic is normal.

3.3.2. Detection procedure

Figure 3 shows our detection procedure, in which two sliding windows are utilized WCuSum and Wentropy.

(1) WCuSum: utilized to establish a dynamic baseline. Its size is fixed at T sec and it slides continuously over time.

(2) Wentropy: employed for attack signature analysis. Its size is fixed at n packets. Wentropy operates non-continuously. It is activated once after a CuSum alarm is triggered. We count Wentropy for detecting the following malicious behaviors by collecting the next n packets for analysis.

No matter whether operating in the RAN or the UPF's IDS/IPS, the CuSum algorithm continuously accumulates slice traffic. If the total traffic received by all Dst IPs in WCuSum for a particular slice, for example, slice i, exceeds Th2, or if the entropy value rises dramatically or decreases (we will describe the former later), we considered the traffic is abnormal. Even if the traffic in that slice is stable, a dramatic change in entropy can also be considered an anomaly. Furthermore, the entropy of traffic received by each Dst IP belonging to slice i within Wentropy is calculated. If an attack is confirmed (or suspected), the SMO and SDN Controller are notified to block the connection (or reduce its bandwidth).

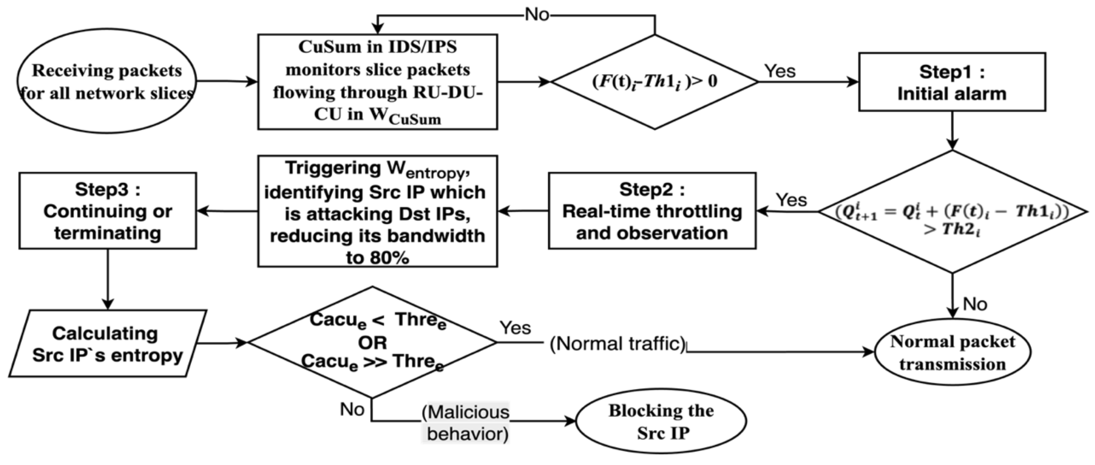
Receiving packets for all network slices
CuSum in IDS/IPS monitors slice packets flowing through RU-DU-CU in W_{CuSum}
No
$(F(t)_i - Th1_i) > 0$
Yes
Step1 : Initial alarm
$(Q^i_{t+1} = Q^i_t + (F(t)_i - Th1_i)) > Th2_i$
Yes
Step2 : Real-time throttling and observation
Triggering $W_{entropy}$, identifying Src IP which is attacking Dst IPs, reducing its bandwidth to 80%
Step3 : Continuing or terminating
No
Calculating Src IP`s entropy
$Cacu_e <$ Thre$_e$ OR $Cacu_e \gg$ Thre$_e$
Yes
(Normal traffic)
Normal packet transmission
No
(Malicious behavior)
Blocking the Src IP

Figure 3: Anomaly Detection procedure.

4. Experimental Results and Validation

Parameters of the following experiments are listed in Table 3. The following metrics are evaluated where TP, TN, FP and FN represent True Positives, True Negatives, False Positives and False Negatives, respectively.

Table 3: Parameters of our experiments.

Category	Parameter Name	Parameter Value	Description
Data Preprocessing	Original data set	Dataset-Unicauca-Version2-87Atts.csv	The raw network traffic dataset used in our experiments
	SAMPLE_ SIZE	300,000	The total number of samples after stratified sampling
Machine Learning Model	Model Type	Bagged Decision Trees (Random Forest)	A classifier trained using fitcensemble
	Cross-Validation Fold	5	Perform 5-fold cross-validation to evaluate models

Attack Scenario Parameter	attack_slice _name	eMBB	The target slice eMBB received DDoS attack
…	…	…	…

4.1 RF Model Construction

This study utilizes the IP Network Traffic Flows Labeled with 75 Apps data set [11], i.e., Dataset-Unicauca-Version2-75Atts.csv, which has a total of 3.5 million samples. The training time on this data set is long. So, we preprocessed the data on the Kaggle platform with the Stratified Sampling approach. That is, after the data application category conversion, we group the data into NSSAI_Types, and then randomly select samples from each group, e.g., NSSAI_Type j, according to Eq. (1). A total of 300,000 samples were selected as the model's training and validation data as

$$\sum_{s=1}^{m} X_s = 300{,}000$$

where m is the number of concerned network slices. The proportion of each NSSAI_Type in the selected 300,000-record subset is identical to the parent data [11], ensuring the representativeness of the experimental samples and the fairness of subsequent model evaluation.

```
Algorithm 2: Model Evaluation and Validation with Matlab
Input: Proportionally sampled data (in CSV format), model hyperparameter space,
       Application_Category to NSSAI_Type/DSCP mapping rules, model M after RF training
Output: Model evaluation metrics (confusion matrix, classification report)
    Step 1: Performing two-stage inference on the test set
 1. Using model M to predict the MergedProtocolName of samples/packets in the test set
 2. Mapping the predicted MergedProtocolName of samples/packets to the final NSSAI_Type
    Step 2: Computing variable evaluation and performance
 3. Generating a confusion matrix based on the NSSAI_TYPE predictions
 4. Performing the 5-fold cross-validation on the final model again
 5. Calculating Precision, Recall, and F1-Score for each slice
 6. Analyzing the individual accuracy for each fold and calculate average of the 5 results
```

Algorithm 2: RF model's evaluation algorithm.

The machine learning model M, implemented in Matlab, was trained using the algorithm shown in Algorithm 1. M consists of 50 decision trees. Algorithm 2 illustrates the algorithm of Model Evaluation and Validation algorithm in Matlab 2025a. Table 4 shows the confusion matrix of the RF classification model M. Table 5 presents the accuracy of M during 5-fold cross-validation. When Fold i is the evaluated subset S, the other four folds are S's training subsets. For example, the accuracy of Fold 1 is the result of testing on Fold 1 after training using Folds 2, 3, 4, and 5. The average of these five accuracy rates is 99.98%. Table 6 lists the model's evaluation metrics, including precision, recall, and F1-Score for each slice (eMBB, mMTC, uRLLC, and Others).

Table 4: Network slices' confusion matrix in training stage.

Predicted / Real	eMBB	mMTC	uRLLC	Others
eMBB	295103 (100.00%)	0	0	0
mMTC	1	160 (90.91%)	15	0

uRLLC	2	0	4482 (99.89%)	3
Others	0	0	26	208 (88.89%)

Table 5: 5-fold cross validation accuracy distribution, overall accuracy: 99.98% on Training stage.

	Fold1	Fold2	Fold3	Fold4	Fold5
Accuracy	99.995	99.99	99.992	99.93	99.995

Table 6: Model's evaluation metrics in Training stage.

Metrics / Slice	Precision	Recall	F1 Score
eMBB	1.0000	1.0000	1.0000
mMTC	1.0000	0.9091	0.9524
uRLLC	0.9909	0.9989	0.9950
Others	0.9858	0.8889	0.9339
Average	0.9941	0.9492	0.9703

4.2 Model Validation

Two experiments are performed to verify M's intrusion detection process. Experiment 1 monitors network-slice traffic during attacks, while Experiment 2 evaluates performance of the 3D5G-O and test schemes.

4.2.1. Experiment 1: Monitoring Network-Slice Traffic

Experiment 1 adopts a stratified sampling dataset for simulation. This dataset is used as packet replay to simulate traffic. Figure 4 shows the traffic generated by each concerned slice. Note that scales of the vertical axes are different for different subgraphs. Two times of DDoS attacks were launched against the eMBB slice at times T=0.121 sec and T=0.522 sec, each lasting 0.08 sec. The attack packets' transmission rate was approximately 9×10^9 bytes/sec, that is, a total of approximately 7.2×10^8 bytes was transmitted for each DDoS attack.

Figure 5 shows the flow accumulation, where the CuSum algorithm utilizes a sliding window of size=0.1 sec. The time interval from 0 to 0.1 sec is defined as the warm-up phase, during which no attack detection is performed. The flow data is only used to calculate the initial CuSum baseline parameters (μ and σ) and alarm threshold, representing Th1 and Th2 for the 0.1-0.2 sec window. That is, starting from 0.1 sec, the Th1 and Th2 for each 0.1-sec window, e.g., the ones between 0.2 and 0.3 sec, are derived from the data of the previous window, i.e., the window between 0.1 and 0.2 sec. As mentioned earlier, when the flow exceeds Th1, the excess value is accumulated by the CuSum algorithm. In Figure 5, Th2 for each window is shown as a red dashed line. During the warm-up phase from 0 to 0.1 sec, Th2 = 0.

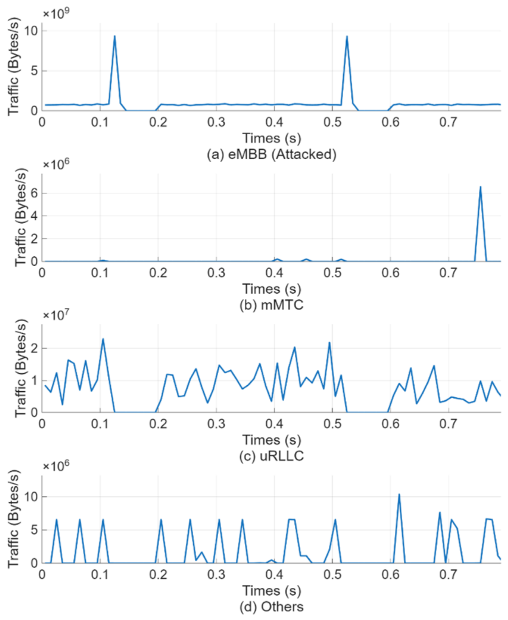

Figure 4: Traffics of detected network slices.

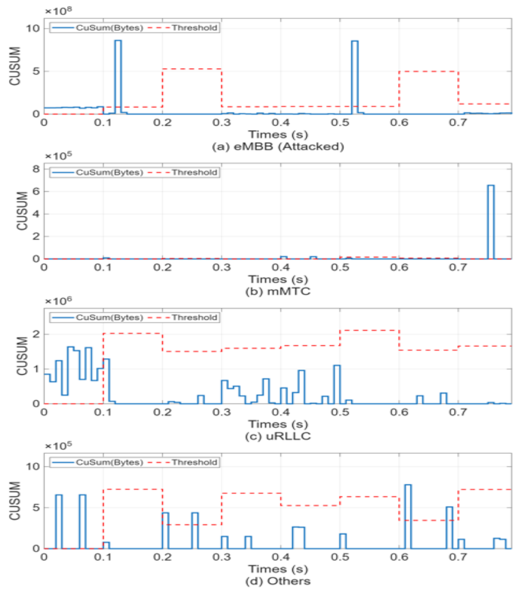

Figure 5: Cumulated Th2 traffics for detected slices by the CuSum (unit: bytes).

In Figure 5(a), the CuSum algorithm accumulates approximately 7.2× 10^8 suspected attack traffic both at T = 0.13 sec and T=0.532 sec, and the traffic begins to decrease almost simultaneously. At T = 0.141 sec (T = 0.541 sec), it drops to 0. This is because during this period, all packets classified to eMBB slices are converted into attack packets. The figure shows that between T = 0.141 and 0.2 sec (from 0.541 to 0.6 sec), the attack connection is blocked, and the traffic remains at 0. Only after the attack is completed and the connection resumes normal packet transmission, the traffic increases again. The system's response time from the start of the attack to the detection of the attack is 9 (=0.13 - 0.121, where 0.121 is the first attack's start time) msec and 10 (=0.532-0.522, where 0.522 is the second attack's start time) msec.

Of course, this detection time is related to the flow rate T, Th1, and Th2. For the same Th1 and Th2, a larger T results in a shorter detection time, and vice versa, smaller T will lengthen the detection time. For the same Th2 value and the same T, a larger Th1 value consumes a longer detection time because each time, the accumulated data decreases. For the same Th1 value and the same T, a larger Th2 value results in a longer detection time because more data exceeding Th1 needs to be accumulated.

The system's response time from the time pint when the attack is detected to the time point when the attacking source IPs are completely blocks was approximately 11 msec (=0.141–0.13) and 9 msec (=0.541–0.532) msec. Both are the time required to check traffic entropy. Furthermore, traffic on other non-attacked slices (as shown in Figures 4(b)–4(d)) did not exhibit significant anomalies, demonstrating the system's traffic isolation capabilities at the slice level. This means that an attack on one slice will not affect the normal operation and services of other network slices.

Figure 5(d) shows that Others exceeded Th2 at T=0.2 sec. However, entropy's detection determined that the traffic of Others was not an attack but normal behavior. The same situation can be observed at other time points, such as 0.25 sec, 0.62 sec, and 0.69 sec. The same situation was illustrated for mMTC in Figure 5(b) at 0.76 sec. Therefore, the traffic continued to be transmitted (see the traffic at the corresponding positions in Figures 4(a)-4(d)), which also shows that entropy's detection can effectively determine whether there is an attack.

4.2.2. Experiment 2: Performance Evaluation

In the following, we will evaluate the 3D5G-O and other ML approaches, including Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. Table 7- 9 present their confusion matrices. As shown in Table 7, the RF model demonstrates exceptionally high classification accuracy with only a negligible number of misclassified samples, about 0.0% (≈(13+1+4+26)/300,000). The results are almost the same to those listed in Table 4, indicating that RF works properly. In contrast, the LSTM model as illustrated in Table 8 performs relatively well on the eMBB slice. However, all uRLLC samples are misclassified as eMBB. As shown in Table 9, the CNN model's classification capability under the current experimental settings appears severely degraded, as it, similar with the LSTM, predicts most samples as eMBB.

Table 7: The Confusion Matrix of RF model.

Predicted / Real	eMBB	mMTC	uRLLC	Others
eMBB	295103	0	0	0
mMTC	0	163	13	0
uRLLC	0	1	4482	4
Others	0	0	26	208

Table 8: The Confusion Matrix of LSTM model.

Predicted / Real	eMBB	mMTC	uRLLC	Others
eMBB	295094	0	0	0
mMTC	40	136	0	0
uRLLC	4496	0	0	0
Others	234	0	0	0

Table 9: The Confusion Matrix of CNN model.

Predicted / Real	eMBB	mMTC	uRLLC	Others
eMBB	292035	468	2278	318
mMTC	172	0	4	0
uRLLC	4443	6	38	4
Others	232	0	1	1

Table 10: The execution times for 5-fold cross validation.

Model	Execution Time (sec)
RF	118.75

LSTM	785.73
CNN	166.16

As listed in Table 10, under the five-fold cross-validation setting, the RF model exhibited the fastest training speed, followed by CNN and then LSTM. This discrepancy primarily arises from the intrinsic differences in how the three models leverage computational resources, particularly parallel processing capabilities.

First, as an ensemble learning method, the Random Forest model constructs individual decision trees independently, making it highly amenable to parallelization and thereby achieving superior training efficiency on multi-core processors. Second, the convolution operations in CNNs possess inherent locality and redundancy, which, combined with the weighted-sharing mechanism, reduce parameter complexity and computational overhead per iteration, enabling efficient parallelization, especially on GPUs.

In contrast, the sequential nature of LSTM requires modeling long-term dependencies in time series data in a strictly ordered manner, where computations for each time step must await the completion of the preceding step. This inherent sequential dependency, coupled with the complexity of its gating mechanisms, imposes significant constraints on parallelization and leads to the longest training time.

5. Conclusions and Future Studies

This research developed a security scheme, called 3D5G-O, to detect network slicing DDoS attacks mainly at the 5G O-RAN. Experiments demonstrated that this system, through detection at the RAN, can mitigate intrusion traffic and block malicious packets for DDoS, significantly improving the timeliness and efficiency of defense.

Based on the experimental results and validation, we can conclude that, RF model is deployed in the SMO and combined with customized DSCP rules to map traffic from different applications to corresponding network slices. The model achieves classification accuracy of up to 99.98%. This system combines the CuSum algorithm for preliminary abnormal traffic detection and uses entropy analysis to identify the Dst IPs being attacked and the Src IPs conducting the attack. This hybrid approach has been proven to quickly and accurately identify DDoS attacks. Once an attack is detected, the system first restricts traffic. Once confirmed, it completely blocks the source, effectively mitigating the impact and protecting network service performance.

In the future, we would like to speed up the detection process by using GPUs and will also derive 3D5G-O`s reliability model and behavior model so that users can realize the reliability and behaviors before using it. These constitute our future studies.

References

[1] K. Samdanis, X. Costa-Perez and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32-39, July 2016, doi: 10.1109/MCOM.2016.7514161.

[2] V. P. Singh, M. P. Singh, S. Hegde, and M. Gupta, "Security in 5G Network Slices: Concerns and Opportunities," *IEEE Access*, vol. 12, pp. 52727–52743, 2024, doi: 10.1109/ACCESS.2024.3386632.

[3] F. A.Zadeh, Alan Civciss, V. Ravihansa, C. Sandeepa, and M. Liyanage, ”Descriptor: 5G Open Radio Access Network Multi-Modal Intrusion Detection Dataset (NetsLab-5GORAN-IDD),” *TechRxiv*, vol. 00, August 24, 2025, pp.1-10. DOI: 10.36227/techrxiv.175606222.20740292/v1.

[4] O-RAN Working Group 1, "O-RAN.WG1. OAD-R003-V08.00 Technical Specification," Nov. 17, 2022. [Online]. Available: https://drive.google.com/file/d/1-ojReC1iiJRE_jY-xyuN8pzJDLPWW0H6/view?usp=drive_link Accessed on August 20,2025

[5] F. Lau, S. H. Rubin, M. H. Smith and L. Trajkovic, "Distributed denial of service attacks," *Smc 2000 conference proceedings. IEEE international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions'* (cat. no.0, Nashville, TN, USA, 2000, pp. 2275-2280 vol.3, doi: 10.1109/ICSMC.2000.886455.

[6] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2nd Quart., 2023, doi: 10.1109/COMST.2023.3239220.

[7] A. Majeed *et al.*, "Deep Learning-Based Symptomizing Cyber Threats Using Adaptive 5G Shared Slice Security Approaches," *Future Internet*, vol. 15, no. 6, p. 193, 2023, doi: 10.3390/fi15060193.

[8] Z. A. E. Houda, H. Moudoud, and B. Brik, "Federated Deep Reinforcement Learning for Efficient Jamming Attack Mitigation in O-RAN," *IEEE Transaction on Vehicle Technology*, vol. 73, no. 7, pp. 9334–9343, Jul. 2024, doi: 10.1109/TVT.2024.3359998.

[9] Z.-X. Wu, Y.-Z. You, C.-C. Liu, and L.-D. Chou, "Machine Learning Based 5G Network Slicing Management and Classification," in *Proc. International. Conference on Artificial Intelligence and Information Communication. (ICAIIC)*, Osaka, Japan, 2024, pp. 371–375, doi: 10.1109/ICAIIC60209.2024.10463325.

[10] Z. Allaw, O. Zein, & A. Ahmad, "Cross-layer security for 5g/6g network slices: an sdn, nfv, and ai-based hybrid framework", *Sensors*, vol. 25, no. 11, p. 3335, 2025. https://doi.org/10.3390/s25113335

[11] J. S. Rojas, "IP Network Traffic Flows Labeled with 75 Apps," Kaggle Datasets, [Online]. Available: https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-75-apps. [Accessed: Aug. 24, 2025].

[12] S. Kumar, "Sharpening Your Model's Performance: A Deep Dive into Macro F1 and Weighted F1 Scores in Multi- Class Classification," *Medium,* Dec. 2024, https://medium.com/@shivakumarg814/sharpening-your-models-performance-a-deep-dive-into-macro-f1-and-weighted-f1-scores-in-b98339eb9dbb.

[13] T. K. Ho, "Random decision forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, Canada, 1995, vol. 1, pp. 278–282, doi: 10.1109/ICDAR.1995.598994.