AI-Driven Network Intrusion Detection and Resource Allocation in Real-World O-RAN 5G Networks

Theodoros Tsourdinis
Dept. of Electrical and Computer Engineering,
University of Thessaly
Sorbonne Universite, CNRS, LIP6
Volos, Greece
ttsourdinis@uth.gr

Nikos Makris
Dept. of Electrical and Computer Engineering,
University of Thessaly
Centre for Research and Technology Hellas, CERTH
Volos, Greece
nimakris@uth.gr

Thanasis Korakis
Dept. of Electrical and Computer Engineering,
University of Thessaly
Volos, Greece
korakis@uth.gr

Serge Fdida
Sorbonne Universite, CNRS, LIP6
Paris, France
serge.fdida@sorbonne-universite.fr

Abstract

5G technology, the latest advancement in mobile networks, promises increased data speeds, reduced latency, and enhanced capacity. However, network performance and user experience can be critically impacted by malicious traffic, identified as anomaly traffic and intrusion methods. Addressing these challenges requires optimized resource sharing and robust network security measures. In this paper, we present an AI/ML-driven Network Intrusion Detection framework with dynamic resource allocation and user management within the O-RAN architecture. Our Anomaly Traffic Detector (ATD) enhances network security by mitigating Denial of Service (DoS) attacks through an xApp that classifies network traffic in real-time and dynamically adjusts network resources and user connections. Experimental evaluations show that our system effectively maintains low latency under attack conditions, nearly doubles the throughput for legitimate users, and reduces average CPU usage by up to 15%. We use as reference platforms the OpenAirInterface, and FlexRIC for programming the slice and user connectivity decisions at the RAN level, and evaluate our scheme under real-world settings in a testbed environment.

This work is licensed under a Creative Commons Attribution International 4.0 License.

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0489-5/24/11
https://doi.org/10.1145/3636534.3697311

CCS Concepts

• **Networks → Network experimentation**; **Programmable networks**; **Mobile networks**; • **Security and privacy → Mobile and wireless security**.

Keywords

5G, O-RAN, Security, AI/ML, Slicing, OAI, OpenAirInterface, FlexRIC

ACM Reference Format:
Theodoros Tsourdinis, Nikos Makris, Thanasis Korakis, and Serge Fdida. 2024. AI-Driven Network Intrusion Detection and Resource Allocation in Real-World O-RAN 5G Networks. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3636534.3697311

1 Introduction

The advent of the 5th generation of mobile networks marks a transformative era in telecommunications unlocking countless opportunities for developing cutting-edge applications. However, a notable challenge persists in the absence of dynamic mechanisms for resource sharing among end users. This deficiency blocks the achievement of Key Performance Indicators (KPIs), often falling short due to under-provisioning. Furthermore, the deficiency in network optimization results in energy wastage, characterized by over-provisioning.

Additionally, network security is emerging as a critical concern, particularly with the increasing sophistication of network intrusion methods. Malicious attacks violate data integrity and privacy and significantly impact network performance. A network attack such as Denial of Service (DoS) can cause 5G core network functions such as the User Plane

Function (UPF) to fail and even cause the Radio Access Network (RAN) to malfunction [1]. The effects of these security threats extend beyond networking failures. They introduce inefficiencies in the use of resources, increase operating costs, and require recovery efforts. Furthermore, monolithic-closed telecommunications infrastructures often lack the adaptive mechanisms to dynamically manage these threats, leading to vulnerabilities.

An effective strategy for optimizing resources and improving network performance involves classifying users and the traffic they generate. By identifying and classifying traffic patterns, more efficient resource allocation is possible, ensuring that legitimate users receive the necessary bandwidth while mitigating the impact of malicious activities. The Open-RAN (O-RAN) architecture [2] offers a fertile ground for such strategies, as it exposes the RAN functionalities by controlling them through Radio Intelligent Controllers (RIC) via open interfaces. O-RAN's architecture enables a strategic logic guiding network optimizations through a three-step process: infer, decide, and determine. By integrating AI/ML, we can infer the network's current state and take the appropriate actions in real-time. This allows us to analyze past network behaviors, make informed decisions, and determine the most effective actions to optimize resource allocation and enhance security.

In this work, we adopt this three-step process by proposing an end-to-end 5G innovative framework that leverages AI/ML techniques to classify network traffic in real-time and dynamically adjust resource allocation and user management within the O-RAN architecture. Specifically, we developed an intrusion detection xApp that utilizes AI/ML models, trained on real-world datasets, to classify user traffic and make appropriate slicing and user management decisions on Radio Resource Control (RRC) level within the network. Our framework is a real-world solution, developed and tested on OAI [3] using standardized O-RAN interfaces and Service Models (SM). The ultimate goal of our solution is to suppress the network attacks and to maintain and even enhance the user experience during such incidents.

2 Related Work

The development of Network Intrusion Detection Systems (NIDS) has been studied extensively, with a significant focus on integrating AI/ML techniques to improve detection accuracy. A comprehensive survey in [4] underlines the importance of integrating machine learning algorithms into network anomaly detection systems, providing an in-depth review of Supervised Learning (SL) and Reinforcement Learning (RL) models. In [5], the authors proposed a Deep Learning-based (DL) self-adaptive architecture for anomaly detection, demonstrating the system's capability to handle fluctuating network traffic and achieve efficient anomaly detection performance. Similarly, [6] achieved high accuracy scores by converting network flows into images for analysis by a Convolutional Neural Network (CNN). Furthermore, Federated Learning (FL) architectures have been introduced to NIDS for cloud-native 5G Networks [7], showcasing the benefits of distributed learning in maintaining data privacy. Many works also focused explicitly on DoS attacks by proposing DL strategies and architectures for O-RAN 5G networks [8][9][10]. These studies highlight the importance of AI/ML in identifying and suppressing such attacks, although they often rely on simulated environments that may not reflect real-world complexities. Towards integrating these models into actual 5G and O-RAN networks, [11] designed an early detection system for DoS attacks using a custom RIC in srsRAN [12], yet it lacked mechanisms for subsequent network actions post-classification. A more holistic approach is presented in [13], where attacks are classified with high accuracy over the air using OSC-RIC [14] in an LTE testbed, and countermeasures are deployed to maintain low latency.

Regarding inference and resource allocation in O-RAN networks, [15] proposed an RL-based slicing framework to reduce Service Level Agreements (SLA) violations, evaluated within the OpenRAN Gym [16]. Similarly, a DL-based service-aware slicing scheme in [17] demonstrated high user experience and QoS within the OAI platform. The FlexSlice framework introduced in [18] involves redesigning the MAC scheduler for multi-level resource allocation, showing significant improvements in dynamic RAN slicing. Moreover, [19] presented an end-to-end O-RAN control loop for radio resource allocation in SDR-based 5G networks, focusing on real-time adaptability and resource efficiency through AI-driven xApps. In [20], authors leveraged RL for enabling 5G Dynamic Time Division Duplexing (TDD) within the O-RAN framework, achieving reduced latency.

Although these studies present advanced solutions for NIDS and RAN control/slicing, they do not integrate both anomaly detection and dynamic resource allocation in real-world environments. Existing works either focus on anomaly detection without subsequent network actions to mitigate detected anomalies or implement RAN control/slicing without considering real-time anomaly detection. Moreover, most NIDS solutions are based on simulations or assume the availability of full features during testing, which may not reflect the constraints of real-world systems.

In this work, we address this gap by designing, implementing, and evaluating a real-time network intrusion detection xApp within the O-RAN framework. Our solution combines real-time anomaly detection, dynamic resource allocation, and user management in a real-world setup. Specifically, our xApp employs AI/ML models trained on real-world datasets to classify network traffic and dynamically adjust resource allocation and user management. It identifies malicious users and triggers RRC connection release to mitigate their impact on the network, while prioritizing legitimate users through end-to-end slicing. By integrating our solution into the OAI platform and leveraging standardized O-RAN interfaces and Service Models (SM) from FlexRIC, we demonstrate a practical implementation that enhances network security and efficiency.

3 System Architecture

Our overall setup is illustrated in Fig. 1 and consists of an end-to-end O-RAN 5G Network based on OAI and FlexRIC. The target facility used for the development, application, and evaluation of the AI-driven NID O-RAN 5G network is the NITOS testbed, which is part of SLICES-RI [21]. The deployment specifications are summarized in Table 1. Below we outline the main components of the solution that enable the continuous classification of traffic and the subsequent slicing and user connectivity management that seamlessly enables high Quality of Experience (QoE) to the end-users. A video demonstration of the experiment setup is provided in the following link[1], while the experiment can be reproduced, by following the instructions and deploying the code available in Github [2].

Table 1: Experimental Setup

System	Description
CPU	Intel(R) Core(TM) i7-14700 @ 2.101 GHz
Cores	20
GPU	NVIDIA GeForce RTX 4070
RAM	32GB
Operating System	Ubuntu 22.04.2 LTS
5G-Core Network	OAI v2.0.1
5G-RAN/E2-AGENT	OAI v2.0.0
O-RAN RIC	FlexRIC dev
O-RAN SM	RC v01.03
5G-UE	OAI v2.0.0
Packet Manipulator	Scapy
Dataset	KDDCUP'99 [22]
ML Library	TensorFlow

3.1 General Architecture and Management of the network functions

Starting from the Core Network functions, we relied on OAI's implementation and deployed them as microservices utilizing Docker containers. These functions include the basic 5G core components such as AMF, AUSF, SMF, UDR, UDM, and multiple UPFs with the different Single Network Slice Selection Assistance Information (S-NSSAI) values configured. An S-NSSAI configuration contains a Slice Service type (SST) and Slice Differentiator (SD). This enables a full end-to-end slicing as a UE may access multiple slices over the same gNB. Each slice may serve a particular service type with an agreed SLA. Since the user traffic is passing through the GTP tunnels in the UPFs, the UPF is a critical point for classifying malicious user behaviors and identifying the user demands. 3GPP underlined the importance of the core traffic by standardizing the Network Data Analytics Function (NWDAF) function which mines the core data statistics and analyzes them. Considering that there is no integration of NWDAF on the O-RAN standardized architecture, we propose that NWDAF could be placed on the non-RT RIC and parse the user data through the O1 interface as the core network functions could be deployed on a Service Management and Orchestration framework. Subsequently, the NWDAF can enforce policies and send useful traffic summaries via the A1 interface to the RT RIC controller, which can then apply policies to the RAN through an xApp.

Since there is not yet an open implementation of the NWDAF functionality, we implemented our custom solution, the Anomaly Traffic Detector (ATD). This network function analyzes the traffic on the UPFs by leveraging a packet manipulator which in our case is Scapy [23]. In our scheme, the ATD plays the role of the NWDAF. We also employ the RT RIC from FlexRIC. We utilize FlexRIC since it has the least overhead compared to most RIC implementations [24] and because it is O-RAN-compliant providing an E2 agent, nearRT-RIC, and an xApp SDK framework. Therefore, our E2-Agent is OAI's gNodeB, and the xApp is an application we developed utilizing FlexRIC's SDK to infer the RAN Functionalities of E2-Agent utilizing mainly the RC SM.

3.2 Dataset and Machine Learning

The ATD network function, beyond analyzing the user data on the UPF side, it incorporates an intelligent mechanism to classify malicious and normal traffic, by utilizing an ML model that is part of its architecture.

This ML model was trained on a real-world dataset KDD-CUP'99 which is the most widely used data set for the evaluation of network intrusion systems [22]. The dataset contains a substantial number of instances, with over 4 million for training and around 311,029 for testing. The dataset includes

---

[1] Video demonstration available: https://youtu.be/4hx1mAvhXMY

[2] Link to reproducing the experimental setup: https://github.com/teo-tsou/oai-anomaly-detection

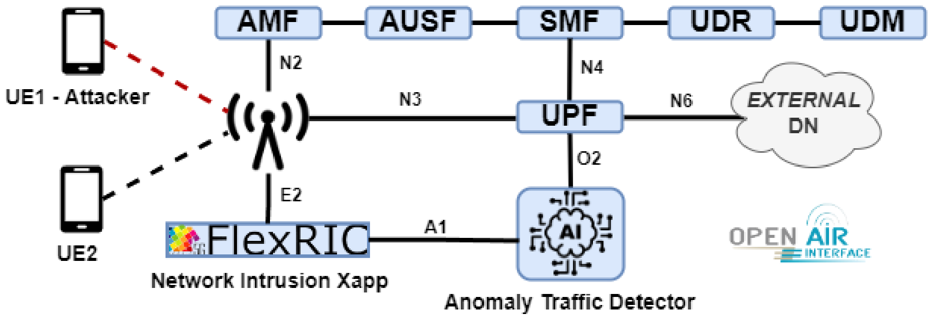


Figure 1: Experimental Setup: End-to-End Deployment of the AI-Driven Network Intrusion Detection 5G Network.

a huge variety of features related to the basic network connection characteristics such as packet header information and advanced features such as content-related information. We selected five important features: the protocol type which defines the protocol used in the connection (TCP, SCTP, UDP), the service that is running in the destination network such as HTTP, FTP, or SSH, the flag which establishes the status of the connection such as normal (SF), rejected (REJ), or reset (RST), and finally the source and destination bytes. These features were chosen for their relevance in distinguishing between normal and malicious traffic and their compatibility with real-time analysis through Scapy, which extracts information only at the packet level. Furthermore, the dataset contains 4 attack labels: Probing Attack, Remote to Local Attack, Denial of Service Attack, and User to Root Attack. Our preprocessing steps are described below:

- Label Conversion: Converting multi-class labels into binary labels: 1 for any attack, 0 for normal.
- Flag Conversion: Converting the dataset's flag values to a format compatible with Scapy.
- Feature Selection: Selecting the Scapy-related/relevant features.
- Encoding and Scaling: To standardize the data we use OneHotEncoder for categorical features and MinMaxScaler for numerical features.

After preprocessing, we trained and excessively evaluated several AI/ML learning models using the TensorFlow implementation, including Random Forest, One-Class SVM, Local Outlier Factor, KNN, and Autoencoders. The comparison of the models led us to employ the Random Forest model due to its better performance compared to the other models in terms of accuracy and training/inference times.

3.3 Anomaly Detection and Countermeasures

The detailed operation of our framework is illustrated in Fig. 2. The ATD unit utilizing Scapy, continuously monitors UPF traffic and classifies clients based on their IP and S-NSSAI values. It manipulates each packet in real-time, extracting the necessary features that our ML model was trained on. After collecting the first N packets, the ATD preprocesses these features and feeds them into the Random Forest classifier. Then the Random Forest by applying a sliding window mechanism processes N=30 packets at a time, classifying the traffic as benign or malicious. The reason we selected 30 packets-window is to reduce infer/prediction times as close to real-time and avoid false outliers in the classification with a larger input range. Finally, the ATD sends the anomaly percentage per UE to the xApp for the RAN Control and countermeasures.

The functionality of the xApp is summarized in Algorithm 1. First, it connects to FlexRIC's RT RIC and subscribes to the RC SM offered by the E2 Agent. Then accepts incoming socket connections from ATD clients and initializes the necessary data structures for UE identification. In the main loop, it listens for ATD messages, and for each message, it extracts the UE ID, the S-NSSAI values, and the anomaly ratio per UE and updates the UE-related data structures. Then it determines the physical RB allocation based on the anomaly ratios it receives per UE, with the formula given by Eqn. 1. To avoid the total RB allocation exceeding 100%, it scales down proportionally the allocation through Eqn. 2 and 3. Finally, when a UE's anomaly ratio reaches 100%, the xApp classifies it as an attacker and triggers an RRC UE Connection release, causing the UE to disconnect from the network. Subsequently, the xApp reassigns the PRB allocation to the remaining UEs, ensuring efficient resource distribution.

$$\text{PRB}_i = (1 - \text{AnomalyRatio}_i) \cdot 100 \tag{1}$$

$$\text{ScalingFactor} = \frac{100}{\text{TotalPRB}} \tag{2}$$

$$\text{AdjustedPRB}_i = \text{PRB}_i \cdot \text{ScalingFactor} \tag{3}$$

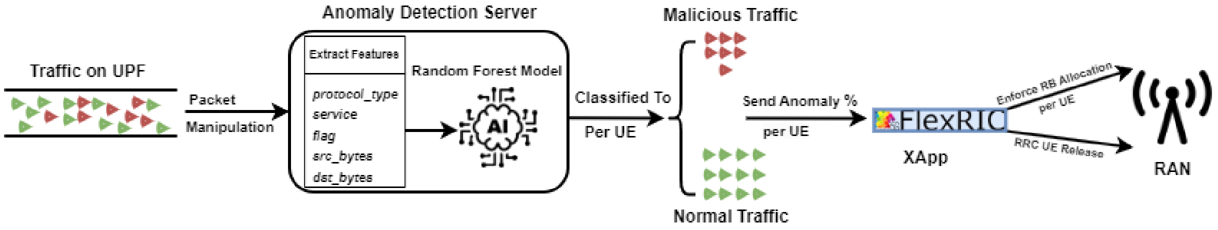


Figure 2: Detailed Architecture of the AI-Driven Network Intrusion Detection System.

Algorithm 1 xApp Functionality

```
Initialization:
Connect to RT-RIC and subscribe to the RC SM.
Set up socket and accept connections from ATD clients.
Initialize ue_data structure with default values.
Main Loop:
while True do
    Receive and Process Messages:
    Listen for incoming messages from ATD clients.
    Parse messages to extract UE ID, S-NSSAI, and anomaly ratios.
    Update ue_data structure.
    Traffic Analysis and Classification:
    Determine PRB allocations based on anomaly ratios.
    Resource Allocation and Enforcement:
    if a UE is an attacker (100% anomaly ratio) then
        Set PRB allocation to 0% for the attacker.
        Distribute remaining PRB among other UEs.
        Trigger RRC release for the attacker UE.
    else
        Adjust PRB allocations to ensure total does not exceed 100%.
        Apply slicing changes to enforce PRB allocations.
    end if
end while
```

4 Experimental Evaluation

To evaluate our solution, we first compared the performance of the ML models. Then, we tested the efficiency of our solution in both the preservation and enhancement of network performance and user experience during different anomaly scenarios, including DoS attacks.

We measured the ML model's performance using three metrics: accuracy, ROC AUC, and F1 scores. Accuracy measures the classified instances among all cases, calculated using Eqn. 4, where *TP* stands for True Positives, *TN* for True Negatives, *FP* for False Positives, and *FN* for False Negatives. ROC AUC distinguishes between classes by plotting the true positive rate (TPR) against the false positive rate (FPR). It is calculated by Eqn. 5. The F1 Score is the harmonic mean of precision and recall. It is calculated by Eqn. 6, where *Precision* is $\frac{TP}{TP+FP}$ and *Recall* is $\frac{TP}{TP+FN}$. We can observe from Fig. 3a that the Random Forest model had the best performance with high accuracy, ROC AUC, and F1 scores, making it the most reliable for our NIDS. The autoencoder also had similar performance but required more computational resources. Additionally, we measured the training and inference times for the different models, as shown in Table 2. The inference times correspond to the 30-packet window predictions. The Local Outlier Factor model achieved the shortest times for both training (0.77 sec) and inference (0.6 ms). However, due to its lack of accuracy, we explored the Random Forest model, which had a low training time (4.36 sec) and inference time (3.4 ms), making it suitable for our real-time system. On the other hand, the Autoencoder and KNN models had significantly longer training and inference times, which may limit their practical applicability in dynamic network environments such as ours. With these observations, we focused on the Random Forest Model due to its balanced performance and efficiency. Fig. 3b presents the confusion matrix for the Random Forest model, where 89.46% of benign traffic and 80.37% of attack traffic were correctly classified during testbed integration. These percentages, although lower compared to some literature, demonstrate the efficacy of our classifier under the constraints of real-world system integration. The limited feature set used, which is compatible with real-time analysis through Scapy, affects the overall accuracy. Despite these limitations, our framework achieves relatively high accuracy.

To evaluate our solution under realistic conditions, we designed several traffic scenarios in our testbed connecting two UEs. In the initial scenario illustrated in Fig. 4a, in the beginning, both UEs share roughly the same throughput since they slice equal amounts of RBs and both generate normal traffic. At the marked point with a red dotted line, UE1 begins to send some malicious packets into the network, generated via Scapy using the test (unseen) part of the KDDCUP'99 dataset. Due to the absence of any NID mechanisms in this scenario, both UEs continue to share the same network resources even after the introduction of malicious traffic, leading to a noticeable degradation in performance for the normal user, UE2. In the subsequent scenario in Fig. 4b, we introduce our NID solution alongside the xApp. As soon as the malicious packets are inserted into the network, our system successfully classifies them as abnormal and relocates the RBs based on the anomaly percentage per UEs for every sliding window of 30 packets. Consequently, the QoE for the legitimate UE (UE2) improves significantly as it gets the most RBs, while the QoE for the suspicious UE (UE1) declines.

The third scenario demonstrated in Fig. 5a explores the network's vulnerability to a DoS attack, executed from UE1 using Hping3, without any defensive actions. During this attack, both UEs experience a dramatic drop in throughput to nearly zero. This illustrates the impact of the DoS attack across the 5G network without any defensive mechanism. In the final scenario (see Fig. 5b), we evaluate the resilience of our system to the same DoS attack, but this time with our defensive solution activated. Our system identified the DoS attack since the anomaly percentage of the 30-packet sliding window reached a 100% threshold. Then the xApp triggers an RRC connection release specifically for UE1. This isolates the attacking UE and prevents it from further degrading network performance. UE2 experiences minimal disruption, although its throughput almost doubled and remained largely stable, illustrating the system's effectiveness in detecting and actively preventing sustained network attacks. This ensures that normal network users maintain QoE even under attack conditions.

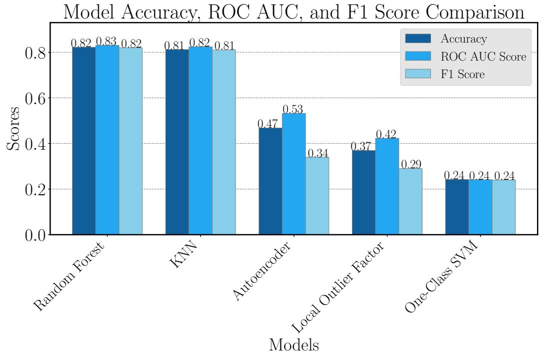
Furthermore, we measured the impact of the DoS attack on normal user latency. Specifically, we measured the Round Trip Time (RTT) for UE2 under attack conditions, both with and without our defense mechanisms, as illustrated in Fig. 6. The results indicate a significant rise in RTT during the attack, with latency peaking at nearly 3 sec without our NIDS. When our xApp is running, the RTT remains substantially lower, maintaining an average latency of approximately 18 ms. That indicates that our framework successfully suppressed the attack and kept the user's latency at a low level.

Finally, our solution demonstrates significant energy efficiency improvements. As depicted in Fig. 7, we monitored the CPU usage within the UPF docker container during a DoS attack, both with and without our NIDS solution. With our defense mechanism, we observed an average reduction of up to 15% in CPU usage, reversing the failure of UPF and enhancing energy savings.

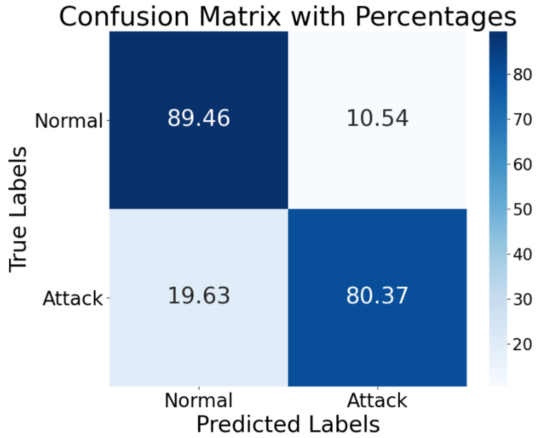
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$\text{ROC AUC} = \int_0^1 \text{TPR}(\text{FPR}) \cdot d(\text{FPR}) \tag{5}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$



(a) Model Training Evaluation: Accuracy, ROC AUC, and F1 Score Comparison.

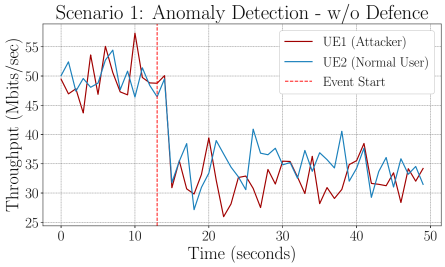


(b) Confusion Matrix for the Random Forest Model.

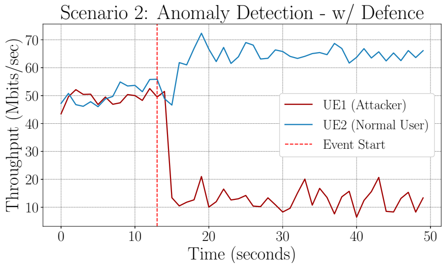
Figure 3: Confusion matrix for the Random Forest Model.

Model	Train Time (s)	Infer Time (ms)
Random Forest	4.36	3.4
One-Class SVM	125.98	6.14
Local Outlier Factor	0.77	0.6
KNN	5.51	9.14
Autoencoder	181.46	23.11

Table 2: Training and inference times for various machine learning models

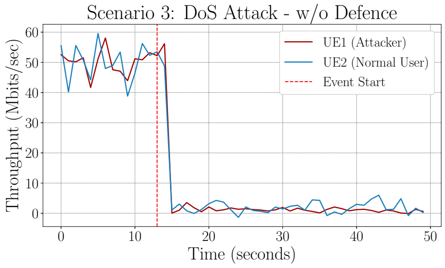


(a) Anomaly Traffic w/o Defence

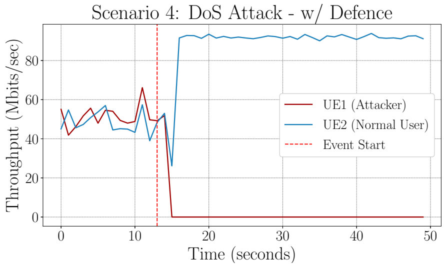


(b) Anomaly Traffic w/ Defence

Figure 4: Anomaly Traffic w/ and w/o Defence; red line denotes when the anomaly traffic generated.



(a) DoS Attack w/o Defence



(b) DoS Attack w/ Defence

Figure 5: DoS Attack w/ and w/o Defence; red line denotes when the attack started.

5 Conclusion

In this work, we have demonstrated an AI/ML-driven framework for Network Intrusion Detection and dynamic resource/user management within the O-RAN architecture, focusing on

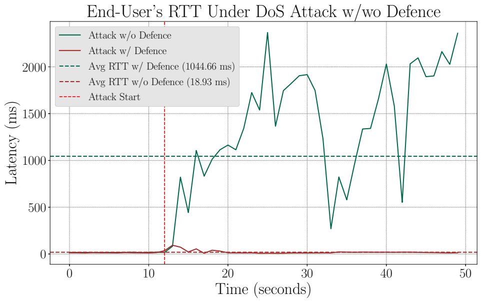


Figure 6: End-User’s RTT Under DoS Attack w/wo Defence

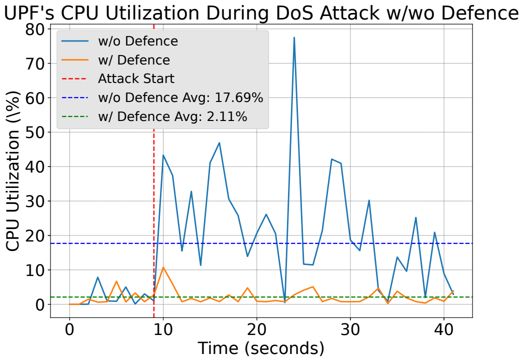


Figure 7: UPF’s CPU Utilization During DoS Attack w/wo Defence

enhancing network security and optimizing 5G network performance. Our solution effectively handles anomaly traffic and mitigates intrusion methods such as Denial of Service (DoS) attacks through real-time traffic classification and dynamic slicing and RRC UE connection management, by extending the RC SM in FlexRIC. Through extensive evaluation of various AI/ML models, the Random Forest model was selected for its balanced performance and efficiency. Despite the limitations of our system, which include relying on available features during packet manipulation that can affect the model’s accuracy, our solution still achieved good accuracy rates. Our experimental results highlight the benefits of our framework: maintaining low latency under attack conditions, nearly doubling throughput for legitimate users, and reducing CPU usage by up to 15%. In the future, we foresee extending our solution by dynamically managing/relocating edge computing resources (e.g. Multiple Access Edge Computing services) based on the anomaly detection mechanisms of our system.

6 Acknowledgments

The research leading to these results has received funding from the European Union's Horizon Europe Research and Innovation Programme under the Grant Agreement No. 101079774 (SLICES-PP), and from the European Union's Horizon Europe research and innovation programme (SNS-JU) under the Grant Agreement No 101139285 (NATWORK). The European Union and its agencies are not liable or otherwise responsible for the contents of this document; its content reflects the view of its authors only.

References

[1] Raja Ettiane, Abdelaali Chaoub, and Rachid Elkouch. Toward securing the control plane of 5G mobile networks against DoS threats: Attack scenarios and promising solutions. *Journal of Information Security and Applications*, 61:102943, 2021.

[2] Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges, 2022.

[3] Florian Kaltenberger, Guy de Souza, Raymond Knopp, and Hongzhi Wang. The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap. In *WSA 2019; 23rd International ITG Workshop on Smart Antennas*, pages 1–5, 2019.

[4] Song Wang, Juan Fernando Balarezo Serrano, Kandeepan Sithamparanathan, Akram Al-Hourani, Karina Gomez Chavez, and Ben Rubinstein. Machine Learning in Network Anomaly Detection: A Survey. *IEEE Access*, PP:1–1, 11 2021.

[5] Lorenzo Fernández Maimó, Ángel Luis Perales Gómez, Félix J. García Clemente, Manuel Gil Pérez, and Gregorio Martínez Pérez. A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks. *IEEE Access*, 6:7700–7712, 2018.

[6] Jordan Lam and Robert Abbas. Machine Learning based Anomaly Detection for 5G Networks, 2020.

[7] Salah Bin Ruba, Nour El-Houda Yellas, and Stefano Secci. Anomaly Detection for 5G Softwarized Infrastructures with Federated Learning. In *2022 1st International Conference on 6G Networking (6GNet)*, pages 1–4, 2022.

[8] Jung-Erh Chang, Yi-Chen Chiu, Yi-Wei Ma, Zhi-Xiang Li, and Cheng-Long Shao. Packet Continuity DDoS Attack Detection for Open Fronthaul in ORAN System. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pages 1–5, 2024.

[9] C.T. Shen, Y.Y. Xiao, Y.W. Ma, J.L. Chen, Cheng-Mou Chiang, S.J. Chen, and Y. C. Pan. Security Threat Analysis and Treatment Strategy for ORAN. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 417–422, 2022.

[10] Giorgi Iashvili, Maksim Iavich, Razvan Bocu, Roman Odarchenko, and Sergiy Gnatyuk. Intrusion Detection System for 5G with a Focus on DOS/DDOS Attacks. In *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 2, pages 861–864, 2021.

[11] Bruno Missi Xavier, Merim Dzaferagic, Diarmuid Collins, Giovanni Comarela, Magnos Martinello, and Marco Ruffini. Machine Learning-based Early Attack Detection Using Open RAN Intelligent Controller, 2023.

[12] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano, and Doug J. Leith. srsLTE: an open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, WiNTECH '16, page 25–32, New York, NY, USA, 2016. Association for Computing Machinery.

[13] Azuka Chiejina, Brian Kim, Kaushik Chowhdury, and Vijay K. Shah. System-level Analysis of Adversarial Attacks and Defenses on Intelligence in O-RAN based Cellular Networks, 2024.

[14] F. Bimo, F. Feliana, S. Liao, C. Lin, D. F. Kinsey, J. Li, R. Jana, R. Wright, and R. Cheng. OSC Community Lab: The Integration Test Bed for O-RAN Software Community. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 513–518, Los Alamitos, CA, USA, oct 2022. IEEE Computer Society.

[15] Raoul Raftopoulos, Salvatore d'oro, Tommaso Melodia, and Giovanni Schembra. DRL-based Latency-Aware Network Slicing in O-RAN with Time-Varying SLAs. 02 2024.

[16] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. OpenRAN Gym: An Open Toolbox for Data Collection and Experimentation with AI in O-RAN, 2022.

[17] Theodoros Tsourdinis, Ilias Chatzistefanidis, Nikos Makris, Thanasis Korakis, Navid Nikaein, and Serge Fdida. Service-aware real-time slicing for virtualized beyond 5G networks. *Computer Networks*, 247:110445, 2024.

[18] Chieh-Chun Chen, Chia-Yu Chang, and Navid Nikaein. FlexSlice: Flexible and real-time programmable RAN slicing framework. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pages 3807–3812, 2023.

[19] Asheesh Tripathi, Jaswanth S R Mallu, Md. Habibur Rahman, Abida Sultana, Aditya Sathish, Alexandre Huff, Mayukh Roy Chowdhury, and Aloizio Pereira Da Silva. End-to-End O-RAN Control-Loop For Radio Resource Allocation in SDR-Based 5G Network. In *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*, pages 253–254, 2023.

[20] Karim Boutiba, Miloud Bagaa, and Adlen Ksentini. On enabling 5G Dynamic TDD by leveraging Deep Reinforcement Learning and O-RAN. In *2023 IEEE/IFIP NOMS*, 2023.

[21] Serge Fdida, Nikos Makris, Thanasis Korakis, Raffaele Bruno, Andrea Passarella, Panayiotis Andreou, Bartosz Belter, CÃ©dric Crettaz, Walid Dabbous, Yuri Demchenko, and Raymond Knopp. SLICES, a scientific instrument for the networking community. *Computer Communications*, 193:189–203, 2022.

[22] Ibrahim Obeidat, Nabhan Hamadneh, Mouhammd Al-kasassbeh, and Mohammad Almseidin. Intensive Preprocessing of KDD Cup 99 for Network Intrusion Classification Using Machine Learning Techniques, 2018.

[23] Rohith Raj S, Rohith R, Minal Moharir, and Shobha G. SCAPY- A powerful interactive packet manipulation program. In *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, pages 1–5, 2018.

[24] Robert Schmidt, Mikel Irazabal, and Navid Nikaein. FlexRIC: an SDK for next-generation SD-RANs. In *Proceedings of the 2021 17th CoNEXT*. ACM.