Federated Learning for Anomaly Detection in Open RAN: Security Architecture Within a Digital Twin

Yasintha Rumesh*, Dinaj Attanayaka*, Pawani Porambage*, Jarno Pinola*, Joshua Groen†, Kaushik Chowdhury†
*VTT Technical Research Centre, Finland. Email: *[firstname.lastname]@vtt.fi
†Northeastern University, Boston, USA. Email: †[groen.j, k.chowdhury]@northeastern.edu

***Abstract*—The Open Radio Access Network (Open RAN) specifies the evolution of RAN with a disaggregated, open and intelligent architecture to meet the requirements of next-generation networks. While this provides flexibility and optimization for RAN, it raises new security concerns, potentially increasing vulnerability to cyber threats through disaggregated elements. We introduce a security architecture that functions as a platform to evaluate configurations and train security algorithms within a Network Digital Twin (NDT), which is compliant with the O-RAN architecture defined by the O-RAN Alliance. The elements of the security architecture reside within the NDT and facilitate the training of machine learning (ML) models, which play a pivotal role in the O-RAN security operations. To exemplify this framework, we demonstrate a hierarchical Federated Learning (FL) based anomaly detection algorithm that can be applied for three traffic slices in O-RAN. We use Colosseum, an O-RAN-compliant emulation system, to generate time-series data for training. Our trained model is able to detect anomalous traffic and identify traffic slice types with over 99% accuracy.**

***Index Terms*—Open Radio Access Network, Network digital twin, Anomaly detection, Federated learning**

I. Introduction

In the Open Radio Access Network (Open RAN), there is a developing trend of disaggregation, which entails the gradual separation of distinct functional elements and components that have traditionally comprised an integrated network infrastructure [1]. This separation extends to hardware, software, and network services, allowing them to exist as distinct entities that can be administered and operated independently. While this approach offers advantages in terms of flexibility and optimization, it also introduces new cyber threats against Open RAN (O-RAN) and its operations as each disaggregated element potentially becomes an entry point for cyberattacks.

The process of RAN softwarization and the increased openness of its architecture, including communication interfaces, align the implementation of O-RAN with the trends in efficient code delivery through Continuous Integration and Continuous Development (CI/CD) practices. When Machine Learning (ML) models are deployed in the O-RAN environment (i.e., defined by O-RAN Alliance), there are many potential issues that arise in terms of both the development and deployment stages of the model life cycle. In the development stage, issues may appear for data acquisition, data reliability, model structures, model updates, and hyperparameters. In the deployment stages, potential problems arise with respect to model decay, fairness, generalization ability, robustness, or numerical stability. Therefore, it is vital to choose the appropriate Artificial Intelligence (AI) technique, using synthetic data and managing the ML lifecycle properly in the RAN Intelligent Controllers (RICs) platform to overcome these challenges [2].

Federated Learning (FL), including its various forms such as hierarchical or distributed FL, is well-suited for scenarios involving the disaggregated nature and accordance with the general principles such as securing data privacy of the O-RAN framework [3] . It employs locally generated data to train models and combines model updates to train a global model. FL models can be effectively trained to detect anomalies or intrusions in O-RAN [4]. These models can be deployed within RICs to analyze real-time network traffic, identify anomalies, pinpoint their origins, and potentially detect security attacks before they can affect the RAN infrastructure. Another advantage of FL is that it can reduce the communication overhead of transferring data from different interfaces to a centralized location. Therefore, in this paper, we present a hierarchical FL-based anomaly detection mechanism and demonstrate its applicability using three traffic classes generated from an O-RAN emulator. We also present a Network Digital Twin (NDT) with a security architecture for O-RAN, as shown in Figure 1, where its components contribute to continuously updating and training FL models responsible for O-RAN security operations.

The remainder of the paper is organized as follows. Section II describes the background on O-RAN and NDT security. Section III presents the proposed NDT security framework. Section IV and V respectively elaborates on the O-RAN experimental setup and the numerical results obtained. Finally, Section VI provides the conclusions and future works.

II. Background and Related work

According to the reference architecture proposed by the O-RAN alliance [5], the O-RAN components are disaggregated into Central Unit (O-CU), Distributed Unit (O-DU) and Radio Unit (O-RU) accompanied by the RICs, i.e., near-Real Time (RT) and non-RT, Service Management and Orchestration (SMO) framework, and open interfaces. O-RAN promises the possibility of higher security measures with interoperability of different hardware and software components, protected RAN disaggregation with interface security, higher availability due to disaggregated architecture and software security different stages of O-RAN lifecycle [1].

NDTs can be used for training and testing AI/ML algorithms, taking data from the physical and synthetic data from NDTs. In the context of RAN, NDTs have to take into account

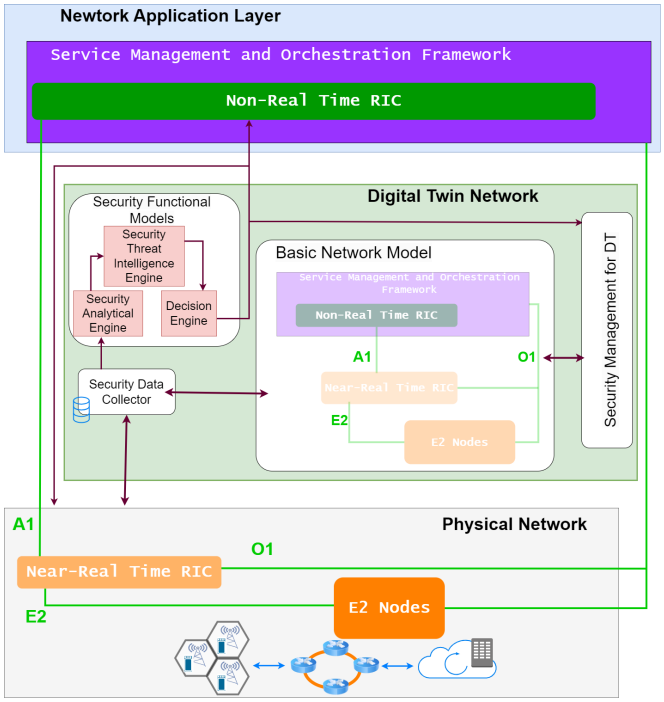

Fig. 1: Proposed security architecture for O-RAN with NDT.

the highly dynamic nature of the wireless communication medium. A RAN NDT finds a compromise between complexity, accuracy, and flexibility to keep up with the continuous changes in the physical network. The authors in [6] introduce a NDT design incorporating a data repository responsible for collecting and preserving data from the physical RAN. This encompasses service mapping models that digitally depict the various components and functions of the physical RAN and manage the NDT components and their interactions. Following this approach, different network monitoring and optimization applications can then utilize a RAN NDT instance customized to their specific needs.

The authors in [7] list some requirements for efficient deployment of an operational RAN NDT that provides the means to monitor and predict (e.g., through simulations, emulations, or AI models) operational states of the physical network, and directly control its configuration based on the analysis performed in the NDT. They highlight the need of ***distributed approaches*** for balancing the AI/ML model accuracy, training time, and related communication overhead. As explained in [8], an O-RAN NDT could enable the training of rApps and xApps with up-to-date information both on network level data and with specific measurements on channel conditions and mobility patterns of individual User Equipments (UEs). An O-RAN NDT with UE DTs will enable AI-driven network security, threat, and fault detection by using real-time and historical data to train anomaly detection algorithms.

The application of FL in O-RAN has been advocated for optimizing resource allocation [9] and enhancing access control mechanisms [10]. Building on our prior research, which employed peer-to-peer FL for anomaly detection within O-RAN, we recognize industry recommend offline training of ML models as a best practice [3]. However, there is a notable gap in the literature regarding the challenges associated with the training ML models offline. Addressing this, we introduce a novel security framework that incorporates NDT to mitigate these training challenges.

III. Security Architecture with NDT

As shown in Figure 1, the logical components of security architecture are located inside the NDT. In the process of security threat identification, NDTs can be used to predict potential network disruption and network flow anomalies [8]. For instance, each base station within its coverage will continuously send performance management data to the NDT. These may include aggregated metrics such as service availability, service quality, packet drop ratio, service accessibility level, etc. This data can be used for ML models to train for anomaly detection. Although the current O-RAN ML workflow specification does not support the real-time control loop and model training at E2 nodes yet [3], we utilize those in the NDT for the model training to showcase further possible enhancements.

The upper layer is the network application layer. This performs the optimization of security policies and network-level security configuration based on user requirements, resource availability or other appropriate parameters. When the verification process is complete, the NDT layer sends the control updates for the security configuration and security policy updates to the physical network through the southbound interfaces. The main components of the NDT security architecture are described below.

Security data collector: This component collects security data and logs related to security operations. It identifies the security configurations made with the given interfaces, such as A1, O1, O2, and E2. This will also facilitate the efficient and up-to-date storage of large-scale security data.

Security data models: This should include the **Basic Network Model** blueprint and the **Security Functional Model** responsible for security services. The basic network model blueprint refers to the network element model and the network topology model, where the NDT is mapped with the accurate physical network in complete scale. The security functional models refer to the security data models that are assigned for security analysis, threat identification, and attack mitigation, which are established by making full use of the security data related to a specific application scenario. The security functional models can be defined with respect to the security analytical engine, security threat intelligence engine, and decision engine. The security data collector collects security data from both the physical network and DT and feeds those data to the security analytical engine to train AI/ML models.

Security management for DT entity management: This completes the security management functions of the NDT while also coordinating with topology and model management of data models. In particular, security management is responsible for all types of security operations related to NDT.

This may include confidentiality, integrity, and availability protection of the lifecycle of data security, model security, and interactive security of the NDT.

IV. Evaluation of FL-based Anomaly Detection

Here we describe the experimental setup, traffic analysis, model architecture and data processing.

A. Experimental set-up

As presented in the logical network architecture in Figure 2 and adhering to O-RAN reference architecture given in [5], the near-RT RIC can be connected to multiple E2 nodes and one non-RT RIC can be connected to multiple near-RT RICs. We consider an experimental architecture as shown in Figure 2 where one or more gNBs (i.e., a combination of RU, DU and CU) with an E2 interface connect to one near-RT RIC. Each gNB is able to support multiple traffic slices. In our experiment, we choose to use three broad 5G slices: enhanced mobile broadband (eMBB), massive machine type communication (mMTC), and ultra reliable low latency communication (URLLC). UEs are assigned to the appropriate traffic slices. The gNB records a wide range of Key Performance Indicators (KPIs) and periodically reports these KPIs to an xApp in the near-RT RIC. For each traffic slice, we generate both normal traffic and attack traffic that comprises anomalies.

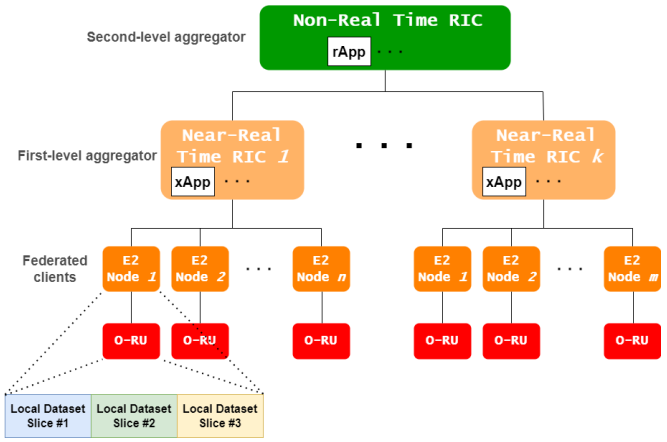

Fig. 2: Logical network architecture considered for simulations

1) Normal Traffic Class: For the normal traffic class, real-world 5G traces were collected in a variety of conditions for each traffic slice (i.e., eMBB, mMTC, URLLC) and stored in the security data collector. The packet arrival rates and payload sizes are based on real user traffic and do not follow a statistical distribution closely. Thus, we consider the UEs to have non-Independent and Identically Distributed (IID) data distribution. These traces are replayed in an O-RAN-compliant emulation environment using Colosseum [11] to generate realistic KPIs. Now, the Colosseum emulator behaves as the basic network model in our NDT. These KPIs are reported per UE basis for the xApp operating in the near-RT RIC every 250 ms. In this way, we generate a robust dataset for the normal traffic class that represents a wide range of real user traffic patterns.

2) Anomalous Traffic Class: To generate the anomalous traffic class, we develop two distinct attack models. The first model focuses on a User Datagram Protocol (UDP) Distributed Denial of Service (DDoS) attack, where an attacker-UE inundates the gNB with a substantial volume of UDP packets, thereby degrading system performance. To create this attack, we initially examine Packet Capture (PCAP) files from the malicious traffic dataset available in [12]. Drawing insights from this traffic analysis, we devise a statistical approach to simulate a UDP DoS attack within our Colosseum-based O-RAN environment. In this simulation, we model the DDoS attack by having each UE generate packets with an arrival rate λ determined by a Poisson distribution and packet sizes based on a Normal distribution. In our experimental scenario, we set $\lambda = 3.3 \times 10^{-5}$ seconds. For packet sizes, we employ two distinct distributions: $U_1 \sim \mathcal{N}(404, 100)$ and $U_2 \sim \mathcal{N}(1400, 1600)$ in bytes. We term this simulated attack as UDP_Poisson.

The second model introduces a more sophisticated attack variant known as the bandwidth hog attack. This attack represents an attempt to disguise a DDoS attack by closely mimicking realistic packet arrival rates. However, it employs artificially large packet sizes, leading to network congestion. To generate this attack, we utilize the original user traces but increase the payload size by adding $D = 70 + X$ bytes, where $X \sim \mathcal{N}(30, 100)$. All of the simulated attacks are stored in the security data collector.

B. Traffic analysis

Figure 3 illustrates the packet size behaviour of the realistic UDP DoS attack of [12] compared to our simulated UDP_Poisson attack in time-domain. The arrival packet sizes of the actual UDP DoS attack are approximated by the simulated attack using packet size U_1 (i.e., UDP_Poisson_t1). A larger packet size U_2 (i.e., UDP_Poisson_t2) is used to saturate the network further. This analysis reveals that we could approximate the real-world UDP DoS attack using the arrival rate and the size of the packets.

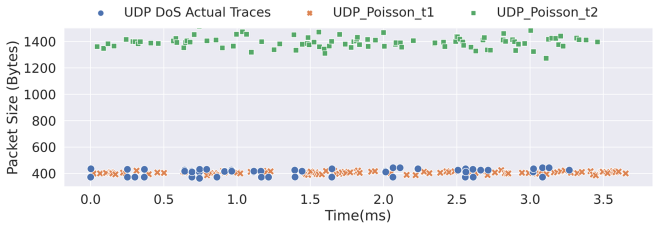

Fig. 3: Time domain analysis on packet size of realistic and simulated UDP_Poisson attacks.

TABLE I: DT classifier performances

Maximum Depth of Tree	Accuracy	Precision	Recall	F1-score
3	50%	23%	28%	23%

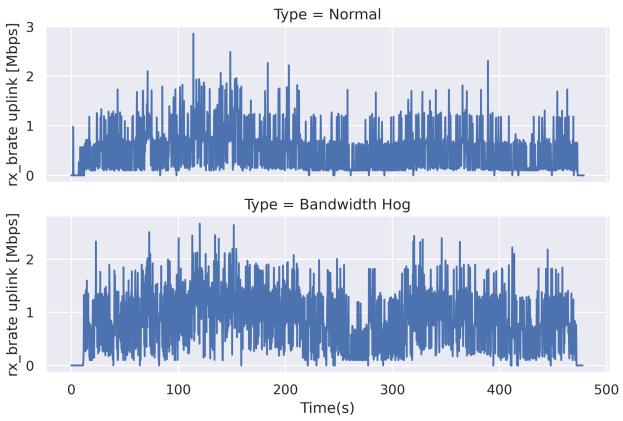

Fig. 4: Time domain analysis for rx_brate uplink [Mbps] feature in normal eMBB slice vs. eMBB slice with simulated bandwidth hog attack.

TABLE II: Decision rule for each of the classes based on DT classifier

Slice Type	**Decision rule**
eMBB	$0.473 <$ rx_brate uplink [Mbps] ≤ 0.731 **and** ul_n_samples > 93.5 **or** rx_brate uplink [Mbps] ≤ 0.731 **and** $22.5 <$ ul_n_samples ≤ 93.5
URLLC	rx_brate uplink [Mbps] ≤ 0.473 **and** ul_n_samples > 93.5
Anomaly	rx_brate uplink [Mbps] ≤ 0.731 **and** ul_n_samples ≤ 22.5 **or** $0.731 <$ rx_brate uplink [Mbps]

TABLE III: Dataset features and model hyperparameters

Feature	**Description**
dl_n_samples	Downlink number of samples transmitted since last time stamp
dl_buffer [bytes]	Downlink current queue length in bytes
tx_brate downlink [Mbps]	Downlink throughput in Mbps
tx_pkts downlink	Downlink number of packets transmitted since last timestamp
ul_n_samples	Uplink number of samples received since last time stamp
ul_buffer [bytes]	Uplink current queue length in bytes
rx_brate uplink [Mbps]	Uplink throughput in Mbps
rx_pkts uplink	Uplink number of packets received since last time stamp
LSTM Model Hyperparameters	**Description**
Optimizer	Adam
Learning rate	0.001
Batch size	128
Loss function	Cross entropy loss
Window size	100/50/10/4/1
Clusters	3
Local epochs	5
Cluster rounds	5
Global rounds	20

Figure 4 depicts the second variation in the UDP DoS attack, bandwidth hog, compared to normal slice traffic. The plots show the uplink throughput in Megabit per second (Mbps) in eMBB slice under normal conditions (top plot) and bandwidth hog attack (bottom plot) separately. The traffic patterns exhibited by the bandwidth hog closely resemble those of normal slice traffic, rendering it a significantly stealthy form of attack. During the traffic analysis, we tried to identify a linear classifier or a set of rules from the features to classify the traffic classes. In our study, we employed a Decision Tree (DT) classifier and visualized the resulting DT in Figure 5, and the detection accuracy for the classifier is given in Table I. Table II interprets the decision rules extracted from the DT. Although the classifier gives a human-readable rule-based logic to classify the slice traffic, the accuracy of the classifier needs to be improved. This discovery, however, serves as an eye-opening illustration of the formidable challenge of distinguishing between legitimate and malicious network traffic. It hints at the intrinsic complexity of the task, where both legitimate and malicious traffic can sometimes appear deceptively similar. This realization is a compelling motivation to adopt more sophisticated ML methods.

C. Model architecture and data processing

Our goal is to detect malicious UEs which are distributed across the network and to identify the slice type based on the KPIs generated at the gNB per UE. These KPIs are described in the Table III. From the traffic analysis, high usage of malicious UE's uplink channel highlighted the uplink side features given in Table III as relevant features for the anomaly slice detection. The accurate classification of normal traffic slices to eMBB, URLLC, and mMTC required both uplink and downlink features given in Table III. To analyze the traffic flows, we use a Long Short-Term Memory (LSTM)-based classification model. The main motivation is that LSTM is designed to avoid long and complex dependency issues, and can remember extensive historical information [13]. The model consists of one LSTM layer and one fully connected layer followed by a softmax layer (Figure 6). The hyperparameters used during the training are described in Table III.

Every client is considered to have one normal traffic slice (i.e., eMBB, mMTC, URLLC) and at least one anomalous traffic slice. The data pre-processing involves two main steps: data scaling and data windowing. Standardization was applied for data scaling, and data windowing was implemented to structure the data sequentially, enabling it to serve as input for the LSTM model. Given the temporal dependencies in the data, partitioning the time series data into training, testing, and validation sets was performed according to sequential time values. The first 80% of the time-series records were allocated to the training set. The subsequent 20% of the time-series records were designated as the testing set. For the validation of the model during training rounds, the last 20% of time records from the training set were extracted and defined as the validation set.

To comply with the hierarchical and distributed architecture as presented in Figure 2, where data is distributed across E2 nodes, we use FL for the anomaly detector training process. Another purpose of using FL in ML model training in NDT is to avoid model decay when deployed in the production environment. In other words, the performance of a centrally

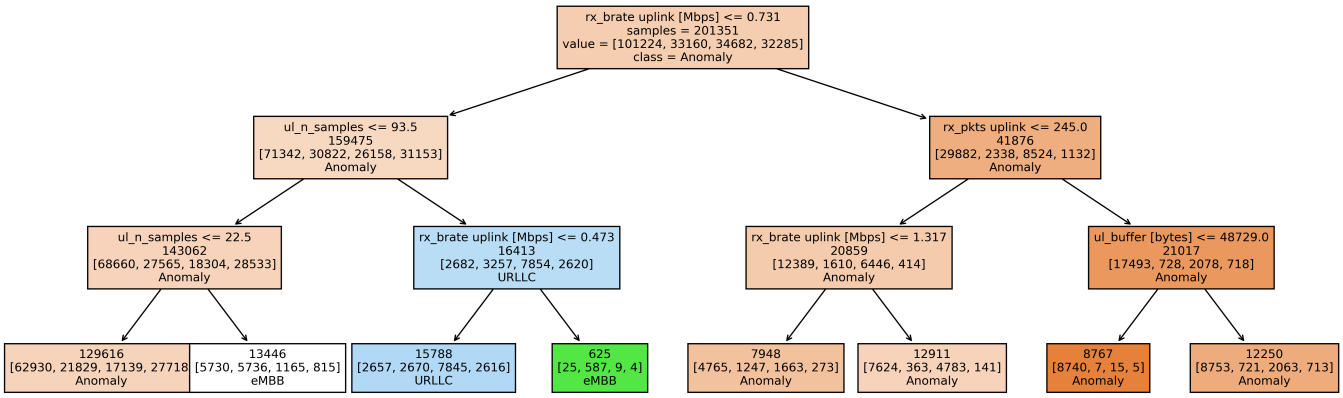

Fig. 5: DT visualization for the slice traffic classification.

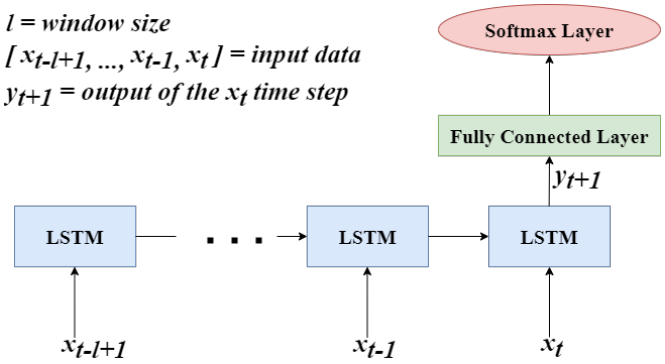

Fig. 6: LSTM classification model architecture.

trained model in a federated environment may not be as good as a model trained in a federated manner in the same environment [14].

We consider E2 nodes as federated clients where each client has local traffic slice data to train their local models. The model aggregation is performed at two levels separately, near-RT RIC and non-RT RIC. In the first level of aggregation, one near-RT RIC and its corresponding E2 nodes are taken as one cluster. First, federated client models in E2 nodes are trained utilizing locally available data, and the model vectors are communicated to the relevant first-level aggregators hosted in near-RT RICs. After the aggregation, the calculated aggregator model vector is sent back to each local trainer in the same cluster, indicating the end of a cluster round. Furthermore, after a pre-defined number of cluster rounds, the first-level aggregators communicate their model vectors to the second-level aggregator residing in a centralized non-RT RIC. Then, the global aggregator performs the averaging of the received models and transmits the global model vector back to all the federated clients by completing one FL global round.

V. Numerical Results and Discussion

This section presents the numerical results and the observations we make on them. One interesting behaviour is that when attempting to simulate the DoS attack, UE uplink capacity saturates before the gNB slice PRBs saturate. Therefore, a single UE cannot perform a simple DoS on a gNB. Thus, the given attack scenario was simulated as a DDoS attack where multiple UEs target to saturate the gNB slice PRBs. Identifying such limitations and proactive emulation of network attacks in a controlled environment are benefits of incorporating NDT, which can help improve the security of the O-RAN framework.

Synthetic data has been used to train FL models offline, as real-world traffic is initially unlabeled. The security data collector in NDT is considered a trusted data source. RIC platforms are trusted entities that are used for model aggregation and initialization. E2 nodes can be malicious. The purpose of bringing FL model training to E2 nodes is to improve detection time and reduce privacy risks. Once the FL model is deployed in the live network, it will continue to train using real-world traffic. Proposing training and inference to E2 nodes prevents attackers from getting traction beyond E2 nodes and reduces the communication overhead in the E2 interface. ML model training in NDT was conducted as FL since centrally trained ML models lose performance in a distributed environment. It has shown that the ML model needs a few training rounds in the federated setting to achieve the previous accuracy [14]. During this recovery time, the network is exposed to a potential attacker.

During evaluations, different ML models were considered to be the anomaly detector. DT, LSTM autoencoders as an unsupervised learning technique, and LSTM classification model as a supervised learning technique. The hyperparameter tuning was also done extensively for all the possible methods. Only the LSTM classification model showed promising results. This depicts the benefit of having the NDT as the offline learning platform for the O-RAN, which avoids this kind of risk in ML model deployment in the actual network.

Performance results for the trained LSTM classification model are provided in Table IV. Model performance increases as the window size is reduced. This is a remarkable improvement compared to the baseline results for the DT classifier

given in Table I. The confusion matrix for the trained model with window size one is provided in Figure 7. We observed noticeable accuracy enhancement in traffic classification with the window size reduction. This indicates our dataset has shorter temporal dependencies. Hence, the model could make real-time predictions based on our dataset. This emphasizes the significance of NDT, which could be utilized to fit the model's configuration to the particulars of the data.

TABLE IV: Performance results for LSTM classification model with different window sizes

Window size	Accuracy	Precision	Recall	F1-score
1	99.87%	99.87%	99.83%	99.85%
4	99.73%	99.77%	99.65%	99.71%
10	99.6%	99.6%	99.49%	99.54%
50	96.65%	94.21%	95.35%	94.44%
100	95.34%	93.04%	93.48%	92.0%

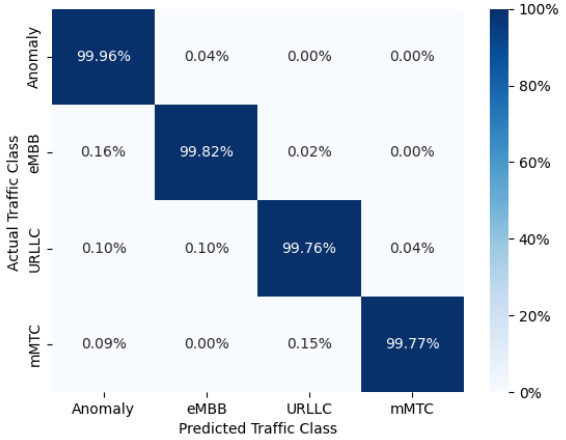

Fig. 7: The confusion matrix for LSTM classification model with window size one.

A key observation is that if a UE transmits anomalous traffic only (i.e., no normal traffic), our model is able to identify malicious UEs with 100% accuracy under any window size. If the UEs have both normal and anomalous traffic, classification accuracy depends on the window size as given in Table IV.

VI. Conclusion and Future Directions

This paper has presented a comprehensive security framework for O-RAN, incorporating the NDT architecture as a robust platform for the evaluation and training of security algorithms. The security architecture is described inside the NDT, enabling the training of ML models, poised to play a crucial role in enhancing O-RAN security. We have demonstrated the effectiveness of this framework through a hierarchical FL-based anomaly detection algorithm that detects anomalous traffic across three traffic slices in O-RAN. We also have proven that a simple rule-based traffic classifier is insufficient to detect anomalies with a higher accuracy. Our trained LSTM model exhibits exceptional accuracy in detecting anomalous traffic, marking a significant step forward in securing the O-RAN ecosystem. We have shown that ML models can be trained in NDT using simulated scenarios before deploying them in the physical network in accordance with the general principles in the O-RAN framework [3]. With this NDT architecture, we intend to achieve adaptive model learning by periodically feeding network data, which may also include zero-day anomalies. ML workflow proposed in the NDT can be deployed in the O-RAN framework in future architecture versions to process real-time network traffic to detect anomalies, identify root causes, and predict potential attacks. The performance of the LSTM classification model should be further validated using more sophisticated attacks, such as variants of DDoS attacks, botnet attacks, and jamming attacks.

Acknowledgment

This work was supported by these projects: Hexa-X-II (Grant Agreement no. 101095759), funded by EU HORIZON-JU-SNS-2022 call; XcARet, funded by Academy of Finland.

References

[1] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[2] M. Q. Hamdan, H. Lee, D. Triantafyllopoulou, R. Borralho, A. Kose, E. Amiri, D. Mulvey, W. Yu, R. Zitouni, R. Pozza, B. Hunt, H. Bagheri, C. H. Foh, F. Heliot, G. Chen, P. Xiao, N. Wang, and R. Tafazolli, "Recent advances in machine learning for network automation in the o-ran," *Sensors*, vol. 23, no. 21, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/21/8792

[3] O-RAN ALLIANCE, "AI/ML workflow description and requirements," O-RAN.WG2.AIML-v01.03, 2021.

[4] D. Attanayaka, P. Porambage, M. Liyanage, and M. Ylianttila, "Peer-to-peer federated learning based anomaly detection for open radio access networks," in *Proceedings of IEEE International Conference on Communications*, 2023.

[5] "O-ran alliance." [Online]. Available: https://www.o-ran.org/

[6] I. Vilà, O. Sallent, and J. Pérez-Romero, "On the design of a network digital twin for the radio access network in 5g and beyond," *Sensors*, vol. 23, no. 3, p. 1197, 2023.

[7] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, Jan. 2022.

[8] A. Masaracchia, V. Sharma, M. Fahim, O. A. Dobre, and T. Q. Duong, "Digital twin for open ran: Towards intelligent and resilient 6g radio access networks," *IEEE Communications Magazine*, 2023.

[9] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Federated deep reinforcement learning for resource allocation in o-ran slicing," in *IEEE Global Communications Conference*. IEEE, 2022, pp. 958–963.

[10] Y. Cao, S.-Y. Lien, Y.-C. Liang, and K.-C. Chen, "Federated deep reinforcement learning for user access control in open radio access networks," in *IEEE International Conference on Communications*, 2021.

[11] L. Bonati, P. Johari, M. Polese, D'Oro *et al.*, "Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2021, pp. 105–113.

[12] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *International Carnahan Conference on Security Technology (ICCST)*, 2019, pp. 1–8.

[13] Y. Li and Y. Lu, "LSTM-BA: DDoS detection approach combining LSTM and Bayes," in *IEEE international conference on advanced cloud and big data (CBD)*, 2019, pp. 180–185.

[14] J. Nguyen, J. Wang, K. Malik, M. Sanjabi, and M. Rabbat, "Where to begin? on the impact of pre-training and initialization in federated learning," 2022.