

Rogue Cell: Adversarial Attack and Defense in Untrusted O-RAN Setup Exploiting the Traffic Steering xApp

Eran Aizikovich, Dudu Mimran, Editia Grolman, Yuval Elovici, Asaf Shabtai
Ben Gurion University of the Negev

Abstract

The Open Radio Access Network (O-RAN) architecture is revolutionizing cellular networks with its open, multi-vendor design and AI-driven management, aiming to enhance flexibility and reduce costs. Although it has many advantages, O-RAN is not threat-free. While previous studies have mainly examined vulnerabilities arising from O-RAN’s intelligent components, this paper is the first to focus on the security challenges and vulnerabilities introduced by transitioning from single-operator to multi-operator RAN architectures. This shift increases the risk of untrusted third-party operators managing different parts of the network. To explore these vulnerabilities and their potential mitigation, we developed an open-access testbed environment that integrates a wireless network simulator with the official O-RAN Software Community (OSC) RAN intelligent component (RIC) cluster. This environment enables realistic, live data collection and serves as a platform for demonstrating APATE (adversarial perturbation against traffic efficiency), an evasion attack in which a malicious cell manipulates its reported key performance indicators (KPIs) and deceives the O-RAN traffic steering to gain unfair allocations of user equipment (UE). To ensure that O-RAN’s legitimate activity continues, we introduce MARRS (monitoring adversarial RAN reports), a detection framework based on a long-short term memory (LSTM) autoencoder (AE) that learns contextual features across the network to monitor malicious telemetry (also demonstrated in our testbed). Our evaluation showed that by executing APATE, an attacker can obtain a 248.5% greater UE allocation than it was supposed to in a benign scenario. In addition, the MARRS detection method was also shown to successfully classify malicious cell activity, achieving accuracy of 99.2% and an F1 score of 0.978.

1 Introduction

In recent years, network operators and vendors have begun exploring innovative radio access network (RAN) architectures [13, 18]. The RAN provides wireless connectivity to

mobile devices and acts as the final link between the cellular network and user equipment (UE). Traditionally, RANs have been vendor-specific, i.e., with hardware interfaces and applications optimized for a specific vendor’s equipment, and were operated by a single owner. While this approach enables vendors to deliver integrated and highly optimized solutions, it also has significant drawbacks. Traditional RANs require vendors to develop all components, driving up costs for network operators and creating vendor lock-in, which limits flexibility and innovation. To overcome these drawbacks, an advanced architecture was proposed, the Open Radio Access Network (O-RAN) [56].

O-RAN, which was introduced by the O-RAN Alliance¹, started to gain attention, as it allowed: (1) open accessible shared information to promote multi-vendor deployments, with standardized interfaces between RAN components [30, 40]; (2) adaptivity in real time by using cloud-based and virtualized components managed through software-defined networking (SDN), which enables more flexibility and reduces operational costs [32, 69]; and (3) adaption of the RAN intelligent component (RIC), which is responsible for utilizing artificial intelligence (AI) and machine learning (ML) systems, to reduce human intervention [48, 54].

While previous studies have begun to uncover vulnerabilities in O-RAN’s intelligent components—particularly those exploited by malicious UE for personal gain [44]—the shift to multi-operators in RAN architectures introduces new security challenges. Traditional RANs, where a single operator oversees the entire network, were considered fully trusted. However, the disaggregation of the O-RAN architecture has accelerated the shift toward multi-operator networks, resulting in an untrusted environment since different operators manage various network elements [33, 64].

In this paper, we demonstrate a threat model in which the threat actor is the cell (i.e., a cellular operator). An adversarial cell can disrupt the network in several ways, such as reducing UEs migration to other cells or executing other

¹The O-RAN Alliance: <https://www.o-ran.org/who-we-are>

denial-of-service (DoS) attacks. These types of attacks can be financially driven, since operators may be compensated based on the amount of UE their cells serve. To illustrate this threat, we introduce the APATE (adversarial perturbation against traffic efficiency) attack.

The APATE is an evasion attack [10] targeting the traffic steering (TS) flow (part of O-RAN RIC use cases [50]) which is responsible for dynamically and intelligently managing network traffic. The attacker misleads the TS flow into assigning additional UE to its cell by falsely manipulating the TS’s ML model that is responsible for the quality of experience (QoE) predictions (i.e., attacking the QoE predictor referred to as the QP). The APATE attack works as follows: The attacker trains a substitute model of the QP model and uses this model to craft adversarial samples, i.e., samples that mislead the QP and cause it to make an incorrect prediction. The crafted adversarial samples are then used to query the QP target model, maliciously leading it to forecast an artificially high QoE for the attacker’s cell, by misleading the TS, resulting in an unfair allocation of UE to the malicious cell.

To mitigate these type of risks and ensure that O-RAN’s legitimate activity continues, we propose a mitigation strategy called MARRS (monitoring adversarial RAN reports), specifically designed to detect untrusted actors attempting to disrupt legitimate network operations, such as in attacks like APATE. MARRS is a detection method based on a long short-term memory (LSTM) autoencoder (AE) architecture [34]. MARRS extracts relevant time-series features from the reporting cells and UEs key performance indicators (KPIs) and trains a dedicated AE model for each cell. Then, it uses the compressed latent space from each AE model, concatenated with the aggregated latent spaces from all other cells, to generate new enriched feature vectors that capture contextual information from both the specific cell and the entire network. Next, an additional AE for each cell using the new enriched feature vector is trained, however this time in an attempt to reconstruct the original features. Finally, a classifier is trained to compare the second model’s output to the first model’s input; if the loss between them exceeds a predefined threshold, then the input is classified as untrusted; otherwise it is classified as trusted. The entire training process is performed using benign data to learn benign data behavior, i.e., unsupervised learning. At inference time, MARRS’s classification will indicate whether the examined cell’s KPIs are benign or adversarial.

To demonstrate the APATE attack and evaluate the MARRS detection method, we developed a testbed with a wireless network simulator [17] to emulate a live network topology with moving UE and gNBs (5G cell base stations) and an O-RAN Software Community (OSC) RIC cluster [11] hosting the TS ML models. The use of these components in the testbed enables realistic simulations where UE and cells regularly report KPIs to the RIC and receive real-time reallocation handover requests based on TS handover decisions.

We assess APATE’s impact by simulating two scenarios: (1) a normal benign scenario, and (2) a malicious scenario where one cell within the environment executes an attack. The results of our experiments in the testbed show that on average an adversarial cell was able to obtain a 248.5% greater UE allocation than it was supposed to in a benign scenario. To evaluate our proposed detection method, we test it on simulated malicious scenarios aiming to classify cells’ KPIs reports as trusted or untrusted, to identify malicious activity. MARRS detection method successfully identified malicious cell activity in the test scenarios achieving an accuracy of 99.2% and an F1 score of 0.978.

The main contributions of this paper can be summarized as follows:

1. Present a threat model that takes into account the vulnerabilities resulting from O-RAN’s openness and multi-operator untrusted structure. We demonstrate an attack (APATE) in which a cell is the threat actor targeting the TS flow to obtain more UE to serve.
2. Publish an open-source testbed environment that includes a closed-loop wireless network simulator connected to the OSC RIC cluster that enables realistic and live data collection.
3. We also present MARRS; a practical AI-based detection method to mitigate this attack and future attacks based on this threat model.

2 Background

2.1 O-RAN Architecture

Traditional RAN components are monolithic, vendor-provided black boxes that integrate all layers of the cellular protocol stack. This design limits reconfigurability, hinders coordination between network nodes, and locks operators into specific vendors. In addition, the vendor develops all the components, which drives up costs for network operators and creates vendor lock-in, which limits flexibility and innovation. To address these challenges, O-RAN has emerged as a new paradigm that leverages disaggregated, virtualized, and software-based components connected through open, standardized interfaces. This approach enhances flexibility, supports multi-vendor ecosystems, and enables intelligent, data-driven closed-loop control. By adopting cloud-native principles, O-RAN improves RAN resiliency, adaptability, and innovation potential.

The O-RAN specifications build on the 3GPP long-term evolution (LTE) and new radio (NR) standards, extending the 3GPP NR 7.2 functional split for base stations [56]. This split disaggregates base station functions into three distinct components: the central unit (CU), distributed unit (DU),

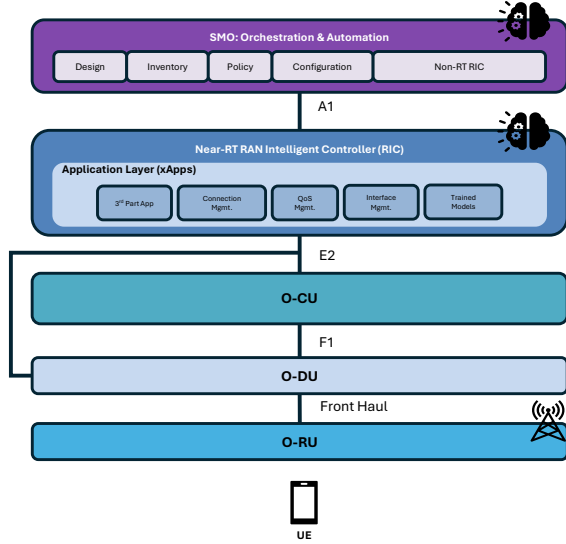


Figure 1: O-RAN architecture high-level overview.

and radio unit (RU), enabling greater flexibility and modularity. These units connect to RICs via open interfaces, enabling the streaming of RAN telemetry and the deployment of control actions and policies. The O-RAN architecture's components and their open interfaces are illustrated in Fig. 1.

2.2 The RAN Intelligent Controllers RIC

The RIC is a key element in the O-RAN architecture, introducing programmable components capable of executing optimization and intelligent routines with closed-loop control. RICs often leverage AI and ML models for various tasks like network slicing, handovers, and scheduling policies. By doing so, RICs significantly enhance network efficiency and performance optimization. Some of the main key benefits of RICs include: Improved network performance and efficiency through AI and ML-driven optimization [9]. Increased flexibility and programmability of the RAN [15]. Reduced operational costs through automation and intelligent resource management [32]. Improved UE experience through more granular and intelligent control of network resources [59]. The RIC as described in the O-RAN specifications consists of two primary levels: the near-real-time RIC (near-RT RIC) and the non-real-time RIC (non-RT RIC) [49, 50].

The **Near RT RIC** acts as the near-real-time decision-making core of the network. It operates with control loop periodicities ranging from 10 milliseconds to 1 second, enabling it to manage dynamic adjustments within the network in near real-time. Designed to oversee multiple RAN nodes (DUs or CUs), the near-RT RIC can influence the quality of service (QoS) for hundreds or even thousands of UE connections. The near-RT RIC hosts specialized applications known as xApps, which perform essential tasks such as RAN

data analysis, traffic steering, and network control. These xApps communicate with the RAN via the E2 interface, receiving real-time data and sending control commands back to adjust network behavior. Additionally, the near-RT RIC provides APIs and services to support the automated lifecycle management of xApps, including onboarding, deployment, and termination. These capabilities ensure seamless internal messaging, conflict mitigation, and operational stability across the network [49].

Non-RT RIC The non-RT RIC is responsible for longer-term (operating on timescales greater than one second) network optimization and policy management to support the near-RT RIC via the A1 interface (e.g., the threshold for UE reallocation, in the TS xApp). It hosts applications called rApps, which can handle tasks such as network slicing, energy saving, and AI/ML model training [50].

2.3 Traffic Steering (TS)

One of the most important tasks of the near-RT-RIC is the TS task, as it is responsible for managing the UE cells' connection in the network. The TS flow illustrated in Fig. 2 demonstrates a network topology where UE is located within the reception range of cells A, B, and C and can be connected by each of these cells. The network operator needs to make a decision regarding which cell the UE should be connected to. There are several TS approaches to make this decision. Many handovers for the TS task are performed using reinforcement learning (RL), due to its policy-based decision-making nature [52, 57]. Another way to perform the TS task is to connect UE to a cell based on the maximum received signal reference power (RSRP). As a UE moves away from its serving cell, the RSRP from that cell decreases over time, while the RSRP from a nearby target cell increases as the UE approaches it [66]. One of the most common approaches presented in the OSC, is the QoE prediction, which is based on QoE prediction for potential new target cells (this is the approach we followed in this paper) [19]. In this TS approach, there are four main xApps involved: KPI monitoring (KPIMON), anomaly detection (AD), QP, and TS. The TS flow illustrated in Fig. 2 contains the following steps; The KPIMON receives telemetry from the cells regarding the network status (e.g., cells and UE's KPIs) and writes them into the RIC influx database (DB). The AD xApp, scheduled to run every 10ms, identifies UE with an anomalous QoE that might need reassignment to another cell. Detection is performed by a pretrained isolation forest (iForest) model, based on UE metrics extracted by the KPIMON xApp and stored in the RIC. The AD xApp sends this list of anomalous UEs to the TS xApp for reallocation. Then, the TS calls the QP xApp for QoE prediction for each neighbor cell of all UE in the list. The QP trains a vector autoregression (VAR) model for every UE neighbor cell, forecasts the QoE, and sends it back to the TS. Finally, based on the QP predictions and a

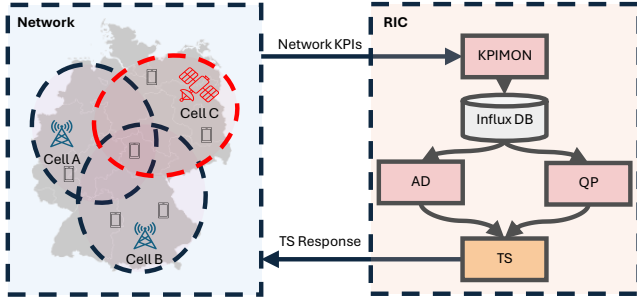


Figure 2: High-level overview of the relevant components in the TS flow.

given A1 policy from the non-RT-RIC, for each UE, the TS decides whether it should stay in its current serving cell or be handed over to a new target cell.

2.4 Multiple Operators Deployments

The evolution of telecommunications infrastructure has witnessed a significant shift toward disaggregation, where different network elements are operated by distinct entities (such as different companies). While multi-vendor deployments have been a longstanding practice in telecommunications, the emergence of multiple operators managing different network segments introduces new trust considerations. This paradigm shift predating O-RAN was primarily driven by cost-efficiency considerations and the understanding that specialized capabilities could be better managed through outsourcing arrangements [24, 41, 42, 46, 51].

The infrastructure-sharing model gained particular prominence with the rise of tower companies like American Tower² and Vantage Towers³. These companies emerged as spin-offs from traditional telecom operators and have since evolved into independent entities managing critical network infrastructure. The participation of third-party mobile tower companies has also led to an increased incidence of base stations (BS) colocation [51]. For example, in the U.S., companies sublease space from independent landlords to deploy BSs belonging to more than one operator on the same premises [67]. In the United Kingdom, Freshwave deployment enables four mobile operators to share the same indoor small cell infrastructure, demonstrating the feasibility of extensive operator collaboration [20, 71]. In Germany, Deutsche Telekom and Telefónica Deutschland established a major infrastructure-sharing agreement, in which thousands of mobile sites were covered through reciprocal network access [39]. Similarly, Vodafone and Orange have implemented extensive network sharing in Spain, particularly focusing on rural and suburban areas [27]. Recent develop-

ments have also facilitated innovative deployment scenarios. For example, the integration of satellite networks as RU/DU units [21] with cellular backbones represents an extreme case of a multi-operator scenario, highlighting both the potential and challenges of such arrangements [37, 75]. Companies like AST SpaceMobile⁴, Lynk Global⁵ and Starlink [37] are developing satellite-to-cellular solutions that will integrate with terrestrial networks, creating new multi-operator paradigms.

3 Related Work

3.1 O-RAN Security

New concepts and technologies are continually being introduced into the RAN, each bringing new cybersecurity threats and significantly expanding the RAN’s attack surface [2, 6, 54, 56, 63]. O-RAN security focuses on several key areas. Recent attacks on traditional RANs are reviewed to assess their applicability to the O-RAN architecture [61]. The open-source and disaggregated nature of O-RAN introduces unique threats, necessitating analysis of the security implications of the O-RAN architecture [44]. Specific vulnerabilities arise from O-RAN’s openness, particularly in xApp access control and the E2 interface, which could allow unauthorized access and manipulation of network policies [31]. The classification of various security-related risks specific to O-RAN includes inadequate logging, lack of encryption, and insufficient access controls, which can lead to security breaches and data integrity problems [38]. The comprehensive summary of the security threats, requirements, and recommended mitigation strategies associated with the O-RAN framework provided by Park et al. [53] highlights the steps required to strengthen the architecture’s resilience against emerging threats. However, as emphasized by Park et al. [53], there are many remaining security risks, as we will demonstrate in this paper.

3.2 Attacks on O-RAN

Adversaries may exploit the inherent vulnerabilities of learning algorithms, and specifically ML algorithms, with various attack techniques, which are referred to as adversarial machine learning (AML) [7, 8, 29, 60]. Recent work [29] provided a comprehensive threat assessment of ML usecases within O-RAN according to a common cybersecurity risk assessment (NIST ontology). In their work, the authors outline the potential adversaries, their capabilities, and their goals, identify threats to ML production systems within O-RAN, and enumerate attacks that can materialize these threats. In addition to their threat modeling, the authors demonstrated

²American Tower: <https://www.americantower.com/why/history>

³Vantage Towers: <https://www.vantagetowers.com/en>

⁴AST: <https://ast-science.com/spacemobile-network/>

⁵Lynk Global: <https://lynk.world/>

Table 1: Summary of Related Work

Paper	Key Points	Actor	Attack	Mitigation
[29]	A novel AML threat assessment methodology with practical demonstrations and tools for high-risk threats in ML-based network management	UE	Evasion & Poison	-
[68]	Anomaly traffic detector that enhances network security by mitigating DoS attack executed by UE on RIC xApps	UE	DoS	ML-Based
[4]	Presents how compromised KPIs can poison the RIC closed loop followed by an LSTM detection method	-	Poison	LSTM
[62]	ML-based detection for preventing DDoS attacks executed by malicious UE	UE	DDoS	ML-Based
[14]	Fast ML-based detection for DDoS attacks on O-RAN	UE	DDoS	ML-Based
[26]	MiTM attack that exploits the open interfaces and poisons the network slicing xApp	-	Poison	DRL AE
[60]	Presents how a malicious xApp reduces the network capacity by performing FGSM and PGD attacks	xApp	Evasion	-
[8]	Evasion attack on the connection management xApp, with defense approaches	xApp	Evasion	Adv Training
Our Paper	Attack by untrusted cell in O-RAN with AI-driven detection	Cell	Evasion	LSTM-AE

how UE can produce manipulated signals that lead to incorrect anomaly detection and QoE classification. By doing so, the UE influences the model’s decision-making process by presenting inputs that are very similar to legitimate data but designed to cause misclassification or incorrect predictions [29].

Sapavath et al. [60] conducted experiments in an O-RAN testbed to demonstrate that both the fast gradient sign method (FGSM) and projected gradient descent (PGD) attacks can effectively manipulate input data for the xApp interference classifier, leading to misclassification. Even minimal adversarial perturbations (i.e. small modifications to input data designed to mislead ML models) have been shown to drastically impair the xApp’s accuracy, which consequently reduces network capacity and increases overall bit loss within the O-RAN system [60].

The authors of [7] analyze O-RAN WG11’s [5] threat model and risk assessment methodology, focusing on DoS and performance degradation threats. They identify specific vulnerabilities, mapping them to potential attacks across critical O-RAN interfaces. Through experiments using an O-RAN deployment, the authors evaluated the impact of DoS and performance degradation attacks on these interfaces, assessing their resilience in various attack scenarios [7].

3.3 O-RAN Attack Mitigation

Growing concerns over attacks on O-RAN ML systems have prompted research on mitigation of such attacks.

Both white-box and black-box evasion attacks have been demonstrated on deep reinforcement learning (DRL)-based traffic steering [52], revealing the impact of adding noise into the UE’s metrics. These attacks showed that PGD attack can

reduce coverage rates by as much as 50%, while jamming attacks result in up to a 25% reduction in coverage. To counter these effects, two mitigation techniques were proposed: (1) adversarial training [25], and (2) regularized training [74]. Both techniques were found to improve model robustness against such attacks [8].

Recent work by Xavier et al. [72] introduced an effective mitigation strategy for several types of DoS attacks. This approach employs ML models to analyze air interface measurements, enabling the early detection of malicious traffic before it disrupts network services. In their followup work, the authors improved their mitigation strategy which is supposed to mitigate all types of DoS attacks while improving its ability to be deployed on real systems [73].

Research directly related to our study on attacks executed by malicious KPIs reaching the RIC has also been performed. In [26], the authors highlighted the vulnerabilities introduced to the intelligent components of O-RAN due to the adoption of open interfaces. Man-in-the-middle attack (MiTM) attackers were shown to be able to inject malicious KPI reports into the E2 interface targeting the near-RT RIC or deliver malicious control actions from the near-RT RIC to E2 nodes. They specifically demonstrated AML attacks on the input KPI reports of a network slicing xApp. To mitigate such threats, the authors proposed a method based on AEs to detect this threat [26]. Another study [4] built on this type of poisoning attack, emphasizing the critical role of KPIs in near-RT RIC control loop use cases. The authors proposed an LSTM model for detecting anomalous KPIs, which was shown to be a robust approach for mitigating such attacks in their evaluation.

To detect Distributed DoS (DDoS) attacks et al [14] demonstrates that XGBoost achieves high precision and re-

call with the fastest execution time compared to random forest and multilayer perceptron, ensuring operations remain within latency requirements. Another study [62] presents a ML-based framework for detecting DDoS attacks in O-RAN, utilizing dApp and xApps to enhance real-time threat detection, while balancing speed and accuracy. It evaluates multiple ML algorithms to identify the best-fit models for anomaly detection and service usage tracking, addressing O-RAN's unique challenges.

To the best of our knowledge, no recent research has considered network cells as untrusted elements in the RAN as potential threat actors or proposed mitigation strategies for such types of attacks. The related works [4, 8, 14, 26, 29, 60, 62] performed on O-RAN attacks and mitigation is summarized in Table 1.

4 Threat Model

We present a threat model based on the NIST ontology for modeling an enterprise security [12, 65]. The proposed threat model considers the following assumptions:

(1) **Multiple Operators RAN Deployments.** The shift of telecommunications infrastructure to disaggregation resulted in different network elements being operated by distinct entities, allowing the reduction of operational cost, achieving both capital expenditures (CAPEX) and operational expenditures (OPEX) savings [24, 41, 42, 46, 51]. Implementation reports and real-world scenarios to support the feasibility of multi-operator deployment are detailed in the background section (Section 2). In addition, as described in 3GPP specifications, multiple cell operators agree on sharing a coverage area taking into account the load balancing between the cells [1]. Similarly, GSMA also released a specification on infrastructure sharing, describing standards for site, tower, RAN and core network sharing [28].

(2) **Financial Model.** In multi-operator deployments, when an operator lacks the resources to serve its clients (UEs), services are provided through a third-party operator. According to Farhat et al. [22, 23] UE payments go to their home operator, and the latter must pay a service price (transaction cost) to the new access operator.

Under these assumptions, an attacker is a malicious cell operator carrying an AML attack targeting the O-RAN TS flow. The attack goal is to gain unfair UEs allocation, thereby increasing its revenue. Such manipulation can degrade the QoE for the victims' UEs and reduce the income of its neighboring benign operators.

The threat model entities and their relations based on NIST ontology as illustrated in Fig. 3 are described as follows:

- **Attacker:** A malicious operator running a malicious cell operating in the O-RAN network. In the remainder of the paper, we will use the term malicious cell.

- **Adversarial Capabilities:** (a) The malicious cell can manipulate the KPIs it reports to the RIC. (b) The malicious cell has knowledge of the targeted TS task flow. This capability follows SOTA AML threat analysis in O-RAN [29].
- **Threat:** A malicious cell disrupts the TS process by influencing the QP model's QoE predictions, resulting in an unfair allocation of UE to the malicious cell.
- **Vulnerability:** Refers to the inherent ability to manipulate the input of the ML model used by the QP, causing it to inaccurately predict QoE.
- **Technique:** Query-based evasion attack.
- **Assets:** The TS task hosted on the near-RT RIC, responsible for the allocation of UE to cells.
- **Impact:** The attacker is able to serve more UE than it is supposed to, which can: (a) malicious cell to receive payment for providing the service to UEs that would gain better service from other cells. (b) reduce the QoE of UE affected by the attack, since providing service to UE that would receive a better QoE by other cells.

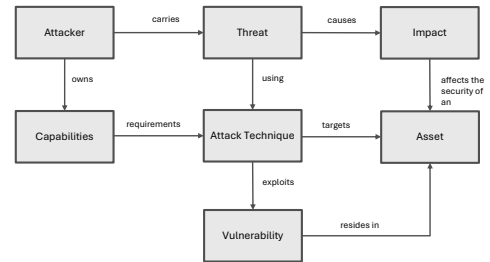


Figure 3: Threat analysis based on NIST ontology [65].

5 APATE Attack

The APATE attack (adversarial perturbation against traffic efficiency), an attack designed to manipulate network TS by attacking the QP model is illustrated in Fig. 4. The attacker's objective is to maximize the amount of UE assigned to attacker's service. To achieve this, the attacker manipulates its own KPIs to mislead the QP model into forecasting a higher QoE than the actual QoE. To execute this attack, the malicious cell needs to know which TS approach (see Section 2.3) is implemented on the network; this will allow the attacker to replicate the behavior of the target model.

The attack unfolds in three main stages: (1) The attacker begins by training a substitute QP model, replicating the behavior of the target model used in the TS flow. (2) Using the substitute model, the attacker learns the model's decision

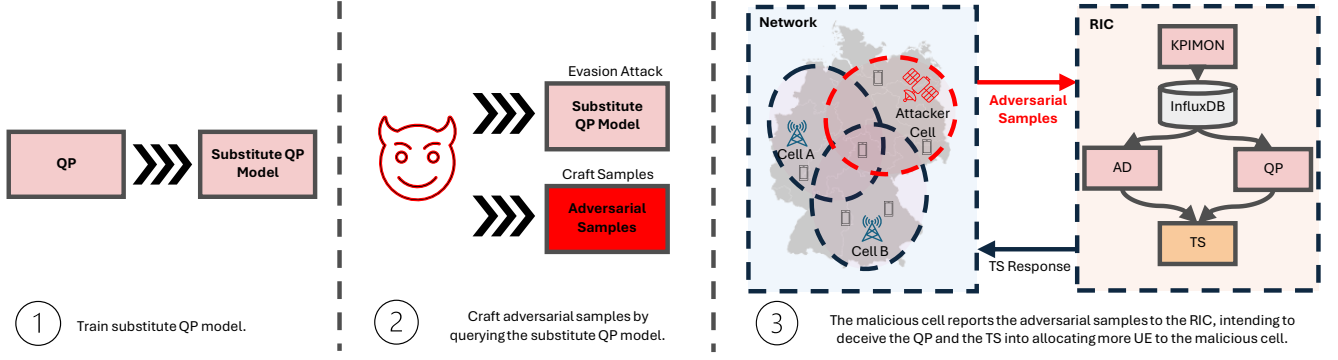


Figure 4: Attack flow steps: (1) the attacker trains a substitute QP model, replicating the behavior of the target model; (2) the attacker employs an adversarial evasion attack to generate adversarial samples; and (3) the attacker reports its generated adversarial samples to mislead the TS to allocate more UE.

boundaries and behavior. By analyzing the decision boundaries, the attacker identifies the minimal input perturbations that need to be added to the cell's KPIs to manipulate the QP model. Then the attacker performs an adversarial evasion attack to generate adversarial samples. (3) The crafted adversarial samples, representing the perturbed KPIs, are reported to the RIC as legitimate data. These adversarial samples are written to the RIC database by the KPIMON xApp. When the AD xApp detects that UE might require cell allocation, the TS xApp requests the QP xApp for a QoE prediction for the potential new target cells (as detailed in Section 2.3). At this point, the QP model predicts an artificially high QoE for the malicious cell based on the adversarial samples, causing the TS xApp to allocate UE to the malicious cell.

The proposed attack is formally described as follows: let $N = (V^{cl}, V^{ue}, E)$ denote the network bipartite graph where $V^{cl} = \{v_1^{cl}, v_2^{cl}, \dots, v_n^{cl}\}$ are the cell's nodes and $V^{ue} = \{v_1^{ue}, v_2^{ue}, \dots, v_m^{ue}\}$ are the UE's nodes, and n and m are respectively the number of cells and number of user equipments. The edges $E \subseteq V^{cl} \times V^{ue}$ are defined by serving connections between pairs of cells and UE. For example, if v_x^{cl} is the serving cell of UE v_y^{ue} , then $(v_x^{cl}, v_y^{ue}) \in E$.

The objective of the attacker cell v_{adv}^{cl} is to find a perturbation noise δ that can be added to its KPI reports R , such that the QP model will predict a higher QoE than it should as presented in Eq. (1) where δ^* represents the optimal perturbation, Q is the QP model, and y is the true QoE prediction. The attacker sends these crafted adversarial samples to the RIC where they are incorporated into the TS flow, as detailed in Section 2.3. This manipulation leads the QP model to overestimate the QoE for the adversarial samples, potentially resulting in unjustified greater UE allocations (v_{adv}^{cl}, v^{ue}) to the adversarial cell.

$$R_{adv} = R + \delta^* \quad (1a)$$

$$\delta^* = \arg \max_{\delta} (\mathcal{L}(Q(R + \delta), y)) \quad (1b)$$

6 MARRS Detection Method

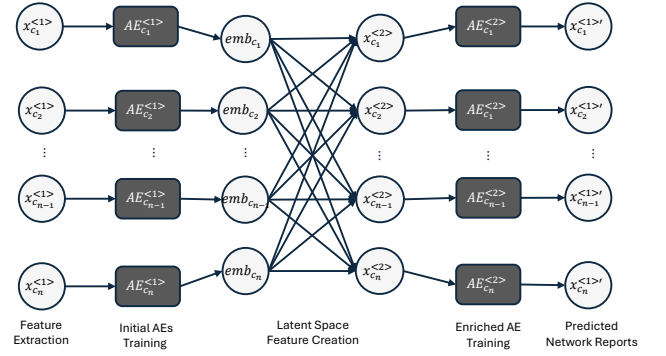


Figure 5: LSTM-autoencoder framework architecture.

The APATE attack poses a significant threat to resource management by targeting the network TS. To mitigate these types of attacks, we propose MARRS (monitoring adversarial RAN reports)—a framework designed to detect adversarial cell telemetry (reported KPIs) in real time. MARRS can be deployed as an xApp on the near-RT RIC, providing immediate notifications about whether a cell's telemetry can be trusted. The detection architecture is based on LSTM networks, which have proven to be particularly effective for anomaly detection in time-series data, especially when combined with AE architectures [34, 58]. LSTM AEs are trained to reconstruct normal input sequences, leveraging an LSTM's ability to capture long-term dependencies in time-series data [70]. We utilize a two-layer combination of LSTM and AEs to capture not only individual cell behavior but the contextual interactions of a specific cell within the entire network.

The detection framework architecture, which is illustrated in Fig. 5, operates as follows: (1) Feature Extraction and Initial AE Training: Relevant time-series features are extracted

from the KPIs reported by cells and UE. Then, for every cell, a dedicated AE model is trained to learn the patterns in these features. (2) Latent Space Feature Creation: Using the latent representations from the first set of AE models, new enriched feature vectors are generated. This is achieved by concatenating the latent space of a specific cell's AE with aggregated latent spaces from other cells, effectively capturing both local and global network contexts. (3) Enriched AE Training: A second AE is trained for each cell, however this time using the enriched feature vectors as input. The objective is to reconstruct the original features from these vectors, leveraging the additional contextual information. (4) Classification of Network Reports: Finally, a classifier is trained to compare the reconstructed output of the second AE with the original input features from the first AE. If the reconstruction loss exceeds a predefined threshold, the input is labeled as untrusted; otherwise, it is labeled as trusted. By integrating contextual insights and multi-stage reconstruction, MARRS serves as a robust mechanism for the detection and mitigation of adversarial activity in the network.

Formally MARRS is described as follows: Given a network denoted as in Section 5, and the network KPI reports R , we aim to identify a framework \mathcal{F} that will reconstruct the original network reports. We examine the loss function score $\ell(\mathcal{F}(R), R)$ between the reconstructed report $\mathcal{F}(R)$ and the original report (R). If it is higher than the threshold \mathcal{T} , the classifier \mathcal{C} will return trusted (i.e., 0), otherwise, untrusted (see Eq. (2)).

$$\mathcal{C}(R, \mathcal{T}) = \begin{cases} 0, & \text{if } \mathcal{T} > \ell(\mathcal{F}(R), R) \\ 1, & \text{if } \mathcal{T} \leq \ell(\mathcal{F}(R), R) \end{cases} \quad (2)$$

6.1 Feature Extraction and Initial AE Training

6.1.1 Feature Extraction

We begin by extracting relevant time-series features from the network, summarized in Table 2. For a given O-RAN network, we extract KPIs such as the physical resource block (PRB) aggregation period, PRB downlink/uplink ratios, and the amount of UE currently in the cell, newly entering UE, and UE leaving the cell. Additionally, we extract UE-specific metrics, including the average and standard deviation of the Packet Data Convergence Protocol (PDCP) downlink and uplink throughput, UE PRB downlink/uplink ratios, and signal quality indicators such as the reference signal received power (RSRP) and signal-to-noise ratio (RSSNIR). In total, these features result in 11 time-series features for each cell. These features were extracted as they are the ones used in the TS flow (defined in the OSC RIC [11]). Finally, all features are standardized to produce scaled values and split to sliding time windows denoted as $X_{c_i}^{<1>}$.

6.1.2 Initial AE Training

After extracting $X_{c_i}^{<1>}$ from the network, we proceeded to train a dedicated AE for each cell, denoted as $AE_{c_i}^{<1>}$. Each AE is based on an LSTM AE architecture and designed to encode every time window $x_{c_i}^{<1>} \in X_{c_i}^{<1>}$ to a new dimensional latent space representation emb_{c_i} and then decode it back to the original $x_{c_i}^{<1>}$. Completing this phase results in a trained LSTM AE for each cell in the network $AE_{c_i}^{<1>}$.

6.2 Latent Space Feature Creation

In this phase, we conduct a second round of feature extraction to generate enriched feature vectors for each cell, capturing both contextual information from the specific cell and the entire network. To achieve this, we leverage the embedded (emb_{c_i}) latent space representations produced by the trained $AE_{c_i}^{<1>}$ from the initial feature set $X_{c_i}^{<1>}$. For each cell v_i^{cl} , we construct a new feature set by concatenating its latent space embedding with an aggregated embedding (e.g., average) derived from the rest of the network.

$$X_{c_i}^{<2>} = emb_{c_i} \cup \frac{1}{n-1} \sum_{j \in V^{cl} \setminus \{v_i^{cl}\}} emb_{c_j} \quad (3)$$

where c_i and c_j represent the features associated with the cell nodes v_i^{cl} and v_j^{cl} respectively and $n = |V^{cl}|$ represents the number of cells in the network. This approach enriches the feature set by capturing both local and network-wide contextual information.

6.3 Enriched AE Training

The next step in our proposed framework involves training a second round of AE models $AE_{c_i}^{<2>}$ for each cell $v_i^{cl} \in V^{cl}$. In this phase, the $AE_{c_i}^{<2>}$ models are designed to encode the enriched feature set $X_{c_i}^{<2>}$ to a new latent space using LSTM layers similar to what was done in $AE_{c_i}^{<1>}$. However, unlike the initial round, these new AEs are trained not to reconstruct their input ($X_{c_i}^{<2>}$) but to decode and reconstruct the first feature set $x_{c_i}^{<1>}$. Through this process the AE models learn to incorporate information from the entire network while effectively leveraging the specific reports from the individual cell, resulting in a more network-context-aware representation.

6.4 Classification of Network Reports

After completing the second round of AE training, the framework is ready to detect malicious activity. To classify and detect this activity, we propagate the cell reports throughout the framework; when cell reports reach the RIC, the first feature set is extracted ($X_{c_i}^{<1>}$) and encoded by its $AE_{c_i}^{<1>}$, producing the latent embedding (emb_{c_i}). This embedding is then used to generate the enriched feature set $X_{c_i}^{<2>}$ as defined

Table 2: Cell and UE Feature $X^{<1>}$

Type	Feature Name	Units	Description
Cell KPIs	Throughput	bps	The amount of data transmitted per unit of time across a cell
	MeasPeriodPrb	kHz	Physical Resource Block (PRB) is defined as a time-frequency resource in the physical layer of wireless communication systems
	Number_UEs	#	Amount of UEs the cell is currently serving
	New_UEs	#	Amount of new UEs in the cell
	Left_UEs	#	Amount of UEs that left the cell
Aggregated UE KPIs	ThpDl_Mean ThpDl_Std	bps	UE's downlink throughput
	Rssnir_Mean Rssnir_Std	dB	RSSNIR (signal to interference & noise ratio) The ratio of the useful signal power to the combined interference and noise power
	Rsrp_Mean Rsrp_Std	dBm	RSRP (Reference Signal Received Power) - The signal strength received by UE from cell

in Eq. (3), which is fed into the second AEs noted as $AE_{c_i}^{<2>}$ to reconstruct the first feature set as $X_{c_i}^{<1>'}$. If the models are trained effectively, the reconstruction loss ℓ between the given feature set extracted from time window report $x_{c_i}^{<1>}$ and the framework output $x_{c_i}^{<1>'}$ is expected to be low for benign reports and high for malicious or compromised reports.

To classify these reports, a threshold \mathcal{T} needs to be defined based on a certain policy provided by the operator. The policy should determine which classification metrics (e.g., recall, precision, F1 score) should be optimized depending on the operator's preferences regarding the network's performance. Finally, reports with reconstruction loss ℓ exceeding \mathcal{T} are classified as untrusted, while those below \mathcal{T} are classified as trusted as in Eq. (2).

6.5 Sequence-Based Detection (S-MARRS)

We propose an extension to MARRS approach which is based on sequence detection denoted as S-MARRS. During inference, given a time window of the same size used in training, we expect the framework to yield a low reconstruction loss for benign time windows and significantly higher losses for malicious time windows. In this approach, both malicious inputs and unrelated outliers that exceed the framework's reconstruction loss threshold are classified as untrusted. This can result in a higher false positive rate (FPR) in the detection method, if a benign sample is classified as untrusted (positive) falsely. To address this issue and reduce the FPR, we apply detection based on sequences of time windows, using specific classification rules defined on the entire sequence. In this detection method, given a trained framework \mathcal{F} , sequence of time windows $S = (R_1, R_2, \dots, R_k)$, classification rule $\mathcal{R}\mathcal{L}$ (e.g. "majority vote"), and threshold \mathcal{T} , we classify the entire sequence as either trusted or untrusted according to the rule. For example, consider a sequence of network time windows' KPI reports $S = (R_1, R_2, \dots, R_k)$ size k , "majority vote" rule, and classifier \mathcal{C} (as in Eq. (2)), we

classify reports (R_1, R_2, \dots, R_k) :

$$\mathcal{CS}(S, \mathcal{T}) = \begin{cases} \text{untrusted,} & \text{if, } \frac{k+1}{2} \leq \sum_{j=1}^k \mathcal{C}(R_j, \mathcal{T}) \\ \text{trusted,} & \text{else} \end{cases} \quad (4)$$

In this example, if most of the sequence exceeds \mathcal{T} , the entire sequence is classified as untrusted. This approach allows us to reduce the false positives that may arise due to outliers and focus more on malicious behavior.

7 Test Environment

We developed a simulation testbed environment that contains two primary components, as illustrated in Fig. 6; (1) Wireless Network Simulator [17]: Which simulates real-time network scenarios involving UE and gNB cells. (2) O-RAN Software Community (OSC) Near-RT RIC Platform [11]: Deployed within a Kubernetes cluster. This platform serves as a dynamic hosting environment for the relevant xApps in the TS flow, including the AD, QP, and TS xApps.

In deployment, these two components are isolated from each other to simulate real-world usage and communicate within a closed-loop system via a REST API. The interaction occurs as follows: At the end of each simulation iteration, the simulator reports the current KPIs (both UE and cell KPIs) to the RIC cluster. Simultaneously, the TS xApp in the RIC platform generates handover requests based on those KPIs and sends them back to the simulator for UE allocation.

7.1 Wireless Network Simulator

The simulator is deployed separately from the near-RT-RIC cluster on an AWS EC2 instance running Ubuntu 20.04 LTS (Focal Fossa). The simulation begins with the configuration of network parameters, including the geographical topology size, initial locations and velocities of UE, and cell types (e.g., gNB, eNB) along with their respective locations. The

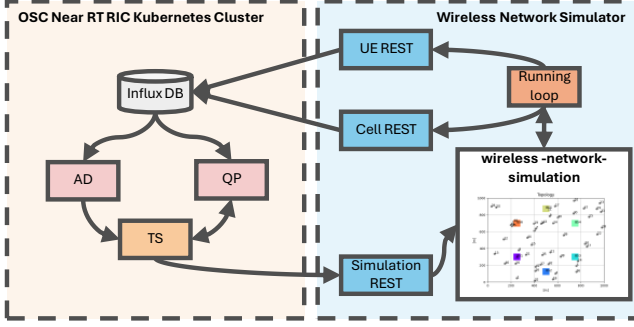


Figure 6: Testbed Environment; left - OSC near-RT RIC Kubernetes cluster and right - the wireless network simulator.

simulation operates with two parallel threads, one of which manages the core simulation loop, while the other hosts a Flask app that listens for handover requests from the RIC. Upon receiving a request, the Flask app processes it and updates the target cell allocation for the relevant UE. In each simulation iteration, UE either moves randomly or following predefined trajectories, and the simulator generates updated metrics for both UE and cells, formatted to align with the RIC requirements. Each report has the following KPIs: UE ID, serving cell ID, location, timestamp, PDCP aggregation period, PDCP throughput, PRB report timestamp, PRB aggregation period, PRB throughput ratios, reference signal received power (RSRP), reference signal received quality (RSRQ), and signal-to-noise ratio (SNIR). Cell KPIs include the cell ID, timestamp, PDCP aggregation period, PDCP throughput, PRB aggregation period, and PRB throughput. In this way, the TS handover requests are reflected in real time in the simulation, allowing live UE allocation.

7.2 OSC RIC Cluster

The RIC platform, deployed as a Kubernetes cluster on an AWS EC2 instance running Ubuntu 20.04 LTS (Focal Fossa), hosts the TS flow xApps (AD, QP, and TS) as individual pods. It receives KPIs from the simulator, which functions as the RAN. Upon receiving these KPIs, the KPIMON xApp processes the metrics and writes them to the RIC’s InfluxDB pod. Once the data is populated in the RIC database, the TS flow begins as described in Section 2.3, generating a handover request. This request is sent back to the simulator, completing one iteration of the closed-loop testbed. The process repeats continuously until the simulation loop is done.

8 Evaluation

8.1 Experimental Setting

8.1.1 Attack

Data Collection To demonstrate and evaluate the impact of the APATE attack, we set up a network topology with six gNB cells and 50 UEs randomly moving within this topology as shown in Fig. 8a. At the end of each iteration of the simulation loop, the simulator reports the state of the network to the RIC cluster as described in Section 7. Then, the KPIMON xApp populates the RIC DB with the reported KPIs, and the TS flow begins. Once the TS makes a handover decision (as described in Section 4) for the UE, the TS sends it back to the simulator. The simulator receives the TS handover request and updates the environment based on its decision.

Attack Scenarios We evaluate APATE in two attack scenarios using the described closed-loop simulation: (1) Single-Attack Scenario (SAS): A single malicious cell (BS5) executes the APATE attack, manipulating its KPI reports to mislead the TS into allocating it more UEs. (2) Multi-Attack Scenario (MAS): Two malicious cells (BS1 and BS5) simultaneously execute the APATE attack, manipulating their KPI reports to mislead the TS into allocating them more UEs.

For comparison, we establish corresponding benign baseline scenarios where all cells report trusted KPI telemetry to the RIC. To accurately model real-world attack progression, we initialize both attack scenarios using identical conditions to their benign baseline scenarios, while the benign scenarios initialized randomly. The velocity steps are consistent across all scenarios.

Adversarial Sample Generation To execute the APATE, we employed the HopSkipJump attack [16] from the Adversarial Robustness Toolbox (ART) [47] to generate adversarial samples. This involved categorizing the QP outputs into four quality levels: poor, average, good, and excellent, with the goal of manipulating the QP to forecast a higher quality category to the attacker cell than the true one.

8.1.2 Detection

Data Collection To evaluate MARRS’s detection capabilities, we train the framework as described in Section 6. The training process begins with data collection, conducted through closed-loop benign simulation scenarios. In these scenarios, we initialize a network topology consisting of six gNB cells and 50 UEs, randomly moving within the topology as illustrated in Fig. 8a. From these simulations, we extract relevant features, as detailed in Table 2 resulting overall dataset for a training size of 10531 records.

Model Training The feature set, denoted as $X^{<1>}$, serves as the input for training the first layer of $AE^{<1>}$ s in the framework. The $AE^{<1>}$ architecture consists of an LSTM encoder

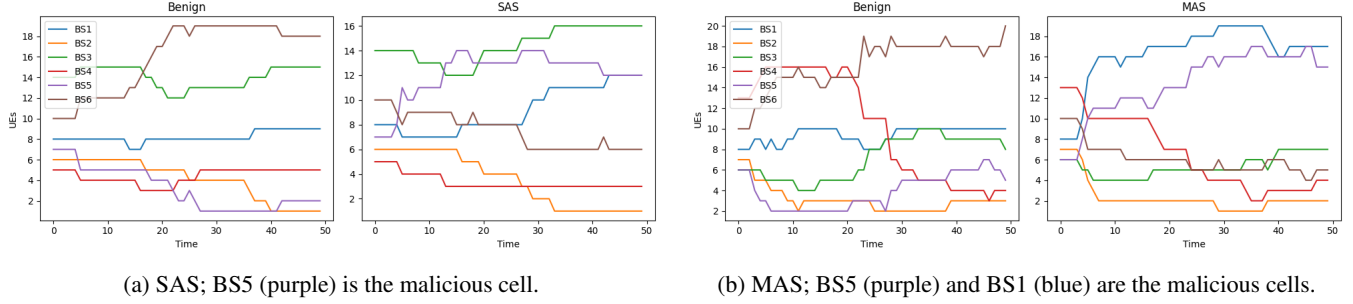


Figure 7: Amount of UE for each cell (BS1-BS6) during each of the network iterations (x-axis) in the examined scenarios: the SAS, the MAS, and the corresponding benign scenario.

Table 3: Amount of UE for each cell in each scenario - corresponding to Fig. 7.

	SAS							MAS						
	Benign			Malicious			Difference	Benign			Malicious			Difference
cell ID	mean	min	max	mean	min	max	%	mean	min	max	mean	min	max	%
BS1	8.14	7	9	8.81	7	12	108.25%	8.97	8	10	13.83	8	19	154.16%
BS2	4.64	1	6	4.21	1	6	90.77%	4.27	2	7	3.73	1	7	87.46%
BS3	14.01	12	15	14.31	12	16	102.14%	6.82	4	10	5.55	4	7	81.40%
BS4	4.59	3	5	3.83	3	5	83.49%	11.04	3	16	8.45	2	13	76.53%
BS5	4.27	1	7	10.61	7	14	248.50%	4.56	2	7	11.21	6	17	245.68%
BS6	14.34	10	19	8.21	6	10	57.27%	14.34	10	20	7.23	4	10	50.39%

followed by an LSTM decoder with a fully connected (FC) output layer. The models are implemented in PyTorch [55], using the Adam optimizer and mean squared error (MSE) as the loss function. After training the $AE^{<1>}$ s, we extract the second feature set ($X^{<2>}$), according to Eq. (3), which is subsequently used to train the next layer of AEs ($AE^{<2>}$). The $AE^{<2>}$ s use the same architecture, optimizer, and loss function as the $AE^{<1>}$ s. Both AE layers are trained for 200 epochs and the hyperparameters such as the number of LSTM layers hidden size, and learning rate are tuned using *Optuna* [3]. In these experiments, we set the threshold policy \mathcal{T} to maximize the F1 score in the classification processes.

Compared Benchmarks All compared benchmarks below were trained using the same dataset and evaluated on the same test set. (1) Isolation Forest (IF), an anomaly detection algorithm that isolates outliers by recursively partitioning data and scoring it based on the number of splits required to isolate an observation [36]. (2) One-Class SVM (OCSVM), which learns a decision boundary to separate benign data from outliers, treating all training data as belonging to one class [35]. (3) Autoencoder (AE), which learns to compress and reconstruct data using linear layers, with anomalies detected by measuring reconstruction error, assuming that benign data have lower errors than the anomalous data.

8.2 Experimental Results

8.2.1 Attack

The experimental results of the APATE attack are presented in Fig. 7 and summarized in Table 3. Fig. 7 presents the network state for both attack scenarios and the corresponding benign scenarios detailed in Section 8.1: (1) SAS with BS5 as the malicious cell (Fig. 7a), and (2) the MAS with both BS1 and BS5 as malicious cells (Fig. 7b). Each cell (BS1-BS6) is presented in a different color, with the y-axis representing the amount of connected UE during each iteration. Table 3 summarizes the two scenarios for each cell. Each row represents a cell, while the columns represent the different scenarios; for each scenario, presented the average amount of UE served, the percentage difference from the benign scenario, and minimum and maximum UE counts.

When examining the results regarding the SAS, we see a significant increase of 248.5% in the average amount of UE served by the malicious cell BS5 in the malicious scenario compared to the benign scenario. Additionally, we observe a higher minimum number of serving UE for BS5 in the malicious scenario, indicating that fewer UE left compared to the benign scenario. In the MAS, we see the attack's impact across the entire network. The malicious cells BS1 and BS5 increased the amount of their served UE by 154.16% and 248.5% respectively, while their neighbor cell BS6 suffered a 50.39% reduction in its average amount of served UE.

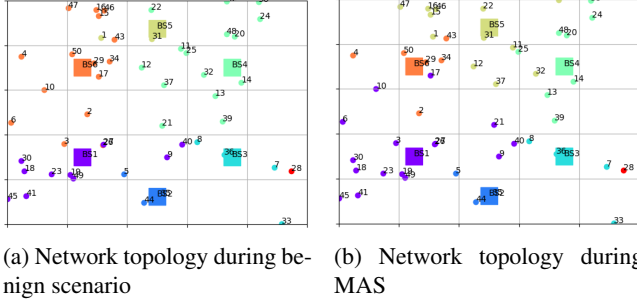


Figure 8: Network topology during the benign scenario (Fig. 8a) and MAS (Fig. 8b). The boxes (BS1 - BS6) are the cells, and the circles (1-50) are the UE IDs. The colors represent the UE’s association to a cell. Both Fig. 8a and Fig. 8b provide a snapshot of the same simulation iteration.

Fig. 7 demonstrates the impact of the attack by comparing UE distribution patterns across cells over time. In the SAS shown in Fig. 7a, where malicious cell BS5 (purple) executes the attack, we observe different behavior between the benign and malicious scenarios. While the benign scenario shows BS5’s UE count decreasing over time, the malicious scenario shows a significant increase in its UE allocations. The MAS shown in Fig. 7b, where both BS1 (blue) and BS5 (purple) execute the attack, shows several distinct patterns. In the benign scenario, both BS1 and BS5 maintain relatively stable UE counts. However, during the attack scenario, both malicious cells demonstrate increases in their UE allocations. Notably, this attack significantly impacts neighboring cell BS6 (brown), which experiences a substantial reduction in UE connections compared to its high allocation in the benign scenario.

Fig. 8 presents the network states for two scenarios: the benign scenario (Fig. 8a) and the MAS (Fig. 8b). In both figures, circles represent UE, and boxes represent cells, with each UE’s color indicating its serving cell. Both Fig. 8a and Fig. 8b provide snapshots for a specific simulation iteration. The figures illustrate the impact of the attack, which affects not only the attacker’s cell but also neighboring cells, particularly BS6 (orange), by reducing the amount of UE it serves.

8.2.2 Detection

To evaluate MARRS method, we first demonstrate the importance of gathering data over time in the simulation testbed. The results are presented in Table 4, where each row presents different subsets of the training set that were used (x_1, x_2, x_3, x_4), and the columns present the accuracy, precision, recall, and F1 scores. As can be see in the table, the more time the system operates, the more data it collects; accordingly, the classification metrics results improve.

Compared Benchmarks We compared MARRS’s performance to that of other detection methods as detailed in Sec-

Table 4: Accuracy of the MARRS method over time.

Training Set	Accuracy	Precision	Recall	F1 Score
x_1	0.873	0.793	1	0.884
x_1, x_2	0.984	0.986	0.922	0.949
x_1, x_2, x_3	0.964	0.932	1	0.965
x_1, x_2, x_3, x_4	0.992	0.958	1	0.978

tion 8.1. The results are presented in Table 5, where each row represents a method used to detect the APATE attack, and the columns contain the classification metric values on the test set. As can be seen, MARRS outperforms all other methods on the F1 score and accuracy metrics.

Table 5: Performance of the examined detection methods.

	Method	Accuracy	Precision	Recall	F1
Benchmarks	IF	0.837	0.522	1	0.69
	OCSVM	0.871	0.578	0.985	0.730
	LAE	0.873	0.793	1	0.884
	MARRS	0.992	0.958	1	0.978

Ablation Study In ablation studies, components of an ML model are systematically removed or altered to assess their impact on the model’s overall performance. The goal is to determine how each part contributes to the overall effectiveness of the model [43, 45]. We employed this evaluation process to examine the effectiveness of MARRS’s architecture (see Fig. 5). In the first experiment of the ablation study just the first layer of AEs ($AE^{<1>}$) was used to encode and reconstruct network KPIs and classify them based on their reconstruction loss. In the second experiment, we trained the AE for each cell using the concatenated average features from the rest of the network along with the cell’s own feature set ($AE^{<1+>}$). This is in contrast to our detection method which also employs two additional steps: latent space feature creation and enriched AE training (described in Section 6).

The results are summarized in Table 6, where each row represents the experiment ($AE^{<1>}$ and $AE^{<1+>}$), and the columns contain the classification metric values. The results show how incorporating latent space features as contextual information from the entire network improves the detection of malicious cell reports.

Sequence-Based Detection (S-MARRS) Fig. 9 presents the results of our sequence-based detection approach (see Section 6.5), which reduces the false positives that may result from outliers and focuses on malicious behavior by examining a sequence of KPI time windows instead of an individual time window. The classification rules used in this experiment are as follows: A (all rule): The sequence is classified

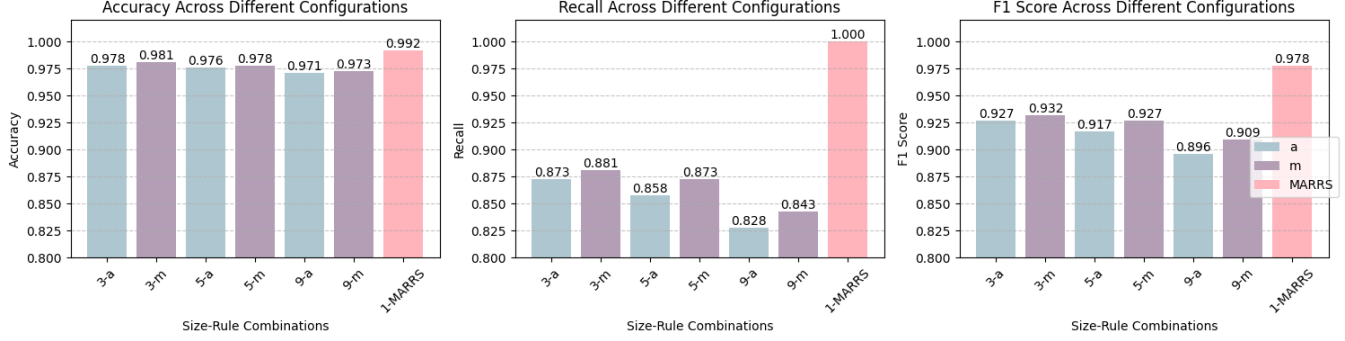


Figure 9: Sequence-based detection approach results.

Table 6: Ablation study results.

Layer	Accuracy	Precision	Recall	F1 Score
$AE^{<1>}$	0.917	0.932	1	0.964
$AE^{<1+>}$	0.966	0.934	1	0.965
MARRS	0.992	0.958	1	0.978

as untrusted if all windows within the sequence exceed the threshold \mathcal{T} ; otherwise, it is classified as trusted. M (majority rule): The sequence is classified as untrusted if the majority of windows within the sequence exceed the threshold \mathcal{T} ; otherwise, it is classified as trusted.

Note that on the x-axes in Fig. 9, the number before the letters "A" or "M" indicates the sequence size. The evaluation results for accuracy, recall, and F1 score are summarized in Fig. 9. In terms of precision, all configurations of "A" and "M" obtained a perfect score of 1, however when a sequence-based configuration was not used, which we refer to as the 1-MARRS (see Fig. 9 the last column) detection approach, precision of 0.958 was obtained. The primary goal of adopting a sequence-based detection approach is to minimize the FPR, and the results demonstrate that we have achieved this objective. The FPR directly impacts precision and indirectly influences metrics like recall and the F1 score. In configurations where precision is a perfect 1, this indicates the absence of false positives, i.e., benign time windows were not misclassified as untrusted. On the other hand, the non-sequence detection approach achieved a perfect recall of 1 but at the expense of a higher FPR, which reduced its precision compared to the sequence-based configurations. This highlights the tradeoff between high recall and precision when the FPR is not adequately controlled.

9 Discussion

Throughout this paper, we have examined the security vulnerabilities arising from multi-operator deployments. We

have demonstrated the need for robust security solutions and regulatory constraints, compliance, and auditing in this rapidly evolving domain. According to the introduced threat model presented in Section 4, we demonstrated how malicious cells can exploit the multi-operator network to manipulate the TS flow and unfairly increase their UE allocations (the APATE attack). The risks that arise from applying the APATE attack, not only undermine the integrity of the network but also degrades the QoE for UEs, presenting new security challenges for the O-RAN architecture.

However, the presented threat model (Section 4) is not limited to scenarios that only involve malicious operators as threat actors. A similar threat can emerge even in single-operator networks if a cell's supply chain is compromised. An attacker who gains control of a cell via a supply chain breach could execute the APATE attack, resulting in disruptions similar to those detailed in Section 8 such as QoE reduction, which could harm the operator's reputation. Deploying MARRS on the near-RT RIC offers a solution to these challenges as well, as it treats all telemetry as untrusted, enabling the detection and mitigation of such threats.

10 Conclusion and Future Work

This paper addresses O-RAN vulnerabilities in multi-operator environments. We introduce the APATE to demonstrate how a malicious operator can exploit these vulnerabilities by executing an evasion attack that misleads O-RAN traffic steering and disrupts network load distribution. To counter such threats and ensure the continuation of O-RAN's legitimate operation, we developed the MARRS framework, designed to detect compromised telemetry in real time. Our evaluation reveals MARRS achieves high precision, recall, and F1 scores in detecting malicious cell behavior.

Future work can focus on enhancing the MARRS framework by developing an innovative approach for automated policy selection using deep reinforcement learning (DRL). Currently, the policies (Section 6.4) in MARRS must be

manually managed by the operators, a process which may be prone to human error. However, leveraging the RIC architecture, it is possible to train a DRL-based policy-making model within the service management and orchestration (SMO) layer. This model would be able to be deployed as rApp on the non-RT RIC and integrated with the MARRS xApp hosted on the near-RT RIC via the A1 interface. This automated approach could significantly improve the proposed MARRS’s adaptability and efficiency while reducing human involvement.

References

- [1] 3rd Generation Partnership Project. TR 22.852: Study on RAN Sharing Enhancements (Release 12 & 13). Technical report, Technical Specification Group Radio Access Networks, 2013.
- [2] Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, Jude Okwuibe, Mika Ylianttila, and Andrei Gurtov. Overview of 5g security challenges and solutions. *IEEE Communications Standards Magazine*, 2(1):36–43, 2018.
- [3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.
- [4] Hamed Alimohammadi, Sotiris Chatzimiltis, Samara Mayhoub, Mohammad Shojafar, Seyed Ahmad Soleymani, Ayhan Akbas, and Chuan Heng Foh. Kpi poisoning: An attack in open ran near real-time control loop.
- [5] O-RAN ALLIANCE. WG11: O-RAN Work Group 11 (Security Work Group) Security Requirements and Controls Specifications, 2024.
- [6] Wilfrid Azariah, Fransiscus Asisi Bimo, Chih-Wei Lin, Ray-Guang Cheng, Navid Nikaein, and Rittwik Jana. A survey on open radio access networks: Challenges, research directions, and open source approaches. *Sensors*, 24(3):1038, 2024.
- [7] Pau Baguer, Girma M Yilma, Esteban Municio, Gines Garcia-Aviles, Andres Garcia-Saavedra, Marco Liebisch, and Xavier Costa-Pérez. Attacking o-ran interfaces: Threat modeling, analysis and practical experimentation. *IEEE Open Journal of the Communications Society*, 2024.
- [8] Ravikumar Balakrishnan, Marius Arvinte, Nageen Himayat, Hosein Nikopour, and Hassnaa Moustafa. Enhancing o-ran security: Evasion attacks and robust defenses for graph reinforcement learning-based connection management. *arXiv preprint arXiv:2405.03891*, 2024.
- [9] Bharath Balasubramanian, E Scott Daniels, Matti Hiltunen, Rittwik Jana, Kaustubh Joshi, Rajarajan Sivaraj, Tuyen X Tran, and Chengwei Wang. Ric: A ran intelligent controller platform for ai-enabled cellular networks. *IEEE Internet Computing*, 25(2):7–17, 2021.
- [10] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrđić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23–27, 2013, Proceedings, Part III 13*, pages 387–402. Springer, 2013.
- [11] Fransiscus Asisi Bimo, Ferlinda Feliana, Shu-Hua Liao, Chih-Wei Lin, David F Kinsey, James Li, Rittwik Jana, Richard Wright, and Ray-Guang Cheng. Osc community lab: The integration test bed for o-ran software community. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 513–518. IEEE, 2022.
- [12] Ron Bitton, Nadav Maman, Inderjeet Singh, Satoru Momiyama, Yuval Elovici, and Asaf Shabtai. Evaluating the cybersecurity risk of real-world, machine learning production systems. *ACM Computing Surveys*, 55(9):1–36, 2023.
- [13] Leonardo Bonati, Michele Polese, Salvatore D’Oro, Stefano Basagni, and Tommaso Melodia. Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead. *Computer Networks*, 182:107516, 2020.
- [14] Paulo Ricardo Branco da Silva, João Paulo Henriques Sales de Lima, Erika Costa Alves, William Sanchez Farfan, Victor Aguiar Coutinho, Thomas William do Prado Paiva, Daniel Lazkani Feferman, and Francisco Hugo Costa Neto. Evaluation of the latency of machine learning random access ddos detection in open ran. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 2306–2311, 2024.
- [15] 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. IEEE, 2023.
- [16] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient

- decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294. IEEE, 2020.
- [17] Emanuele De Santis, Alessandro Giuseppe, Antonio Pietrabissa, Michael Capponi, and Francesco Delli Priscoli. Satellite integration into 5g: deep reinforcement learning for network selection. *Machine Intelligence Research*, 19(2):127–137, 2022.
- [18] Panagiotis Demestichas, Andreas Georgakopoulos, Dimitrios Karvounas, Kostas Tsagkaris, Vera Stavroulaki, Jianmin Lu, Chunshan Xiong, and Jing Yao. 5g on the horizon: Key challenges for the radio-access network. *IEEE vehicular technology magazine*, 8(3):47–53, 2013.
- [19] Marcin Dryjański, Łukasz Kułacz, and Adrian Kliks. Toward modular and flexible open ran implementations in 6g networks: Traffic steering use case and o-ran xapps. *Sensors*, 21(24):8173, 2021.
- [20] Keith Dyer. Freshwave says four way operator sharing on same indoor small cells a world first, 2024.
- [21] ESA. Spacetime and O-RAN Interfaces 5G/6G NTN, 2024. <https://connectivity.esa.int/projects/spacetime-and-oran-interfaces-5g6g-ntns>.
- [22] Soha Farhat, Zahraa Chahine, Abed Ellatif Samhat, Samer Lahoud, and Bernard Cousin. Access selection and joint pricing in multi-operator wireless networks: A stackelberg game. In *2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, pages 38–43. IEEE, 2015.
- [23] Soha Farhat, Abed Ellatif Samhat, Samer Lahoud, and Bernard Cousin. Best operator policy in a heterogeneous wireless network. In *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*, pages 53–57. IEEE, 2014.
- [24] Soha Farhat, Abed Ellatif Samhat, Samer Lahoud, and Bernard Cousin. Radio access network sharing in 5g: strategies and benefits. *Wireless Personal Communications*, 96:2715–2740, 2017.
- [25] Ian J Goodfellow. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [26] Joshua Groen, Salvatore D’Oro, Utku Demir, Leonardo Bonati, Michele Polese, Tommaso Melodia, and Kaushik Chowdhury. Implementing and evaluating security in o-ran: Interfaces, intelligence, and platforms. *IEEE Network*, 2024.
- [27] Vodafone Group. Vodafone announces expanded network sharing agreement with orange in spain, 2019.
- [28] GSMA. Mobile infrastructure sharing.
- [29] Edan Habler, Ron Bitton, Dan Avraham, Dudu Mimiran, Eitan Klevansky, Oleg Brodt, Heiko Lehmann, Yuval Elovici, and Asaf Shabtai. Adversarial machine learning threat analysis and remediation in open radio access network (o-ran). *arXiv preprint arXiv:2201.06093*, 2022.
- [30] Mahmoud A Hasabelnaby, Mohanad Obeed, Mohammed Saif, Anas Chaaban, and MJ Hossain. From centralized ran to open ran: A survey on the evolution of distributed antenna systems. *arXiv preprint arXiv:2411.12166*, 2024.
- [31] Cheng-Feng Hung, You-Run Chen, CHI-Heng Tseng, and Shin-Ming Cheng. Security threats to xapps access control and e2 interface in o-ran. *IEEE Open Journal of the Communications Society*, 2024.
- [32] Eugina Jordan. Ric: The next phase of open ran, 2024.
- [33] Shin Yuan Kee. How does open ran add value in multi-operator sharing?, 2021.
- [34] Younjeong Lee, Chanhoo Park, Namji Kim, Jisu Ahn, and Jongpil Jeong. Lstm-autoencoder based anomaly detection using vibration data of wind turbines. *Sensors*, 24(9):2833, 2024.
- [35] Kun-Lun Li, Hou-Kuan Huang, Sheng-Feng Tian, and Wei Xu. Improving one-class svm for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, volume 5, pages 3077–3081. IEEE, 2003.
- [36] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth IEEE international conference on data mining*, pages 413–422. IEEE, 2008.
- [37] Lixin Liu, Yuanjie Li, Hewu Li, Jiabo Yang, Wei Liu, Jingyi Lan, Yufeng Wang, Jiarui Li, Jianping Wu, Qian Wu, et al. Democratizing {Direct-to-Cell} low earth orbit satellite networks. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 791–808, 2024.
- [38] Madhusanka Liyanage, An Braeken, Shahriar Shahabuddin, and Pasika Ranaweera. Open ran security: Challenges and opportunities. *Journal of Network and Computer Applications*, 214:103621, 2023.
- [39] Johannes Maisack. Deutsche telekom and telefónica share network infrastructure to enhance network coverage, 2021.

- [40] Simona Marinova and Alberto Leon-Garcia. Intelligent o-ran beyond 5g: Architecture, use cases, challenges, and opportunities. *IEEE Access*, 12:27088–27114, 2024.
- [41] Jan Markendahl and Amirhossein Ghanbari. Shared smallcell networks multi-operator or third party solutions-or both? In *2013 11th International symposium and workshops on modeling and optimization in mobile, Ad Hoc and wireless networks (WiOpt)*, pages 41–48. IEEE, 2013.
- [42] Jan Markendahl, Amirhossein Ghanbari, and Bengt G Mölleryd. Network cooperation between mobile operators-why and how competitors cooperate? In *IMP conf, Atlanta*, 2013.
- [43] Richard Meyes, Melanie Lu, Constantin Waubert de Puiseau, and Tobias Meisen. Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*, 2019.
- [44] Dudu Mimran, Ron Bitton, Yehonatan Kfir, Eitan Klevansky, Oleg Brodt, Heiko Lehmann, Yuval Elovici, and Asaf Shabtai. Evaluating the security of open radio access networks. *arXiv preprint arXiv:2201.06080*, 2022.
- [45] Alessio Molinari. Designing a performant ablation study framework for pytorch, 2020.
- [46] NEC Corporation. Ran sharing: Nec’s approach towards active radio access network sharing. Techreport, NEC Corporation, 2013.
- [47] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*, 2018.
- [48] Solmaz Niknam, Abhishek Roy, Harpreet S Dhillon, Sukhdeep Singh, Rahul Banerji, Jeffery H Reed, Navrati Saxena, and Seungil Yoon. Intelligent o-ran for beyond 5g and 6g wireless networks. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 215–220. IEEE, 2022.
- [49] O-RAN ALLIANCE. WG2: Non-real-time RAN Intelligent Controller and A1 Interface Workgroup, 2022.
- [50] O-RAN ALLIANCE. WG3: Near-real-time RIC and E2 Interface Workgroup, 2022.
- [51] Johnson Opadere, Qiang Liu, Tao Han, and Nirwan Ansari. Energy-efficient virtual radio access networks for multi-operators cooperative cellular networks. *IEEE Transactions on Green Communications and Networking*, 3(3):603–614, 2019.
- [52] Oner Orhan, Vasuki Narasimha Swamy, Thomas Tetzlaff, Marcel Nassar, Hosein Nikopour, and Shilpa Talwar. Connection management xapp for o-ran ric: A graph neural network and reinforcement learning approach. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 936–941. IEEE, 2021.
- [53] Heejae Park, Tri-Hai Nguyen, and Laihyuk Park. An investigation on open-ran specifications: Use cases, security threats, requirements, discussions. *CMES-Computer Modeling in Engineering & Sciences*, 141(1), 2024.
- [54] Imtiaz Parvez, Ali Rahmati, Ismail Guvenc, Arif I Sarwat, and Huaiyu Dai. A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, 20(4):3098–3130, 2018.
- [55] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [56] Michele Polese, Leonardo Bonati, Salvatore D’oro, Stefano Basagni, and Tommaso Melodia. Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys & Tutorials*, 25(2):1376–1411, 2023.
- [57] Francesco Delli Priscoli, Alessandro Giuseppi, Francesco Liberati, and Antonio Pietrabissa. Traffic steering and network selection in 5g networks based on reinforcement learning. In *2020 European Control Conference (ECC)*, pages 595–601. IEEE, 2020.
- [58] Alaa Sagheer and Mostafa Kotb. Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. *Scientific reports*, 9(1):19038, 2019.
- [59] Nisar Sanadi. How will ric leverage ai/ml to improve user experience?, December 2023.
- [60] Naveen Naik Sapavath, Brian Kim, Kaushik Chowdhury, and Vijay K Shah. Experimental study of adversarial attacks on ml-based xapps in o-ran. *arXiv preprint arXiv:2309.03844*, 2023.
- [61] Chih-Ting Shen, Yu-Yi Xiao, Yi-Wei Ma, Jiann-Liang Chen, Cheng-Mou Chiang, Shiang-Jiun Chen, and Yu-Chuan Pan. Security threat analysis and treatment strategy for oran. In *2022 24th International Conference on*

- Advanced Communication Technology (ICACT)*, pages 417–422. IEEE, 2022.
- [62] Seyed Ahmad Soleymani, Mohsen Eslamnejad, Hamed Alimohammadi, Ayhan Akbas, Chuan Heng Foh, and Mohammad Shojafar. Ddos detection and mitigation using d/xapp in o-ran.
 - [63] Sanaz Soltani, Mohammad Shojafar, Ali Amanlou, and Rahim Tafazolli. Intelligent control in 6g open ran: Security risk or opportunity? *arXiv preprint arXiv:2405.08577*, 2024.
 - [64] Otto T. Neutral host: how open ran and neutral host paves the way for 5g, 2021.
 - [65] Elham Tabassi, Kevin J Burns, Michael Hadjimichael, Andres D Molina-Markham, and Julian T Sexton. A taxonomy and terminology of adversarial machine learning. *NIST IR*, 2019:1–29, 2019.
 - [66] Muhammad Tayyab, Xavier Gelabert, and Riku Jäntti. A survey on handover management: From lte to nr. *IEEE Access*, 7:118907–118930, 2019.
 - [67] Tower Genius LLC. Cell tower co-location. <https://www.cell-phone-towers.com/Cell-Tower-Colocation.html>, 2018. Accessed: Jul. 30, 2018.
 - [68] Theodoros Tsourdinis, Nikos Makris, Thanasis Korakis, and Serge Fdida. Ai-driven network intrusion detection and resource allocation in real-world o-ran 5g networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 1842–1849, 2024.
 - [69] Vodafone. Technology and innovation open ran, 2024.
 - [70] Yuanyuan Wei, Julian Jang-Jaccard, Wen Xu, Fariza Sabrina, Seyit Camtepe, and Mikael Boulic. Lstm-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sensors Journal*, 23(4):3787–3800, 2023.
 - [71] Andrew Wooden. Freshwave pumps out 4g from all four uk operators in one unit, 2024.
 - [72] Bruno Missi Xavier, Merim Dzaferagic, Diarmuid Collins, Giovanni Comarella, Magnos Martinello, and Marco Ruffini. Machine learning-based early attack detection using open ran intelligent controller. *arXiv preprint arXiv:2302.01864*, 2023.
 - [73] Bruno Missi Xavier, Merim Dzaferagic, Irene Vilà, Magnos Martinello, and Marco Ruffini. Cross-domain ai for early attack detection and defense against malicious flows in o-ran. *arXiv preprint arXiv:2401.09204*, 2024.
 - [74] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33:21024–21037, 2020.
 - [75] Xiangming Zhu and Chunxiao Jiang. Integrated satellite-terrestrial networks toward 6g: Architectures, applications, and challenges. *IEEE Internet of Things Journal*, 9(1):437–461, 2021.