# Misconfiguration in O-RAN: Analysis of the impact of AI/ML

Noe M. Yungaicela-Naula[a,*], Vishal Sharma,[a] and Sandra Scott-Hayward[a]

[a]Centre for Secure Information Technologies (CSIT), Queen's University Belfast, Belfast, BT3 9DT, Northern Ireland, UK

ARTICLE INFO

*Keywords*:
Open RAN
O-RAN
ML
5G
6G
Security
xApp
Misconfiguration

ABSTRACT

User demand on network communication infrastructure has never been greater with applications such as extended reality, holographic telepresence, and wireless brain-computer interfaces challenging current networking capabilities. Open RAN (O-RAN) is critical to supporting new and anticipated uses of 6G and beyond. It promotes openness and standardisation, increased flexibility through the disaggregation of Radio Access Network (RAN) components, supports programmability, flexibility, and scalability with technologies such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), and cloud, and brings automation through the RAN Intelligent Controller (RIC). Furthermore, the use of xApps, rApps, and Artificial Intelligence/Machine Learning (AI/ML) within the RIC enables efficient management of complex RAN operations. However, due to the open nature of O-RAN and its support for heterogeneous systems, the possibility of misconfiguration problems becomes critical. In this paper, we present a thorough analysis of the potential misconfiguration issues in O-RAN with respect to integration and operation, the use of SDN and NFV, and, specifically, the use of AI/ML. The opportunity for AI/ML to be used to identify these misconfigurations is investigated. A case study is presented to illustrate the direct impact on the end user of conflicting policies amongst xApps along with a potential AI/ML-based solution to this problem. This research presents a first analysis of the impact of AI/ML on misconfiguration challenges in O-RAN.

arXiv:2403.01180v2 [cs.NI] 26 Apr 2024

## 1. Introduction

As 5G evolves, the transition to 6G, which is expected beyond 2030 [1], attempts to reinvent human engagement with digital spaces. Extended reality, networked robots, wireless brain-computer interfaces, holographic telepresence, and e-health with body area networks are among the anticipated uses of 6G [2]. These applications necessitate support for new capabilities for Enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC) [3, 4]. To achieve these goals, major reshaping of existing 5G and 6G architectures is necessary, with a focus on offering flexibility, configurability, and automation.

The Radio Access Network (RAN), a critical and costly component in wireless networks, is part of the innovation in 5G and 6G. This component, which may be considered the most complex part of cellular networks, is undergoing transition through technologies such as Open RAN (O-RAN)[1]. O-RAN has a disaggregated, virtualized, and software-based strategy, linking components via open interfaces and enabling interoperability among vendors [5]. Furthermore, the Artificial Intelligence/Machine Learning (AI/ML) integration in O-RAN enables intelligent management of RAN resources, addresses optimisation challenges, and elevates the user experience [6]. Particularly, O-RAN introduces the RAN Intelligent Controller (RIC), which houses third-party applications (rApps and xApps) powered by AI/ML that streamline RAN operations and manage complexity [7, 8].

As a result, unlike previous RAN technologies, O-RAN has the potential to provide programmability, optimisation, and end-to-end automation in 5G and 6G. However, realising this potential is dependent on the correct configuration and operation of O-RAN components. Neglecting these factors may result in a variety of misconfiguration difficulties.

Misconfiguration is defined by the National Institute of Standards and Technology (NIST) as *an incorrect or suboptimal configuration of an information system or system component that may lead to vulnerabilities* [9]. In this respect, misconfiguration allows or induces unintended behaviour, hence impacting a system's security posture [10]. Based on these criteria, it could be argued that misconfiguration has a direct and indirect impact. The direct impact is a decrease in system performance, while the indirect impact is an increased vulnerability to security attacks.

Misconfigurations are more prevalent for the 3rd Generation Partnership Project (3GPP) Next-Generation RAN (NG-RAN) than for previous generations such as Universal Terrestrial RAN (UTRAN) and Evolved UTRAN (E-UTRAN). This increased risk is associated with the introduction of new technologies such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), cloud computing, and AI/ML in NG-RAN. When combined with the disaggregation and openness envisioned for O-RAN, as well as the introduction of third-party applications into the RIC, these technologies augment the system's complexity and raise the possibility of misconfiguration. Even minor mistakes in setting up protocols, interfaces, APIs, authentication, and authorization systems might result in new vulnerabilities and security breaches [11].

AI/ML emerges as a possible approach for managing the O-RAN's configuration challenges. It provides automation features for both high-level orchestration and low-level resource optimisation. Nonetheless, the incorporation of AI/ML with O-RAN presents the possibility of misconfigurations. In this context, a thorough examination of both

*Corresponding author
n.yungaicela@qub.ac.uk (N.M. Yungaicela-Naula)
ORCID(s): 0000-0002-3131-0672 (N.M. Yungaicela-Naula)

[1]The study is based on O-RAN, which refers to the Open RAN architecture defined by the O-RAN ALLIANCE.

of these aspects is required in order to comprehend all misconfiguration challenges and engage in discussions about the essential solutions to be adopted.

This paper analyzes misconfiguration concerns in O-RAN, examines the use of AI/ML to identify misconfigurations, and presents a case study that provides insight into the potential consequences of O-RAN system misconfiguration issues. This study draws on a large number of academic publications, white papers from engineering-focused initiatives, and industry and standardisation documents.

### 1.1. Motivation

According to a recent report by Mavenir [12], misconfiguration is the leading cause of cloud-data breaches. Another study by Positive Technologies [13] found that one in every three successful attacks on 4G networks is caused by faulty equipment configuration. In the context of commercial and open-source software, Zhang et al. [14] found that misconfiguration accounted for 31% of server downtime issues, compared to 15% for software faults. These statistics are relevant to the study of the O-RAN system, which is supported by open-source software and cloud computing. Furthermore, most existing 5G deployments primarily follow the Non-Standalone (NSA) approach, indicating a reliance on 4G infrastructure [1, 15].

Misconfigurations in O-RAN are critical, yet they have received little attention. Previous initiatives, including those of research bodies [16, 17, 5], telecommunication standardization bodies, such as O-RAN [18, 19, 20], 3GPP [21], cybersecurity agencies, such as ENISA [22], engineering-focused initiatives, such as TIP [23], and industry documents, such as Mavenir [12], Rimedo Labs [24], Ericsson [25], Rakuten Symphony [26], VMware [27], NEC [28], and others [29, 13], have primarily focused on analyzing security threats. These include threat models, security requirements, security procedures, risks, vulnerabilities, and attack vectors. It is worth noting that while these studies provide a comprehensive and informative overview, they do not provide an in-depth examination of the complexities associated with the potential deployment of O-RAN. In contrast to previous research, this article focuses on misconfigurations, which are a major problem for Mobile Network Operator (MNO)s due to their potential to degrade network performance and expose the system to security threats.

Misconfigurations are unavoidable in O-RAN owing to its open nature [17]. O-RAN supports multiple vendors' elements (e.g., Radio Unit (RU), Distributed Unit (DU), Central Unit (CU), and RIC), supports different versions of hardware and software (e.g., E2 Service Model (E2SM)s), operates across multiple technologies (e.g., multiple Radio Access Technology (RAT) and Standalone (SA) and NSA deployments), and facilitates multi-tenancy with different MNOs. Furthermore, the system's seamless deployment, integration, and operation depend on the joint efforts of many stakeholders or actors. Managing all of this complexity certainly increases the possibility of misconfigurations.

Human errors, whether made by component developers, integrators, engineers, or operators, are the leading cause of misconfiguration [16, 30]. These errors can appear in three forms: slips, which are unintentional errors during the configuration workflow; mistakes, which result from a lack of knowledge in a specific aspect of configuration; and violations, which are intentional errors committed under certain conditions, usually due to a failure to adhere to best practices or rules during peak workload hours [31]. Implementing advanced technologies such as SDN and NFV enhances network configuration accuracy and efficiency. This shift from manual procedures to automated processes reduces mistakes. The fast operation of these automated technologies, however, poses the possibility of increasing error probability. For example, software-based systems such as Virtual Network Function (VNF)s may include unnoticed build errors that have serious implications. Furthermore, even these advanced tools are operated by people, indicating a susceptibility to errors.

Therefore, it is critical to identify misconfiguration issues within O-RAN and investigate the possibilities of AI/ML to address them in order to improve the efficiency and security of RAN deployments.

### 1.2. Our contributions

The contributions are summarized as follows:

1. We provide an overview of AI/ML deployment options in the O-RAN system and offer detailed examples of actual applications. This highlights areas requiring additional studies and development in the application of AI/ML in O-RAN.
2. We provide a detailed analysis of misconfiguration problems in O-RAN, focusing on integration and operation, the use of SDN and NFV, and the use of AI/ML. Extensive examples are provided for each type of misconfiguration to aid understanding of the issues. To the best of our knowledge, this is the first analysis of misconfiguration issues in the context of O-RAN. This analysis reveals both opportunities for novel research solutions and identifies critical issues that must be addressed by network providers in their deployment of O-RAN.
3. We provide an analysis of misconfiguration detection approaches and emphasize how AI/ML can be employed for detection. Examples of Key Performance Indicator (KPI)s for each misconfiguration type are also provided.
4. We present an illustrative example of the impact of conflicting xApps to highlight the potential consequences of O-RAN misconfigurations and the potential of AI/ML to identify them.

The remainder of this paper is organized as follows. Section 2 presents the background of O-RAN and the application of AI/ML within O-RAN. The misconfiguration issues in O-RAN are analyzed in Section 3. Section 4 reports metrics and detection approaches for misconfiguration based on

**Table 1**
**List of important acronyms and definitions.**

| | | | |
|---|---|---|---|
| 3GPP | 3rd Generation Partnership Project | NG-RAN | Next-Generation RAN |
| 5GC | 5G Core | NDT | Network Digital Twin |
| A1 | Connects the non-RT RIC with the Near-RT RIC | non-RT | non-Real-Time |
| AI/ML | Artificial Intelligence/Machine Learning | NS | Network Slicing |
| ANN | Artificial Neural Network | NSA | Non-Standalone |
| CNN | Convolutional Neural Network | O1 | Connects the SMO with the O-RAN for FCAPS |
| CU | Central Unit | O2 | Connects the O-cloud with the SMO |
| CU-CP | CU- Control Plane | O-eNB | O-RAN enabled eNB |
| CU-UP | CU- User Plane | O-RAN | Open RAN |
| DNN | Deep Neural Network | O-RAN-SC | O-RAN Software Community |
| DoS | Denial-of-Service | PCA | Principal Component Analysis |
| DU | Distributed Unit | PRB | Physical Resource Block |
| E1 | Connects the CU-CP with the CU-UP | RL | Reinforcement Learning |
| E2 | Connects the Near-RT RIC with the E2 nodes | RAN | Radio Access Network |
| E2SM | E2 Service Model | RAT | Radio Access Technology |
| E-UTRAN | Evolved UTRAN | RLF | Radio Link Failure |
| eMBB | Enhanced Mobile Broadband | RIC | RAN Intelligent Controller |
| F1 | Connects the CU with the DU (Midhaul) | RRM | Radio Resource Management |
| FH | Fronthaul (Connects the DU with the RU) | RU | Radio Unit |
| FL | Federated Learning | SA | Standalone |
| FCAPS | Fault/Config/Accounting/Performance/Security | SDN | Software-Defined Networking |
| IDS | Intrusion Detection System | SLA | Service Level Agreement |
| KPI | Key Performance Indicator | SMO | Service Management and Orchestration |
| LSTM | Long short-term memory | SON | Self-Organized Networks |
| MAC | Medium Access Control | UE | User Equipment |
| MDP | Markov Decision Problem | URLLC | Ultra Reliable Low Latency Communications |
| MNO | Mobile Network Operator | UTRAN | Universal Terrestrial RAN |
| MLB | Mobility Load Balancing | VNF | Virtual Network Function |
| mMTC | massive Machine Type Communications | VM | Virtual Machine |
| MITM | Man-in-the-middle | WG | Working Group |
| NIST | National Institute of Standards and Technology | X2/Xn | Connects the gNB with other eNBs/gNBs |
| Near-RT | Near-Real-Time | xHaul | Transport network backhaul/midhaul/fronthaul |
| NFV | Network Function Virtualization | Y1 | Service interface for consumers |
| NG | Connects the NG-RAN with the 5GC | | |

AI/ML. This section also introduces the case study of detecting conflicting xApps. Finally, the conclusion and future research are presented in Section 5. Table 1 presents important acronyms and definitions used in this document.

## 2. Background

This section describes the architecture of O-RAN as well as the integration of AI/ML into O-RAN.

### 2.1. O-RAN architecture

Figure 1 depicts the O-RAN architecture, as defined by the O-RAN Alliance [32]. Table 1 contains the definitions of the components. The RAN is divided into three components: the CU, DU, and RU, each of which handles the NG-RAN protocol stack in various split configurations [33]. The CU is subdivided into CU- Control Plane (CU-CP) and CU-User Plane (CU-UP), which are in charge of Radio Resource Management (RRM) in the control plane and user plane, respectively. The E1 interface connects the CU-CP and CU-UP, while the F1 interface connects them to the DU. The DU and RU are linked together by the Fronthaul (Connects the DU with the RU) (FH) interface.

The RIC is the central component of the O-RAN architecture. The RIC is divided into two parts: the Near-Real-Time RIC (Near-RT RIC) and non-Real-Time RIC (non-RT RIC) that handle RAN resources on millisecond and second scales, respectively. The Near-RT RIC uses xApps (third-party apps) to control the CU, DU, and RU. The CU/DU/RU are represented as E2 nodes that expose E2SMs to the Near-RT RIC. The E2SMs describe RAN functions in an open and standardised manner. The Near-RT RIC also includes platform services such as subscription management, security, conflict mitigation, shared data layer, and message infrastructure. Although the O-RAN Working Group (WG)3 [34] has standardised some of these services, detailed specifications for the majority of them are still pending. In addition to xApps, the functioning of the Near-RT RIC is determined by policies received from the A1 interface and Y1 consumers.

The non-RT RIC, located inside the Service Manage-

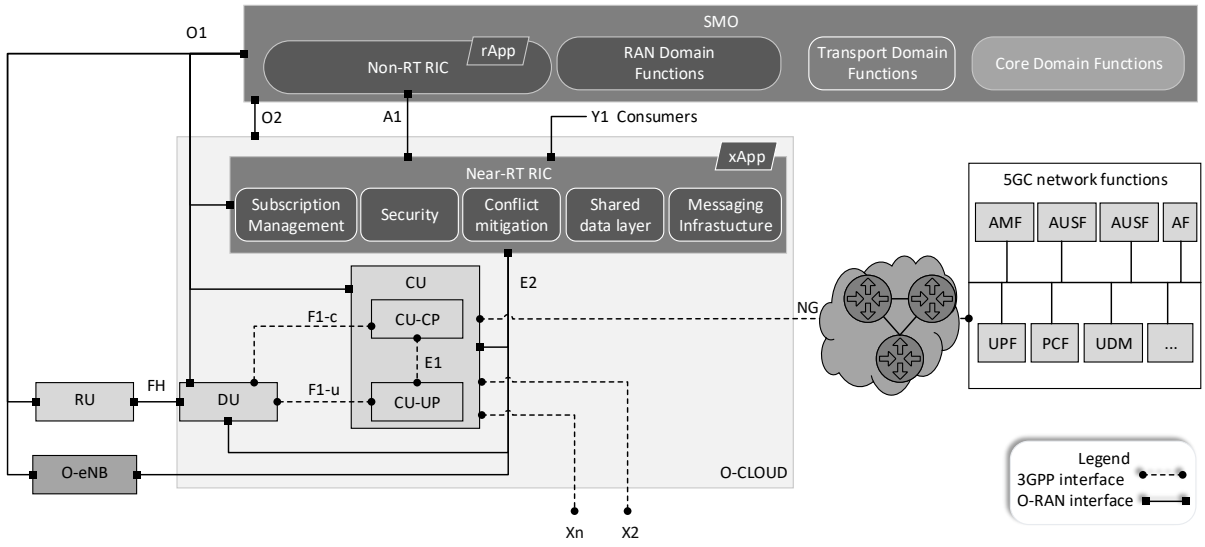


**Figure 1:** O-RAN architecture presented by the O-RAN Alliance and 3GPP [32] (the RAN is connected to the 5GC through the NG interface).

ment and Orchestration (SMO), is in charge of the long-term objectives in the RAN. The non-RT RIC does this by using rApps, which are third-party applications that build policies to operate xApps over the A1 interface. Furthemore, the non-RT RIC, in combination with other RAN domain functions in the SMO, facilitates RAN domain operation. The SMO's higher level of orchestration allows the development of end-to-end solutions by integrating functions across the RAN, transport, and 5G core domains.

The O1 interface allows communication between the SMO and the RIC, as well as between the SMO and the E2 nodes for Fault/Config/Accounting/Performance/Security (FCAPS). Furthermore, the O-RAN connects to the service-based 5G Core (5GC) via the NG interface, and to other g-NBs and e-NBs via the Xn and X2 interfaces, respectively. The O-RAN architecture also integrates O-eNB, which represents a monolithic RAN deployment, with the capabilities of E2SM. Finally, the O2 interface is a critical component of O-RAN, connecting the SMO to the cloud platform (O-Cloud).

For the first time, the 3GPP 5G-Advanced Rel. 18 has standardised the use of AI/ML in the operation of 5G NR [32, 5]. The next section examines the deployment possibilities for AI/ML within the O-RAN architecture.

## 2.2. AI/ML in O-RAN

Figure 2 shows different options to deploy AI/ML models in the O-RAN system according to the O-RAN and 3GPP [35, 36, 37]. These deployment options depend on the use case and thus cover different requirements.

In the single deployment option, the AI/ML model is positioned within a specific component of the O-RAN system. Aside from xApps and rApps, the RIC platform functions, as well as the CU and DU, can serve as hosts for AI/ML models, handling complex processes. In the O-RAN Near-RT RIC architecture, for example, the conflict mitigation function is integrated into a RIC platform function [34], which is expected to employ AI/ML-based conflict detection for xApps.

In the distributed deployment option, the AI/ML models are distributed across various O-RAN components. In the coordinated apps, two or more AI/ML-based applications can collaborate to handle integrated challenges, such as employing coordinated solutions based on rApps and xApps. Model splitting divides and assigns the AI/ML model to various linked parts. For example, some layers of an Artificial Neural Network (ANN) might be assigned to the end-device (user equipment), while the remainder layers are placed in a RIC application. Model sharing entails centralising the model in a component with high availability, high processing and storage capabilities, and enabling other O-RAN system parts with lower capabilities to download the model as needed. Finally, the employment of Federated Learning (FL) models in O-RAN is expected to address distributed applications while protecting user privacy.

Table 2 shows examples of applications that employ AI/ML in O-RAN. The current trend in academia and industry is to investigate single deployments, particularly xApps and rApps based on AI/ML for energy saving, load balancing and mobility optimisation, anomaly detection, and Network Slicing (NS). It is worth noting the collaboration between various industries and academic entities in the creation and testing of xApps and rApps (see Table 2). The cooperative effort of NetAI and VMware [38], where they demonstrated their energy-saving rApp, is an example of such collaboration. The O-RAN Software Community (O-RAN-SC) [39] contains a collection of open-source xApps created and validated by a variety of companies, including but not limited to

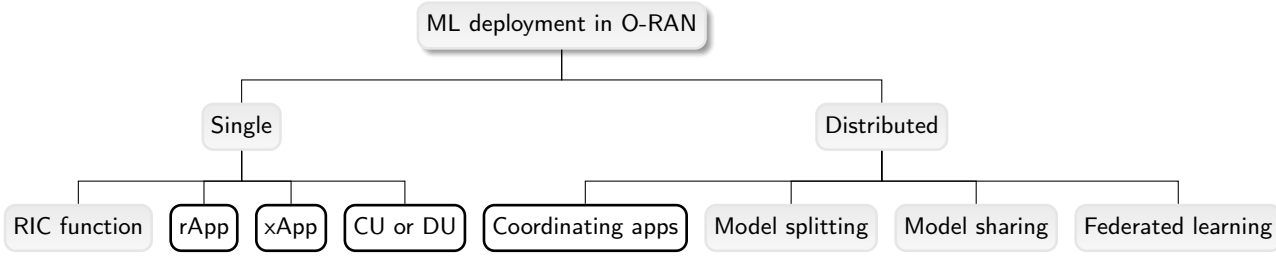


**Figure 2:** Deployment options of AI/ML models in O-RAN. In gray boxes: Minimally or unexplored areas.

AT&T, Rimedo Labs, and others. These collaborations provide a substantial contribution to the progress of O-RAN.

Coordinating apps based on AI/ML are emerging for distributed solutions, such as the energy-saving platform tested by Rimedo Labs and ONF [41], where a traffic steering xApp and an intelligent cell on/off rApp collaborate to optimise RAN energy consumption while maintaining service quality. Another study in [56] showcased an rApp that creates policies to control the behaviour of a RAN slicing xApp. The rApp selects resource allocation policies in the RAN slices using AI/ML, whereas the xApp implements such policies in near real-time.

Furthermore, model sharing has recently been investigated in [53]. In this study, an xApp situated in the Near-RT RIC trains an ML model to detect rogue base stations (RBS). The model is then transferred to the UEs, which use it to detect RBSs. The benefits of this technique include real-time detection, lower computational burden on the UE, and the fact that the models of all UEs in the RAN may be updated at the same time. More research is needed to deploy AI/ML in RIC platform functions as well as for model splitting, model sharing, and FL (see Figure 2).

As illustrated, the deployment of AI/ML in O-RAN has the potential to autonomously and efficiently manage RAN resources. Nonetheless, their implementation poses potential configuration challenges, which are included in the analysis presented in the next section (in particular, Section 3.3).

## 3. Misconfiguration problems in O-RAN

This section explores O-RAN misconfiguration issues in terms of integration and operation, enabling technologies, and AI/ML. Table 3 presents instances of these misconfiguration issues. The table also includes the impacted components and threats linked to each scenario. A misconfiguration problem, according to its definition, can influence either directly, impacting O-RAN performance, or indirectly, presenting a risk of greater susceptibility to threats to security inside O-RAN. These threats are also depicted in the table.

### 3.1. Integration and operation

As the O-RAN has numerous manufacturers, RATs (e.g., WiFi and New Radio (NR)), User Equipment (UE)s (e.g., vehicles and Internet of Things (IoT)), software versions (e.g., E2SMs), applications (e.g., eMBB and URLLC), and so on, it is exceedingly difficult to integrate and operate. The misconfiguration issues that may arise in this context are discussed below.

#### *3.1.1. Integration*

In an O-RAN, the lack of developed standard procedures might lead to uneven deployment. For example, noncompliance with typical xApp discovery, registration, and subscription processes in the Near-RT RIC will impact automated xApp deployment. Furthermore, because current O-RAN apps coexist in 5G and 4G (NR and E-ULTRA) in SA and NSA deployments, this integration might cause several setup issues. For example, the complex process of integrating LTE and 5G into NSA installations necessitates a careful setup and orchestration. This can help avoid mistakes that might jeopardize the overall system performance, such as bottlenecks and resource underutilization [76].

Inadequately built architectures; the use of unneeded or insecure parts (ports, services, accounts, privileges), functions, protocols, and components; and dependence on default configurations are other examples of integration issues. These misconfigurations not only expose the system to prospective attackers but also degrade system performance [18].

#### *3.1.2. Security function*

Three components are required to enable effective O-RAN protection: (i) protecting communication at all interfaces, (ii) guaranteeing the trust-based authentication of communicating endpoints, and (iii) leveraging trusted certificate authorities for identity provisioning [12]. The 3GPP and O-RAN Alliance released security assurance standards for the O-RAN interfaces, including backhaul, midhaul (F1), FH, O1, E2, A1, O2, E1, and Xn [19]. These requirements strive to reduce the threat surfaces to provide O-RAN confidentiality, integrity, and replay protection. In particular, a set of well-proven security protocols, such as SSHv2, Transport Layer Security (TLS), DTLS, IP security (IPsec), and MAC security (MACsec), was chosen.

Nonetheless, the complexities of security protocols, including several sophisticated setups and details, render these protocols vulnerable to misconfigurations. To facilitate the deployment of these protocols, their settings can be incorporated into open-source SSL/TLS libraries [16]. However, improper use of these libraries exposes the network to the introduction of rogue RUs, DUs, CUs, or RICs. Rogue el-

**Table 2**
Examples of existing xApps or rApps developed by industry, research bodies, and the O-RAN-SC for different use cases. The applications can be deployed as single (Sing) or distributed (Dist). Most of these applications use AI/ML.

| Use case | Provider | Application details | Dply | AI/ML details |
|---|---|---|---|---|
| Energy saving | Rimedo Labs [40] | rApp switches on/off cells based on user throughput and power consumption. | Sing | Uses Reinforcement Learning (RL). |
| | ONF [41] | rApp monitors the load of cell 1 and decides to switch it off. The xApp moves the traffic from cell 1 to cell 2, selected by the rApp. | Dist | Not specified. |
| | Nokia [38] | xApp guides gNBs to cover areas served by other gNBs, enabling the shutdown of those cells. | Sing | Not used. |
| | Ericsson [42] | rApp monitors the radio units and network, understands root causes of inefficiencies, and provides recommendations for resolution. | Sing | AI/ML for network clustering and modeling (AI/ML model not specified). |
| | Net AI and VMware [38] | xApp for carrier number forecasting for improved energy efficiency. | Sing | Net-AI forecasting engine (AI/ML model not specified). |
| Load balancing and mobility optimization | Orhan et. al. [43] | xApp for user-cell association and load balancing. | Sing | The problem is formulated using graph ANN and solved using DRL. |
| | Rimedo Labs [40] | xApp commands handover operations based on A1 policies. | Sing | Not used. |
| | O-RAN-SC [39] | xApps for load prediction by CCMC and traffic steering by AT&T and UTFPR. | Sing | Not specified. |
| | Lacava et. al. [44] | xApp maximizes the UE throughput utility through handover control. | Sing | The problem is formulated as a Markov Decision Problem (MDP) and solved using Reinforcement Learning (RL). |
| | Mahrez et. al. [45] | xApp balances the load across cells and optimizes the handover process. It uses xApps for KPI monitoring and anomaly detection. | Sing | Uses isolation forest model to detect anomalous UEs. |
| | Ntassah et. al. [46] | xApp performs UE handovers based on the predicted cell throughput. UE clustering speeds up decision-making. | Sing | K-means for UE clustering and LSTM for cell throughput prediction. |
| | Kasuluru et. al. [47] | xApp forecasts the demand of PRBs of the CU. | Sing | Probabilistic forecasting: Transformers, Simple-Feed-Forward, and DeepAR. |
| | Boutiba et. al. [48] | xApp for dynamic time duplex division (D-TDD). | Sing | Deep deterministic policy gradient for optimal TDD configuration based on uplink/downlink demands. |
| | Mavenir [49] | xApps and rApps for load distribution, Mobility Load Balancing (MLB), Mobility Robustness Optimization (MRO), coverage and capacity optimization (CCO), automatic neighbor relation (ANR), beam control, smart scheduler, and traffic steering. | Sing | AI/ML utilized without specific details being provided. |
| Anomaly detection | Ericsson [50] | rApp analyzes the RAN to detect and classify cell issues. | Sing | AI/ML is used to detect anomaly cells, classify coverage, handover or external issues, and correlate each issue to its root cause level. |
| | Kryszkiewicz et. al. [51] (Rimedo Labs) | xApp detects jamming attacks based on Reference Signal Received Power (RSRP) and Channel Quality Indicator (CQI) values reported by UEs. | Sing | Kolmogorov–Smirnov (KS) is used to detect distribution changes. |

Continuation of Table 2

| Use case | Provider | Application details | Dply | AI/ML details |
|---|---|---|---|---|
| Anomaly detection | Hoffmann et. al. [52] (Rimedo Labs) | xApp is used to model KPI profiles based on the Random Access Channel response. | Sing | Anomaly detection based on the mean value and standard deviation of the KPI. |
| | Huang et. al. [53] | xApp uses the signal strength stability feature to detect rogue base stations (RBS). | Dist | The xApp trains the models RF, KNN, and SVM and transfers them to the UE for RBS' detection. |
| | O-RAN-SC [39] | Anomaly detection by HCL, KPI monitor by Samsung, signaling storm detection by Samsung. | Sing | Not specified. |
| Network slicing | Johnson et. al. [54] | NexRAN xApp performs closed-loop RAN slicing control, using E2SMs for KPI monitoring and NS. | Sing | Not used. |
| | Yeh et. al. [55] (Intel) | xApp determines the quantity of radio resource for each NS and the MAC schedules and enforces these allocations. | Sing | LSTM, temporal CNN, and Seq2Seq for traffic load prediction. |
| | Mallu et. al. [56] | rApp sets policies that regulate xApp behaviour while slicing. These rules govern how xApp manages RAN resources. | Dist | ML for policy selection. |
| | Wiebusch et. al. [57] | xApp predicts uplink resource requirements for UEs. | Sing | LSTM for UE's payload prediction. |
| | Tsampazi et. al. [58] | xApp allocates the PRBs for each slice and decides which MAC scheduling is used per slice. | Sing | Deep RL is used for slicing and scheduling (12 DRL designs are tested.) |
| | Zhang et. al. [59] | Power control xApp and slice-based resource allocation xApp are coordinated to optimize the use of resources in O-RAN. | Dist | The two xApps use Markov Decision Problem (MDP) and Reinforcement Learning (RL), and are coordinated with FL. |

ements can lead to Man-in-the-middle (MITM) attacks that eavesdrop, change, stop, or delay messages in both the control and user planes [17, 64].

Furthermore, adding strong protection measures for interfaces with strict timing constraints might reduce the O-RAN performance. The security of the FH interface is an example of this difficulty. An optimal security protocol option — TLS, IPsec, or MACsec — must consider the overhead associated with bandwidth and latency [64].

Finally, sensitive data in the ORAN system should also be protected. These data include the following: (i) data from system functions, such as logging messages, configuration file exports, CLI, or GUI configurations; (ii) authentication data, such as PINs, passwords, cookies, and cryptographic keys; and (ii) data from system elements, such as UE information, RAN topology information, and ML databases, which contain critical information from the system [77].

### *3.1.3. Conflicting policies*

The total automation of the O-RAN architecture in 5G requires global orchestrators such as the SMO, as well as local orchestrators such as the RIC. Through a human-machine interface, these orchestrators allow operators to communicate their objectives in a high-level language. These intents are subsequently turned into policies that regulate and run various O-RAN system components.

Many misconfigurations might occur when administering policies in O-RAN. When converting intents into low-level rules for operating system components, for example, the quantity of rules created may exceed the system's resources. Furthermore, the time necessary for rule creation, enforcement, and verification may surpass the performance requirements [78, 79].

When several actors seek to manage a function, policy violations become a big challenge. This situation is demonstrated in the O-RAN system by the functioning of an xApp, which receives policies via the A1, O1, and Y1 interfaces. In such cases, failures might develop due to the interplay of different components introducing contradictory policies [30]. When chaining pieces with diverse functions, each with a unique configuration, conflicting rules represent an increased risk. This complication impacts policy communication, potentially resulting in the development of duplicate, shadowed, correlated, or nested rules coming from different intents [80].

The multivendor environment causes conflicts between xApps and rApps in O-RAN. Consider the xApps for conflict power allocation and radio resource allocation [81]. In this case, the power allocation xApp may assign a high transmission power to one resource block, while the radio resource allocation xApp assigns this resource block to a user with a low traffic load. This disagreement will waste limited bandwidth and increase power usage. Additional conflicts may develop as a result of RRM choices made by O-gNB or O-eNB nodes and xApps, potentially creating network instability.

**Table 3**
Misconfiguration problems in O-RAN: Examples of misconfiguration, impacted components, and potential direct (performance) and indirect (security) threats are shown. The example ID is provided for reference in the association with detection approaches in Table 4.

| Area | Aspect | (ID): Example of misconfigurations | Impacted components | Potential threats |
|---|---|---|---|---|
| I&O | Integration | (E1): Enabled default ports, services, accounts, and privileges [18]. | Non-RT RIC, Near-RT RIC, CU, DU, RU. | Security: intruders. |
| | | (E2): Lack of conformance or interoperability with standard procedures (e.g., xApp registration). | Near-RT RIC, Non-RT RIC, xApp, rApps, O2, O1, E1, F1, A1, E2. | Performance: outages. |
| | | (E3): xApps access data from the E2SMs beyond what is strictly necessary. | xApp, CU, DU. | Performance: monitoring overhead. Security: data exposure. |
| | | (E4): Conflicting IP configuration of end-points. | Near-RT RIC, Non-RT RIC, xApp, rApps, CU, DU, RU. | Performance: outages. |
| | | (E5): Utilizing outdated E2SMs. | xApp, CU, DU. | Performance: outages. Security: node exposure. |
| | Security function | (E6): Disabled or improper configuration of security protocols (e.g., SSH) to protect reference points [12, 19]. | A1, O1, O2, E2, F1, E1. | Security: intruders. |
| | | (E7): Disabled or improper configuration of mutual authentication of end-points [60]. | Near-RT RIC, Non-RT RIC, CU, DU, RU. | Security: rogue end-points. |
| | | (E8): Lack of failover for endpoint crashes. | Near-RT RIC, Non-RT RIC, CU, DU, RU. | Performance: outages. |
| | | (E9): Sub-optimal balance between security and CPU utilization in E2 encryption [61]. | CU, DU, RU. | Performance: high CPU usage. |
| | | (E10): Sub-optimal equilibrium between FH protection and performance [62, 63, 64]. | FH. | Performance: high delay and low throughput. |
| | Conflicting policies | (E11): Previous xApp not uninstalled before new installation. | xApp, CU, DU. | Performance: instability. |
| | | (E12): Sub-optimal rule generation. | Non-RT RIC, Near-RT RIC, xApp, rApp, CU, DU. | Performance: resource wastage. |
| | | (E13): A1 policies demand more resources than are available. | Near-RT RIC, xApp, CU, DU. | Performance: resource depletion. |
| | | (E14): Meeting E2 policies involves the demand for high energy usage by E2 nodes [65]. | xApp, CU, DU. | Performance: energy wastage. |
| | | (E15): Conflicting access to radio resources by xApps. | xApp, CU, DU. | Performance: instability. |
| SDN & NFV | SDN | (E16): Malformed packets [66]. | xHaul | Performance: packet retransmission. |
| | | (E17): Unsynchronized controller instances [67]. | xHaul | Performance: instability. |
| | | (E18): Sub-optimal controller placement [68]. | xHaul | Performance: high latency, low reliability, energy wastage. |
| | | (E19): Inconsistent directives between the controller and stateful network devices. | xHaul | Performance: instability. |
| | | (E20): Violation of firewall application [69]. | xHaul | Security: DoS, port scanning. |
| | NFV | (E21): Sub-optimal initial resource assignment during image creation [70]. | Non-RT RIC, Near-RT RIC, xApp, rApp, CU, DU. | Performance: resource wastage, container halting. |
| | | (E22): Sub-optimal service migration [71, 72]. | Non-RT RIC, Near-RT RIC, xApp, rApp, CU, DU. | Performance: service downtime. |

Continuation of Table 3

| Area | Aspect | (ID): Example of misconfigurations | Impacted components | Potential threats |
|---|---|---|---|---|
| SDN & NFV | NFV | (E23): Excessive fragmentation of vDU functions across numerous microservices [33]. | DU. | Performance: high latency, intensive inter-service comms. |
| | | (E24): Incorrect timing/sync between vDU and RU (PTP) [33]. | DU, RU. | Performance: low reliability. |
| | | (E25): Lack of VM/container isolation [73]. | Non-RT RIC, Near-RT RIC, xApp, rApp, CU, DU. | Performance: inconsistency. Security: intruders. |
| AI/ML | Performance and reliability | (E26): Sub-optimal granularity for data collection [74]: Reliability vs. overhead vs. privacy. | A1, E2, CU, DU, xApp, rApp. | Performance: unreliability. Security: data exposure. |
| | | (E27): Unreliable AI/ML model sharing. | rApp, xApps, dApp, E2, A1. | Performance: high end-to-end delay, loss of model data. |
| | | (E28): Misplacement of AI/ML model object [75]. | rApp, xApps, dApp. | Performance: high end-to-end delay. |
| | | (E29): Implicit conflicts between AI/ML decisions. | rApp, xApp, dApp, CU, DU. | Performance: instability. |
| | Model protection | (E30): Sub-optimal protection of training data: encryption vs. reliability. | xApp, rApp, dApp. | Performance: high end-to-end delay. Security: poisoning attacks. |
| | | (E31): Lack or improper anonymization of UE information. | E2, A1. | Security: data exposure. |
| | Explainability | (E32): Use of DNN where not needed. | xApp, rApp, dApp. | Performance: low accuracy, resource wastage. Security: adversarial attacks. |
| | | (E33): Too complex design of AI/ML model. | xApp, rApp, dApp. | Performance: low accuracy. Security: lack of trustworthiness. |

Finally, efficient policy management requires using as few resources as possible inside the O-RAN system, such as containers [79], RUs, and energy usage [65].

## 3.2. Enabling Technologies: SDN and NFV

O-RAN relies heavily on SDN and NFV for programmability and flexibility. However, as analyzed below, they also carry the risk of misconfiguration.

### *3.2.1. SDN*

This technology enables configurable data planes in xHaul networks, as required by end-to-end 5G and 6G systems. Several misconfiguration issues might arise during the functioning of these networks. The integrity of data packets, for example, can be altered by network elements and controlling programmes throughout the transmission process. This effect may manifest as alterations to the packet header, such as changing the VLAN value, resulting in re-transmission events and packet loss. Other data transmission breaches include way-pointing violations, in which packet routes differ from the anticipated device sequence, and traffic locality violations, in which packets must stay inside a defined area [66].

Emerging data plane programmability technologies, such as P4, hold the potential to increase operational flexibility. However, combining P4 with network controllers presents substantial setup issues, such as selecting which operations are offloaded to P4. Furthermore, independent decisions made by P4 devices may result in discrepancies with the controller, resulting in network instability.

Flow-based network management, made possible by software-based controllers, is critical in 5G operations. Nonetheless, misconfigurations caused by several coexisting applications in the controller may result in high-level forwarding policies that the data plane cannot follow [67]. Furthermore, unsynchronized controller instances and uncontrolled network device failures (e.g., switch port failure) might interrupt flow trajectories. These flaws may cause traffic to be dropped or sent via inappropriate paths.

Finally, firewall applications are among the most significant applications in the programmable data plane. A significant challenge in this domain is ensuring continuous compliance of the data plane with the security policy deployed in the firewall application. Inadequate or incorrect network policy, controller software, and packet trajectory verification might result in partial or total firewall application violations [69].

### *3.2.2. NFV*

Every element inside the O-RAN architecture has the possibility of virtualization using technologies such as virtual machines and containers. This technique improves the

RAN operations' flexibility and scalability. However, it increases the possibility of misconfigurations.

The initial setup of resources for a virtual component is critical. To achieve optimal service performance, the operator must establish the proper CPU and memory allocation. When the workload exceeds capacity, insufficient resources may result in service failure or decreased performance [70]. Allocating more resources than necessary for an application, on the other hand, leads to resource waste, which raises deployment costs.

While it is true that virtualized pieces may be scaled, these operations may cause further configuration problems and disrupt service continuity. For example, virtual service replication and migration are used to solve heavy workload scenarios or in the case of a breakdown. Due to the slow replication process or message rerouting, system state inconsistencies may occur in replication, resulting in inconsistent management between the original and replica services [71]. Migration, on the other hand, presents issues such as service recovery time and probable data loss [72].

Furthermore, the increased complexity of controlling and orchestrating many virtual functions increases the potential for misconfiguration, such as insufficient network isolation between separate network functions [73]. Similarly, configuration inconsistencies may arise, such as when a virtual firewall defined at the tenant level is possibly circumvented at the underlying cloud infrastructure level. Inconsistencies of VNFs might reduce system performance and expose services or infrastructure to threats to security [82, 83].

## 3.3. AI/ML

Given that AI/ML is a primary driver of O-RAN advancements and its implementation in O-RAN has already begun (see Table 2) investigating AI/ML misconfiguration issues is critical. In the following, they have been recognised in terms of performance and reliability, model protection, and explainability.

### *3.3.1. Performance and reliability*

Misconfigurations throughout the life cycle of AI/ML applications can have a negative influence on their reliability and performance. In data collection, for example, improperly setting the data resolution within the system for monitoring leads to problems like: (i) insufficient granularity resulting in inefficient controls, such as failure to identify events; (ii) unnecessary high resolution in the monitoring system resulting in system overhead, such as E2 channel saturation; and (iii) the possible disclosure of sensitive data, such as UE-related details.

O-RAN suffers from an absence of data monitoring frameworks specialised to AI/ML applications [74]. For example, the current version of E2SM KPM.v3 (O-RAN WG3) provides detailed metrics for the RAN system that are aligned with 4G LTE (3GPP). These metrics, however, may not fulfil the criteria for particular security solutions, such as strong DoS detectors, which require information at either the packet or flow level.

More proactive E2 nodes can help respond to outages or satisfy the low-latency needs of 6G networks. Integrating pre-processing algorithms, such as Principal Component Analysis (PCA) and auto-encoding, in E2 nodes might help minimise information transfer between these nodes and xApps as well as simplify the AI/ML model structure. Similarly, data augmentation approaches, such as generative adversarial learning, may be useful in supplementing data-hungry applications or in cases when O-RAN system samples are insufficient or unbalanced [84, 85, 86].

Another misconfiguration issue is the lack of protection of user privacy in data collection. As the RAN processes information from all UEs, their data privacy must be secured against AI/ML activities, which are handled by third-party xApps or rApps in O-RAN [37, 21]. Details such as UE position and trajectory forecasts are examples of privacy-sensitive information. Neglecting data privacy issues exposes the system to the possibility of data leaks, which can result in legal ramifications, monetary fines, loss of customer trust, and harm to the reputation of the entities involved.

In terms of AI/ML model performance in O-RAN, if the models fail to achieve the basic requirements given by the use cases, including factors such as accuracy, model size, convergence time, and prediction time, they become unreliable, [87, 75]. AI/ML models may be deployed at multiple locations (see Figure 2) depending on the use case, such as xApps, rApps, and dApps (those deployed at DUs and CUs [75]). However, if the models are misplaced, they might cause unacceptable delays in the target application's end-to-end control loops, resulting in a decline in system performance rather than a benefit.

For distributed deployments of AI/ML (see Figure 2), the O-RAN system must guarantee communication reliability of no less than 99.999% to support the exchange of data and model parameters and to enable communication across modules or partitions of the models [36]. Furthermore, it is critical to provide consistent data feeding as well as the availability of storage and processing resources for AI/ML models as and when they are required. Failure in certain arrangements might lead to unreliable model output.

Finally, AI/ML model decisions may clash with other functionalities inside the O-RAN system. In particular, the incorporation of AI/ML in O-RAN has the potential to generate very complex conflicts, namely implicit conflicts. These conflicts may cause delayed reactions inside the system, making identification a difficult task.

### *3.3.2. Model protection*

Adversarial attacks against AI/ML models, including data poisoning, evasion attacks, and API-based attacks, have been investigated in recent years [88, 6]. Data poisoning attacks affect the AI/ML model training phase, causing the model to learn incorrectly. Data injection, data manipulation (labels, features, and learning parameters), and logic corruption are examples of these attacks. Evasion attacks, such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), target the model inference phase.

Model extraction, model inversion, and membership inference are examples of API-based attacks that take advantage of the exposure of the AI/ML front-end.

Notably, adversarial attacks have greater entrance hurdles in a monolithic and single-vendor RAN architecture since they often lack access to the AI/ML models for most applications [89]. However, the entrance barriers to such attacks are significantly decreased in the O-RAN system, where the components are disaggregated and third-party suppliers of hardware and software are included [90, 91, 6].

Both 3GPP [21] and O-RAN Alliance [18] have conducted studies to better understand the risks connected with the usage of AI/ML models. Three threat models against the AI/ML system were found: (1) poisoning attacks, (2) modifying the ML model, and (3) transfer learning attacks [18]. The lack or misconfiguration of protection for AI/ML models, as well as the use of public datasets to train the models, are the prevalent flaws across these threat models.

Recent efforts in [92, 93, 94, 95] have demonstrated the possibility of adversarial attacks on O-RAN operations. However, they are confined to analysing public datasets or employing minimalist testbeds, raising the question: Is there still a risk of adversarial attacks if the security functions have been appropriately established, i.e., safeguarding the communication interfaces and guaranteeing adequate authorization and authentication to access the AI/ML model and data? At first glance, correct security function configuration may avoid poisoning and API-based attacks. Implementing encryption and decryption methods for training databases at the Near-RT RIC, for example, can function as a measure to protect against data contamination by malicious xApps adopting adversarial approaches, such as FGSM and PGD [96]. However, the question is whether these encryption and decryption operations can be implemented in O-RAN without severely influencing the performance of AI/ML applications.

In cases of evasion attempts, the persistence of the attacks can be seen regardless of whether the security protections inside the O-RAN architecture are appropriately implemented [89]. This is because, even with minimal knowledge of AI/ML processes in the RAN, UEs can operate as adversarial agents, impacting the performance of different applications, e.g., automated modulation categorization and forecasting the Channel Quality Indicator (CQI) [89].

#### *3.3.3. Explainability*

As AI/ML finds use within different areas of O-RAN across 5G and 6G, the lack of explainability within these models may cause significant hesitation, especially when using them in safety-critical use cases such as transportation automation (e.g., trains and Unmanned Aerial Vehicle (UAV)s), vital infrastructure operation (e.g., water and nuclear energy), healthcare, and human-machine brain interfaces [97, 98]. This is especially important when employing Deep Neural Network (DNN)s, which are data-driven models able to surpass standard mathematical or probabilistic models. Yet, such DNNs operate as complex black box models, making it challenging to explain the decisions they make to human specialists, considering the underlying data support and causal logic.

Poorly designed DNN solutions can exacerbate the explainability problem in O-RAN. For example, when a system's mathematical model is well-established, the employment of DNN becomes superfluous. In these circumstances, traditional statistical or signal processing approaches may outperform DNNs. Incorporating DNNs in such settings not only reduces performance and increases vulnerability to adversarial attacks, but it also lacks the critical feature of explainability [7, 97].

Furthermore, the adoption of a sophisticated DNN architecture with an excessive number of parameters and layers, the employment of complex activation functions, and the lack of preprocessing procedures for input features all contribute to DNNs' increased complexity. Such complicated DNN architectures are unneeded in many cases. This unnecessary complexity not only raises the processing needs for DNN decision-making, but it also increases the danger of model overfitting. It is crucial to highlight that certain data-driven models are intrinsically explainable, such as rule-based models, linear models, Bayesian inference, and decision trees. Depending on the application, these solutions may efficiently replace sophisticated DNNs with negligible performance loss.

### 3.4. Summary and insights

Many of the misconfiguration issues presented in Table 3 have been seen in prior systems that incorporated SDN and NFV. Nonetheless, poor setups of these elements become ever more important in the RAN ecosystem, where resources are scarce and expensive. Furthermore, because the application of AI/ML in RAN is new, its effective integration and operation have the potential to cause significant misconfiguration issues, as seen in Table 3.

In addition, the misconfigurations examined in this section may arise at different stages during the implementation lifecycle of the O-RAN system. Recall that misconfigurations are allowed or induced unintended behaviours [10], as described in Section 1. For instance, despite a system operator being provided with the standard specifications and industry best practices for securing O-RAN system interfaces (see (E6) in Table 3), the complexity of the system may result in the operator making inadvertent errors. Seemingly minor misconfigurations can lead to significant security impact, such as system intrusions.

Other misconfiguration problems may be harder to avoid during the early stages of the lifecycle, such as conflicting xApps (see (E2) in Table 3) in a multi-vendor ecosystem. In such cases, the applications may be correctly developed and integrated into the system, but their combined configurations may cause conflicts during operation.

The optimal approach to deal with misconfigurations is to prevent them by adhering to standards, best practices, and employing rigorous verification procedures. Nevertheless, as previously stated, misconfigurations cannot be en-

tirely eradicated, necessitating the consideration of detection strategies, as detailed in the following section.

## 4. AI/ML for misconfiguration detection

This section first describes the problem of misconfiguration detection. Then, it provides an overview of various misconfiguration techniques, emphasizing the role of AI/ML in enhancing detection efficacy.

### 4.1. The problem of misconfiguration detection

In terms of detection, the primary challenge lies in identifying the origin of misconfigurations. For example, if outages occur within our system, various factors could be responsible. One cause might be an xApp failing to adhere to standard procedures when interacting with the E2SMs (see (E2) in Table 3), resulting in its malfunction within the system. Alternatively, conflicts between the xApp and internal RIC functions or other xApps could also lead to outages (see (E15) in Table 3). In this scenario, monitoring the system (using logs, configuration files, KPIs, network packets, etc) or using representation models of the O-RAN system can help locate the problem. Also, it is essential to acknowledge that varied monitoring/detection methods are necessary depending on the specific types of misconfigurations.

In the next section, several approaches for detecting misconfigurations are examined. Given the intricacies of the O-RAN system, manual or partially automated detection methods might not be sufficient. Therefore, we highlight the benefits of leveraging AI/ML techniques to further improve misconfiguration detection performance.

### 4.2. AI/ML-assisted detection approaches

This section explores the potential of AI/ML to detect misconfiguration problems in O-RAN. In this context, Table 4 shows instances of AI/ML-based misconfiguration detection approaches covering various misconfiguration challenges presented in Table 3. Table 4 also displays the KPIs for each misconfiguration problem. The misconfiguration detection approaches are described below.

#### *4.2.1. Active monitoring*

This strategy involves interacting with the system by sending synthetic service requests or probe packets to uncover any misconfigurations within it. For example, to discover enabled default ports in O-RAN (misconf. I&O-(E1) in Table 4: enabled default ports, services, accounts, and privileges), a series of service requests to the target ports can be produced. The same method can be used to detect deactivated security protocols (misconf. I&O-(E6) in Table 4: disabled or improper configuration of security protocols to protect reference points), such as TLS for A1, by sending and evaluating synthetic connection requests. Note that most integration and security function misconfigurations (I&O in Table 3) can be addressed by active monitoring. In these cases, data analytics may be utilized to process large amounts of data.

Furthermore, periodically sending probe packets across the network helps acquire network status metrics such as latency and bandwidth. This method increases network traffic and only detects potential misconfigurations once they have already affected the system, indicating a reactive approach [100]. For example, consider the misconfiguration I&O-(E10) in Table 4 (sub-optimal equilibrium between FH protection and performance). To detect this misconfiguration, a number of packets can be periodically sent to check any inconsistency in the expected end-to-end latency and throughput (based on SLAs). In this setup, the use of AI/ML models enables the detection of anomalies in measurements.

#### *4.2.2. Passive monitoring*

In contrast to active monitoring, this method involves the study of system elements without the use of probe packets. These solutions often employ sniffer tools for real-time telemetry. For example, the message flow within the Near-RT RIC interfaces (A1, E2, O1, Y1) can be monitored to identify protocol misconfigurations (misconf. I&O-(E2) in Table 4: lack of conformance or interoperability with standard procedures), such as xApp registration/deregistration with the Near-RT RIC. AI/ML-based analytics may be used to detect abnormalities in real-time telemetry and identify these misconfigurations [100]. It should be noted that this approach is reactive.

#### *4.2.3. Formal verification*

In this approach, the system is formalised using symbolic methods, such as geometry and set theory, and verification techniques are used to detect misconfigurations. These methodologies can provide rigorous evidence of configuration conformance or violation. However, due to the huge scale of the O-RAN system, these verification approaches may be too expensive. Furthermore, verification delays might result in a substantial time gap during which the network may face lower performance and greater exposure to security attacks [78]. The combination of AI/ML and formal approaches allows for the speedy and verified discovery of misconfigurations, as demonstrated in [73]. For example, to find abnormalities in the generation of policies in O-RAN (misconf. I&O-(E12) in Table 4: sub-optimal rule generation) (e.g., A1 policies), the rule generation can be represented using a minimal interval set model [78]. AI/ML can be used to learn the correlation between this representation and the associated problems (e.g., redundant rules).

The same approach can be applied to detect misconfiguration AI/ML-(E28) from Table 4 (misplacement of AI/ML model object). However, in this case, a tree graph can serve to model the deployment of a set of AI/ML models in an O-RAN system. Then, a formulation based on binary integer linear programming (BILP) can be applied to find the correlation between the proper placement of the AI/ML models and the system's performance (e.g., accuracy of the AI/ML models, end-to-end latency, etc.) [75].

**Table 4**
Examples of detection approaches, KPIs, and use of AI/ML for different misconfiguration types in O-RAN.

| Misconfiguration ID | Detection approach | Description | KPIs | Use of AI/ML |
|---|---|---|---|---|
| I&O-(E1) | Active monitoring | Port scanning and service scanning. | Number of open ports, default accounts, and default passwords. | Data analytic. |
| I&O-(E2) | Passive monitoring | Sniffing of packets for analysis. | Number and type of procedure violations. | Data analytic. |
| I&O-(E6) | Active monitoring | Security protocol verification (e.g., SSH [99]). | Number and type of check failures. | Data analytic. |
| I&O-(E9) | Active monitoring | Packet injection and metric collection. | Round trip time, processing delay, transmission delay, throughput, and CPU utilization. | Anomaly detection. |
| I&O-(E10) | Active monitoring | Packet injection and metric collection. | End-to-end latency and throughout. | Anomaly detection. |
| I&O-(E12) | Formal verification | Model of minimal interval set [78]. | Size of generated rule sets, number of redundant rules, i.e., correlated, shadowing, and imbrication [80]. | Learning of the representation and anomaly detection. |
| I&O-(E15) | NDT | Creation and testing of risk scenarios. | Number and types of conflicting access to resources. | Creation of scenarios and anomaly detection. |
| SDN&NFV-(E16) | Passive monitoring | Sniffing packets for analysis. | Number of malformed packets, header alterations, waypointing violations, and packet re-transmissions. | Data analytic. |
| SDN&NFV-(E18) | Offline modeling | Creation of a model based on network topology and SDN controller configuration [68]. | Round-trip time, switch-to-controller traffic, and controller-to-controller traffic. | Anomaly detection. |
| SDN&NFV-(E20) | Offline modelling | Verification of firewall configuration. | Number of blackholes and path violations (entire or partial). | Network modeling and creation of scenarios. |
| SDN&NFV-(E21) | Active monitoring | Injection of service requests. | Relative CPU usage, memory usage ratio, ratio of service requests, and latency to treat service requests. | Data analytic. |
| SDN&NFV-(E22) | NDT | Creation and testing of migration scenarios. | Downtime of service, UE recovery time, and latency of service. | Creation of scenarios and anomaly detection. |
| AI/ML-(E26) | NDT | Creation of scenarios and testing of data capturing frameworks. | Accuracy, end-to-end latency, and total data disclosure incidents. | Data analytic. |
| AI/ML-(E28) | Formal verification | Representation of the AI/ML model placement using a tree graph [75]. | Accuracy, end-to-end latency, and number of conflicting decisions. | Formulation and solution of the optimization problem. |
| AI/ML-(E30) | Active monitoring | Data retrieval requests. | End-to-end latency, accuracy, and attack success rate. | Creation of attack scenarios. |
| AI/ML-(E33) | Offline modelling | Variogram for feature sensitivity analysis. | Accuracy and end-to-end latency. | Anomaly detection. |

#### *4.2.4. Offline modeling*

This approach entails parsing the network configuration to provide a quantitative model of the network, enabling the proactive detection of misconfigurations that might compromise meeting Service Level Agreement (SLA) goals. Using configuration files (logs and configuration databases) as training sets, AI/ML models may learn basic specs. This approach allows for the discovery of SLA violations in the offline model before they occur in the actual implementation. It should be noted that this technique falls short of recording dynamic traffic fluctuations, resulting in the overlooking of some SLA violations [101]. Consider, for example, misconfiguration SDN&NFV-(E18) in Table 4 (sub-optimal controller placement). In this example, the network topology configuration and specifics of SDN controllers (number of controllers, location, and design) can serve as inputs to an ANN to predict system performance (e.g., throughput and latency) [68]. Based on this model, the SDN placement that causes performance degradation can be identified.

#### *4.2.5. Network Digital Twin (NDT)*

In this approach, a live virtual representation of O-RAN enables a variety of actions, including emulations, testing, optimisation, monitoring, and analysis of novel configurations in a risk-free environment. This reduces the need for real network deployment, resulting in a proactive approach [102]. Using an NDT of the RIC, for example, enables testing of multiple xApps to assess performance and discover any conflicts (see misconfiguration I&O-(E15) in Table 4: conflicting access to radio resources by xApps). In this example, AI/ML can help generate conflicting scenarios. The same approach can be applied to detect misconfiguration SDN&NFV-(E22) in Table 4 (sup-optimal service migration), where NDT can be used to create migration scenarios and to detect potential anomalies on monitored KPIs (e.g., inadequate downtime and latency of services). It should be noted that the implementation of NDT necessitates massive resources in terms of storage, computation, maintenance, and the precision required by the models. AI/ML can help improve the efficiency and precision the of simulate networks and scenarios

It is worth noting that while identifying misconfiguration issues, not every issue necessitates the use of AI/ML approaches. Some integration and operation (I&O) misconfiguration concerns in Table 4 can, for example, be automated without the use of AI/ML, such as calculating the number of open ports and system default accounts and passwords. Nonetheless, due to the vast number of components and interfaces in an O-RAN system, data analytics might be useful in discovering configuration issues in massive datasets.

Furthermore, it is important to recognize that some KPIs in Table 4, such as end-to-end latency and throughput, might signify distinct misconfiguration issues. As a result, tracing back to the source of the misconfiguration to establish the precise misconfiguration type to ease remediation is a major difficulty.

Finally, given the diversity of misconfigurations, different detection approaches may be better suited for certain misconfiguration instances. Therefore, integrating detection approaches into a unified tool can facilitate misconfiguration detection, classification, root cause analysis, and reporting. This system would most likely employ AI/ML to automate and orchestrate the diagnostic process.

### 4.3. Case study: Detection of conflicting xApps

This section analyses conflicting xApps. Initially, this misconfiguration problem and its impact in the RAN is described. Then, a detection framework based on insights from prior research is provided.

#### *4.3.1. Problem description*

Differing from previous RAN generations, in a multivendor O-RAN environment, Near-RT RIC xApps maintain a high level of independence in their optimisation or learning process, with only essential data shared between them. In this sense, xApp developers assume direct and isolated management of the RAN. This condition may result in numerous overlooked conflicts arising during the combined operation of the xApps within the O-RAN system.

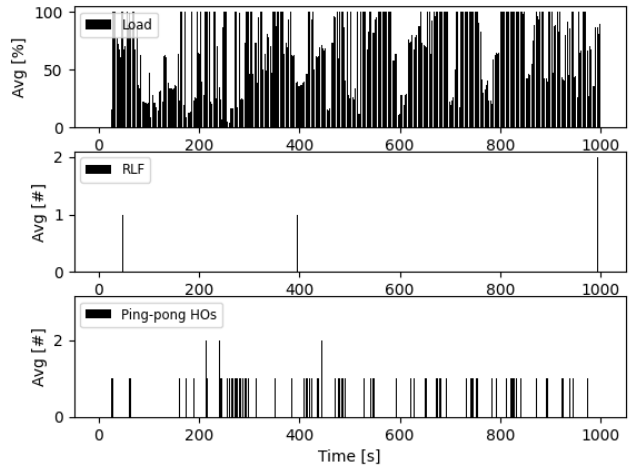


**Figure 3:** Conflicting MLB and MRO. The dataset presented in [104] has been used to illustrate the issue of ping-pong handovers (HOs) between two apps with conflicting objectives. For clarity, the KPIs of 2 gNBs are shown (the original dataset contains data for 19 gNBs). The MLB maintains the balance of the load on the gNBs (top plot), while the MRO maintains the RLFs close to zero (middle plot). The interaction of these xApps causes multiple ping-pong handovers as illustrated in the bottom plot.

Figure 3 exemplifies xApps causing conflicts using the well-known Self-Organized Networks (SON) functions, notably Mobility Load Balancing (MLB) and Mobility Robustness Optimization (MRO). MLB balances traffic distribution among cells to optimise network performance, while MRO ensures robust and stable links to UEs. Both apps change handover settings, resulting in ping-pong handovers [103].

Conflicting xApps, like other misconfiguration issues discussed in this study, can have a direct or indirect impact on the O-RAN. For direct impact, conflicts between RRM choices made by different xApps result in poor performance and network instability. In terms of indirect impact, the lack of a conflict resolution system for xApps exposes the O-RAN system to security threats. That is, rogue xApps might use this condition to launch a DoS attack using competing RRM options.

It is important to highlight that solutions designed to address conflicts between applications in the context of SON in 4G may not be directly applicable to O-RAN. Specifically, in the approaches involving collaboratively optimising and distributing resources [105] or team learning [81], the underlying assumption is that all applications are developed by a single vendor with a comprehensive understanding of the interactions among applications and RAN elements. However, as previously stated, this scenario may not apply to O-RAN. Therefore, a detection of xApp conflicts customized for O-RAN is required.

**Table 5**
Model of conflicts between xApps. $op(P_A)$: operation (change, modification) on the set of parameters $P_A$.

| | xApp A | xApp B | Direct conflict | Indirect conflict | Implicit conflict |
|---|---|---|---|---|---|
| Set parameters | $P_A$ | $P_B$ | $P_A = P_B$ | $P_A \neq P_B$ | $P_A \neq P_B$ |
| System impact | $P_A \rightarrow i_A$ | $P_B \rightarrow I_B$ | $op(P_A) \cap op(P_B) \rightarrow I_S$<br>$I_S = I_A = P_B$ | $op(P_A) \cap op(P_B) \rightarrow I_S$<br>$I_S = I_A = P_B$ | $op(P_A) \cap op(P_B) \rightarrow I_S$<br>$I_S \neq I_A$ and $I_S \neq P_B$ |
| Observation | | | It's known (a priori) which xApps caused $I_S$. | It's known (a priori) which xApps caused $I_S$. | The xApps causing $I_S$ are unknown (a priori). |
| Detection | | | Identify xApps involved in the recent actions (logs). | Identify xApps involved in the recent actions (logs). | Use AI/ML, e.g., MDP, to identify the xApps involved. |
| Example | | | Firewall rules:<br>$P_A$ = {Allow/Deny UE1}<br>$P_B$ = {Allow/Deny UE1}<br>$I_S$ = Granting UE1 | MRO and MLB:<br>$P_A = \{H, TTT\}$<br>$P_B = \{CIO\}$<br>$I_S$ = Handover boundary | Non-explainable AI/ML-based xApps:<br>$I_S$ = Delayed impact |

#### *4.3.2. Detection approach*

The first step in detecting this misconfiguration issue is to comprehend its nature and construct a model. In this regard, the O-RAN WG3 identified three types of conflicts that may arise in O-RAN: direct, indirect, and implicit [34]. Table 5 illustrates a basic model for these conflicts. In direct conflicts, two (or more) xApps, xApp A and xApp B, attempt to operate on the same set of parameters $P_A$ and $P_B$ (i.e., $P_A = P_B$), impacting the same system functions ($I_S$). Different parameters are changed in indirect conflicts (i.e., $P_A \neq P_B$), yet the impact on the same system functions is represented in the system. Finally, in implicit conflicts, different parameters (i.e., $P_A \neq P_B$) are operated on and different system functions are influenced. Particularly, implicit conflicts are challenging to solve since the xApps causing the system impact are not known a priori.

The next step is to design a system that detects the conflicts presented in Table 5. In this context, while recent efforts have contributed to potential conflict detection and mitigation frameworks inside the O-RAN system [104], these efforts have been focused on certain conflict types, and further improvements are necessary to prevent suboptimal outcomes. In addition, a critical concern in the design of conflict detection is to provide a generalised detection solution for the three categories of conflicts, if possible.

Figure 4 presents our proposed framework that helps identify conflicts across xApps, based on the standard Near-RT RIC architecture [34] and earlier research [104]. Note that the automated mitigation of the detected conflicts is also considered. The conflict detection and mitigation functionalities in this framework are implemented as xApps, notably CD xApp and CM xApp. Furthermore, this framework uses other xApps, particularly KPIMON xApp and AD xApp [39], and establishes a new network information database, referred to as xNIB, to store the operations of the xApps.

In the method outlined in Figure 5, both the KPIs of the E2 nodes and the activities of the xApps are monitored. It should be noted that the KPIs are dependent on the use case of the xApps. For example, the mean load of the base station, the number of call blockages, the number of radio connection failures, and the number of handovers, may be monitored to discover conflicts between MLB and MRO xApps [104]. The anomaly detection (AD) xApp evaluates the variability of the KPIs collected by the KPIMON xApp. If a considerable drop in system performance is noticed, the system investigates the actions of xApps (consults the xNIB) to determine the source of the dispute.

In cases of implicit conflicts, more advanced correlation processes, such as Markov Decision Problem (MDP) and Bayesian models, may be required to identify which xApps are generating the conflicts. Once the conflicting xApps have been found, they may be blocked directly based on priority. However, as seen in [104], these strategies may provide suboptimal outcomes. As a result, Reinforcement Learning (RL) can be used to learn the best way to assign priority in order to resolve conflicts and maximise system efficiency. Note that due to the complexities of O-RAN management, at least three components in Figure 5 use AI/ML.

Note that at the time of writing this document, none of the existing open-source or commercial Near-RT RICs have incorporated a conflict detection and mitigation solution. We consider that experimenting with this approach of conflict detection among xApps in an O-RAN testbed could represent a significant milestone in the field of misconfiguration solutions. This will promote the development of multi-vendor xApps for optimizing RAN operations while also guaranteeing reliability and minimizing outages. However, conducting such testing is beyond the scope of this analytical study.

### 4.4. Summary and insights

Although we examined how AI/ML can be used to detect misconfigurations using various detection methodologies, we emphasize that not all misconfiguration issues necessitate the use of AI/ML-based detection approaches. Nonetheless, the introduction of AI/ML can help simplify the analysis of

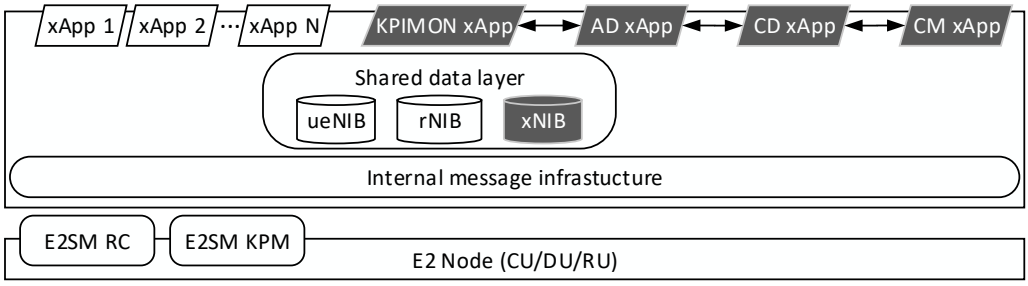


**Figure 4:** Managing conflicts between xApps: detection and mitigation using AI/ML techniques. Shaded components have been incorporated into the original Near-RT RIC architecture of the O-RAN WG3 [34], which include the information database for xApp actions (xNIB) and the xApps KPIMON, AD, CD, and CM.

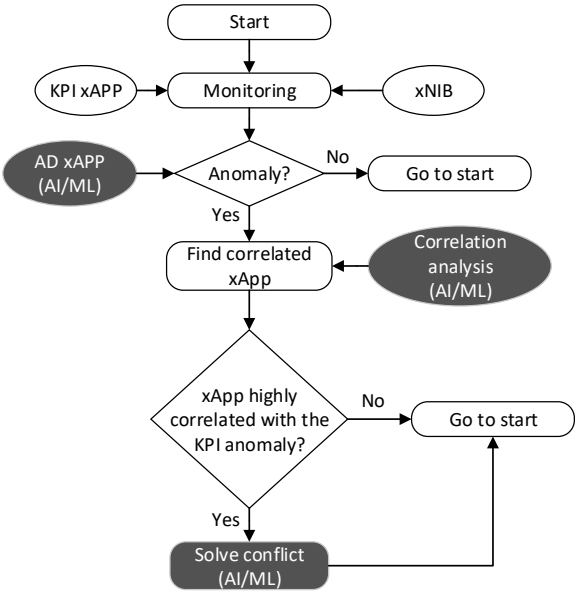


**Figure 5:** Method for detection and mitigation of conflict xApps. Shaded components may require AI/ML techniques.

the metrics captured in O-RAN deployments with a large number of components, applications, and amounts of traffic.

There are a number of commercially available tools (e.g., [106], [107], and [108]) that offer misconfiguration detection. Their documentation indicates the use of both passive and active monitoring to identify integration and security function misconfigurations. Detection tools for the other misconfiguration issues have yet to be developed. Those associated with AI/ML rely on specific application use-cases.

Furthermore, three types of conflicts are considered in the strategy illustrated in the case study. However, although there are several examples of xApps that cause direct and indirect conflicts, to the best of our knowledge, no examples of implicit conflicts have been published. These conflicts are predicted to arise when more AI/ML-powered xApps are added to the O-RAN, particularly if the xApps utilise complex DNNs (non-explainable AI/ML).

The biggest challenge in studying the detection of misconfiguration is that O-RAN technology is still in its early phases of development, making it difficult to evaluate misconfiguration issues in real deployments. Existing experimental testbeds are rather simple. For example, they include just the Near-RT RIC but not the non-RT RIC. As a result, datasets relating to O-RAN misconfigurations are unavailable. Additional efforts are required to produce these materials, allowing the research community to analyze O-RAN misconfigurations and to suggest and test mitigation methods.

## 5. Conclusion

O-RAN characteristics such as disaggregation, openness, and intelligence provide exciting opportunities for innovation in 5G and 6G networks. However, as illustrated in this study, these characteristics may cause misconfiguration issues that can significantly impact on the security and performance of the system.

As the O-RAN develops, certain methods for detecting misconfigurations associated with system integration and operation are emerging. However, use case-specific misconfiguration problems have yet to be explored. For instance, most distributed AI/ML implementations (model sharing, model splitting, and federated learning) are yet to be validated. In this work, we have highlighted the AI/ML-related misconfiguration issues that must be addressed so that the benefit of intelligence in the O-RAN is realised, rather than the intelligence becoming a limiting factor or a source of exploitation.

## Acknowledgments

This work is supported through the NICYBER2025 programme funded by Innovate UK. The ORANSecAI project is a collaboration with Ampliphae. The views expressed are those of the authors and do not necessarily represent the project or the funding agency.

## References


[1] K. Takiishi, "Hype Cycle for CSP Networks Infrastructure 2023- ID G00791652," Available at https://www.gartner.com (2023/09/18), Gartner, Tech. Rep., 2023.

[2] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "AI and 6G Security: Opportunities and Challenges," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 616–621.

[3] IEEE, "IEEE Standard for Packet -based Fronthaul Transport Network," Available at https://standards.ieee.org/ (2023/10/10), IEEE Standards Association, Tech. Rep., 2019.

[4] P. H. Masur, J. H. Reed, and N. K. Tripathi, "Artificial Intelligence in Open-Radio Access Network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 9, pp. 6–15, 2022.

[5] 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*, vol. 25, no. 2, pp. 1376–1411, 2023.

[6] S. Soltani, M. Shojafar, R. Taheri, and R. Tafazolli, "Can Open and AI-Enabled 6G RAN Be Secured?" *IEEE Consumer Electronics Magazine*, vol. 11, no. 6, pp. 11–12, 2022.

[7] C. Yeh, Y.-S. Choi, Y.-J. Ko, and I.-G. Kim, "Standardization and technology trends of artificial intelligence for mobile systems," *Computer Communications*, vol. 213, pp. 169–178, 2024.

[8] A. Giannopoulos, S. Spantideas, N. Kapsalis, P. Gkonis, L. Sarakis, C. Capsalis, M. Vecchio, and P. Trakadas, "Supporting Intelligence in Disaggregated Open Radio Access Networks: Architectural Principles, AI/ML Workflow, and Use Cases," *IEEE Access*, vol. 10, pp. 39 580–39 595, 2022.

[9] A. Johnson, K. Dempsey, R. Ross, S. Gupta, and D. Bailey, "Guide for Security-Focused Configuration Management of Information Systems - NIST SP 800-128," Available at https://doi.org/10.6028/NIST.SP.800-128 (2023/12/11), National Institute of Standards and Technology (NIST), Tech. Rep., 2011.

[10] M. Cook, S. Quinn, D. Waltermire, and D. Prisaca, "Security Content Automation Protocol (SCAP) Version 1.2 Validation Program Test Requirements - NISTIR 7511 Rev. 4," Available at http://dx.doi.org/10.6028/NIST.IR.7511r4 (2023/12/11), National Institute of Standards and Technology (NIST), Tech. Rep., 2016.

[11] M. Zoure, T. Ahmed, and L. Réveillère, "Network Services Anomalies in NFV: Survey, Taxonomy, and Verification Methods," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1567–1584, 2022.

[12] Mavenir, "SECURITY IN OPEN RAN WHITE PAPER," Available at https://www.mavenir.com (2023/09/18), Mavenir, Tech. Rep., 2021.

[13] P. technologies, "5G Security Issues," Available at https://www.gsma.com/membership/resources/positive-technologies-5g-security-issues/ (2023/09/09), Positive technologies, Tech. Rep., 2019.

[14] J. Zhang, R. Piskac, E. Zhai, and T. Xu, "Static Detection of Silent Misconfigurations with Deep Interaction Analysis," *Proc. ACM Program. Lang.*, vol. 5, no. OOPSLA, oct 2021.

[15] K. Takiishi, "Magic Quadrant for 5G Network Infrastructure for Communications Service Providers- ID G00767247," Available at https://www.gartner.com (2023/09/18), Gartner, Tech. Rep., 2023.

[16] D. Mimran, R. Bitton, Y. Kfir, E. Klevansky, O. Brodt, H. Lehmann, Y. Elovici, and A. Shabtai, "Security of Open Radio Access Networks," *Computers & Security*, vol. 122, p. 102890, 2022.

[17] M. Liyanage, A. Braeken, S. Shahabuddin, and P. Ranaweera, "Open RAN security: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 214, p. 103621, 2023.

[18] O-RAN, "O-RAN Working Group 11 (Security Working Group) O-RAN Security Threat Modeling and Remediation Analysis," Available at https://www.o-ran.org/ (2023/10/21), O-RAN Alliance, Tech. Rep., 2023.

[19] ——, "O-RAN Working Group 11 (Security Working Group) Security Protocols Specifications," Available at https://www.o-ran.org/ (2023/10/10), O-RAN Alliance, Tech. Rep., 2023.

[20] ——, "O-RAN Working Group 11 (Security Working Group) Security Requirements Specifications," Available at https://www.o-ran.org/ (2023/10/10), O-RAN Alliance, Tech. Rep., 2023.

[21] 3GPP, "Technical Specification Group Services and System Aspects; Study on the security aspects of Artificial Intelligence (AI)/Machine Learning (ML) for the Next Generation Radio Access Network (NG-RAN) (Release 18)," Available at https://www.3gpp.org/specifications (2023/10/21), 3rd Generation Partnership Project, Tech. Rep., 2023.

[22] NIS, "Report on the cybersecurity of Open RAN," Available at https://digital-strategy.ec.europa.eu/en (2023/12/11), NIS Cooperation Group - European Commission and ENISA, Tech. Rep., 2023.

[23] T. I. Project, "OPEN RAN TECHNICAL PRIORITIES: Focus on Security," Available at https://telecominfraproject.com/openran-mou-group/ (2023/09/09), Telecom Infra Project, Tech. Rep., 2023.

[24] H. Bogucka, P. Kryszkiewicz, M. Hoffmann, and M. Wasilewska, "The O-RAN Whitepaper 2023 - Security in O-RAN," Available at https://rimedolabs.com/o-ran/ (2023/12/11), Rimedo Labs, Tech. Rep., 2023.

[25] Ericsson, "Security considerations of Open RAN: Ensuring network radio systems are open, interoperable, and secure by design," Available at https://www.ericsson.com/ (2023/09/09), Ericsson, Tech. Rep., 2020.

[26] Rakuten, "The Definitive Guide to Open RAN Security," Available at https://symphony.rakuten.com/ (2023/12/11), Rakuten Symphony, Tech. Rep., 2022.

[27] VMware, "Security for Open RAN Architectures in 5G Telco Clouds - Protecting Open Radio Access Networks with Automation and Zero-Trust Architectures," Available at https://telco.vmware.com/ (2023/12/11), VMware, Tech. Rep., 2021.

[28] NEC, "Open RAN security examined - How open, interoperable network design facilitates security improvements," Available at https://www.nec.com/ (2023/12/11), NEC, Tech. Rep., 2022.

[29] M. Korolov, "Top 5 security risks of Open RAN," 2022. [Online]. Available: https://www.csoonline.com/article/573419/top-5-security-risks-of-open-ran.html

[30] A. S. da Silva and A. Schaeffer-Filho, "ARMOR: An Architecture for Diagnosis and Remediation of Network Misconfigurations," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[31] M. Mushi and R. Dutta, "Designing for Proactive Network Configuration Analysis," *Journal of Systemics, Cybernetics and Informatics*, vol. 17, no. 1, pp. 221–239, 2019.

[32] O-RAN, "O-RAN Working Group 1 (Use Cases and Overall Architecture) O-RAN Architecture Description (O-RAN.WG1.OAD-R003-v09.00)," Available at https://www.o-ran.org/ (2023/09/09), O-RAN Alliance, Tech. Rep., 2023.

[33] S. Sirotkin, *5G Radio Access Network Architecture: The Dark Side of 5G*. John Wiley & Sons, 2020.

[34] O-RAN, "O-RAN Working Group 3 (Near-Real-time RAN Intelligent Controller and E2 Interface Workgroup) (O-RAN.WG3.RICARCH-R003-v04.00)," Available at https://www.o-ran.org/ (2023/11/23), O-RAN Alliance, Tech. Rep., 2023.

[35] X. Lin, "Artificial Intelligence in 3GPP 5G-Advanced: A Survey," Available at https://www.comsoc.org/publications/ctn/artificial-intelligence-3gpp-5g-advanced-survey (2023/10/21), IEEE Communications Society, Tech. Rep., 2023.

[36] 3GPP, "Technical Specification Group Services and System Aspects; Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS (Release 18)," Available at https://www.3gpp.org/specifications (2023/10/21), 3rd Generation Partnership Project, Tech. Rep., 2023.

[37] ——, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Study on enhancement for Data Collection for NR and EN-

DC (Release 17)," Available at https://www.3gpp.org/specifications (2023/10/21), 3rd Generation Partnership Project, Tech. Rep., 2022.

[38] O-RAN, "O-RAN Global PlugFest hosted by Deutsche Telekom, EANTC, EURECOM, Orange, Vodafone," 2023. [Online]. Available: https://plugfestvirtualshowcase.o-ran.org/2023/O-RAN_Global_PlugFest_hosted_by_Deutsche_Telekom_EANTC_EURECOM_Orange_Vodafone

[39] O-RAN-SC, "RIC Applications (RICAPP)," 2023. [Online]. Available: https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=1179662

[40] R. Labs, "Rimedo Labs Open RAN (O-RAN)," 2023. [Online]. Available: https://rimedolabs.com/o-ran/

[41] ONF, "RAN Energy Savings Demonstration at Fyuz 2023," Available at https://opennetworking.org/sustainable-5g/ (2023/10/21), Open Networking Foundation, Tech. Rep., 2023.

[42] Ericsson, "Ericsson RAN Energy Cockpit rApp," Available at https://www.ericsson.com/en/ran/intelligent-ran-automation/intelligent-automation-platform/rapps (2023/11/20), Ericsson, Tech. Rep., 2023.

[43] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour, and S. 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)*, 2021, pp. 936–941.

[44] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and Customized Intelligence for Traffic Steering in 5G Networks Using Open RAN Architectures," *arXiv preprint arXiv:2209.14171*, 2022.

[45] Z. Mahrez, M. B. Driss, E. Sabir, W. Saad, and E. Driouch, "Benchmarking of Anomaly Detection Techniques in O-RAN for Handover Optimization," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, 2023, pp. 119–125.

[46] R. Ntassah, G. M. Dell'Aera, and F. Granelli, "xApp for Traffic Steering and Load Balancing in the O-RAN Architecture," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 5259–5264.

[47] V. Kasuluru, L. Blanco, and E. Zeydan, "On the use of Probabilistic Forecasting for Network Analysis in Open RAN," in *2023 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2023, pp. 258–263.

[48] K. Boutiba, M. Bagaa, and A. Ksentini, "On enabling 5G Dynamic TDD by leveraging Deep Reinforcement Learning and O-RAN," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–3.

[49] S. Téral, "RIC as the Next Generation SON for Open RAN and More – May 2021," Available at https://www.mavenir.com/resources/ric-as-the-next-generation-son-for-open-ran-and-more/ (2023/11/20), Mavenir and LightCounting, Tech. Rep., 2021.

[50] Ericsson, "Ericsson Performance Diagnostics," Available at https://www.ericsson.com/en/ran/intelligent-ran-automation/intelligent-automation-platform/rapps (2023/11/20), Ericsson, Tech. Rep., 2023.

[51] P. Kryszkiewicz and M. Hoffmann, "Open RAN for detection of a jamming attack in a 5G network," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–2.

[52] M. Hoffmann and P. Kryszkiewicz, "Signaling Storm Detection in IIoT Network based on the Open RAN Architecture," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–2.

[53] J.-H. Huang, S.-M. Cheng, R. Kaliski, and C.-F. Hung, "Developing xApps for Rogue Base Station Detection in SDR-Enabled O-RAN," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[54] D. Johnson, D. Maas, and J. Van Der Merwe, "NexRAN: Closed-Loop RAN Slicing in POWDER -A Top-to-Bottom Open-Source Open-RAN Use Case," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, ser. WiNTECH '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 17–23.

[55] S.-P. Yeh, S. Bhattacharya, R. Sharma, and H. Moustafa, "Deep Learning for Intelligent and Automated Network Slicing in 5G Open RAN (ORAN) Deployment," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 64–70, 2024.

[56] J. S. Mallu, J. F. Santos, A. P. da Silva, P. Sethi, V. Radhakrishnan, and L. DaSilva, "AI/ML Data-driven Control Loop for Managing O-RAN SDR-based RANs," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–2.

[57] R. Wiebusch, N. A. Wagner, D. Overbeck, F. Kurtz, and C. Wietfeld, "Towards Open 6G: Experimental O-RAN Framework for Predictive Uplink Slicing," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 4834–4839.

[58] Tsampazi, Maria and D'Oro, Salvatore and Polese, Michele and Bonati, Leonardo and Poitau, Gwenael and Healy, Michael and Melodia, Tommaso, "A Comparative Analysis of Deep Reinforcement Learning-based xApps in O-RAN," in *Proceedings of IEEE GLOBECOM*, Kuala Lumpur, Malaysia, December 2023.

[59] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Federated Deep Reinforcement Learning for Resource Allocation in O-RAN Slicing," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 958–963.

[60] Fijitsu, "A brief look at O-RAN Security (White paper)," Available at https://www.fujitsu.com/ (2023/10/10), Fijitsu, Tech. Rep., 2022.

[61] J. Groen, S. DOro, U. Demir, L. Bonati, M. Polese, T. Melodia, and K. Chowdhury, "Implementing and Evaluating Security in O-RAN: Interfaces, Intelligence, and Platforms," *arXiv preprint arXiv:2304.11125*, 2023.

[62] D. Dik and M. S. Berger, "Transport Security Considerations for the Open-RAN Fronthaul," in *2021 IEEE 4th 5G World Forum (5GWF)*, 2021, pp. 253–258.

[63] C. Lipps, A. Tjabben, M. Rüb, J. Herbst, S. P. Sanon, R. Reddy, Y. Munoz, and H. D. Schotten, "Designing Security for the Sixth Generation: About Necessity, Concepts and Opportunities," in *European Conference on Cyber Warfare and Security*, vol. 22, no. 1, 2023, pp. 267–275.

[64] R. Harrilal-Parchment, I. F. Pujol, and K. Akkaya, "Performance Evaluation of Quantum-Resistant Open Fronthaul Communications in 5G," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[65] P. Porambage, J. Pinola, Y. Rumesh, C. Tao, and J. Huusko, "XcARet: XAI based Green Security Architecture for Resilient Open Radio Access Networks in 6G," in *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2023, pp. 699–704.

[66] M. Neves, B. Huffaker, K. Levchenko, and M. Barcellos, "Dynamic Property Enforcement in Programmable Data Planes," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1540–1552, 2021.

[67] J. Wang, H. Qi, Y. He, W. Li, K. Li, and X. Zhou, "FlowTracer: An Effective Flow Trajectory Detection Solution Based on Probabilistic Packet Tagging in SDN-Enabled Networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1884–1898, 2019.

[68] W. Jiang, H. Han, M. He, and W. Gu, "ML-based pre-deployment SDN performance prediction with neural network boosting regression," *Expert Systems with Applications*, vol. 241, p. 122774, 2024.

[69] W. Saied, F. Jaidi, and A. Bouhoula, "A Comprehensive Solution for the Analysis, Validation and Optimization of SDN Data-Plane Configurations," in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020, pp. 1–7.

[70] M. Mekki, N. Toumi, and A. Ksentini, "Microservices Configurations and the Impact on the Performance in Cloud Native Environments," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, 2022, pp. 239–244.

[71] A. Huff, M. Hiltunen, and E. P. Duarte, "RFT: Scalable and Fault-Tolerant Microservices for the O-RAN Control Plane," in *2021 IFIP/IEEE International Symposium on Integrated Network Man-*

*agement (IM)*, 2021, pp. 402–409.

[72] S. Ramanathan, K. Kondepu, M. Razo, M. Tacca, L. Valcarenghi, and A. Fumagalli, "Live Migration of Virtual Machine and Container Based Mobile Core Network Components: A Comprehensive Study," *IEEE Access*, vol. 9, pp. 105 082–105 100, 2021.

[73] A. Oqaily, Y. Jarraya, L. Wang, M. Pourzandi, and S. Majumdar, "MLFM: Machine Learning Meets Formal Method for Faster Identification of Security Breaches in Network Functions Virtualization (NFV)," in *Computer Security – ESORICS 2022*. Cham: Springer Nature Switzerland, 2022, pp. 466–489.

[74] H. Wen, P. Porras, V. Yegneswaran, and Z. Lin, "A Fine-Grained Telemetry Stream for Security Services in 5G Open Radio Access Networks," in *Proceedings of the 1st International Workshop on Emerging Topics in Wireless*, ser. EmergingWireless '22. Association for Computing Machinery, 2022, p. 18–23.

[75] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Network Automation through Orchestrated Intelligence in the Open RAN," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 270–279.

[76] Telenet, "Risk Aware: Key issues and challenges in deploying open RAN," 2023. [Online]. Available: https://tele.net.in/risk-aware-key-issues-and-challenges-in-deploying-open-ran/

[77] 3GPP, "Technical Specification Group Services and System Aspects; Catalogue of general security assurance requirements (Release 18)," Available at https://www.3gpp.org/specifications (2023/10/10), 3rd Generation Partnership Project, Tech. Rep., 2023.

[78] H. Pan, Z. Li, P. Zhang, P. Cui, K. Salamatian, and G. Xie, "Misconfiguration-Free Compositional SDN for Cloud Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2484–2499, 2023.

[79] H. Kermabon-Bobinnec, M. Gholipourchoubeh, S. Bagheri, S. Majumdar, Y. Jarraya, M. Pourzandi, and L. Wang, "ProSPEC: Proactive Security Policy Enforcement for Containers," in *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 155–166.

[80] G. Li, H. Zhou, B. Feng, G. Li, H. Zhang, and T. Hu, "Rule Anomaly-Free Mechanism of Security Function Chaining in 5G," *IEEE Access*, vol. 6, pp. 13 653–13 662, 2018.

[81] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Team Learning-Based Resource Allocation for Open Radio Access Network (O-RAN)," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 4938–4943.

[82] S. Lakshmanan Thirunavukkarasu, M. Zhang, A. Oqaily, G. S. Chawla, L. Wang, M. Pourzandi, and M. Debbabi, "Modeling NFV Deployment to Identify the Cross-Level Inconsistency Vulnerabilities," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2019, pp. 167–174.

[83] Z. Kotulski, T. W. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, T. Osko, and J.-P. Wary, "Towards constructive approach to end-to-end slice isolation in 5G networks," *EURASIP Journal on Information Security*, vol. 2018, pp. 1–23, 2018.

[84] B. Hughes, S. Bothe, H. Farooq, and A. Imran, "Generative Adversarial Learning for Machine Learning empowered Self Organizing 5G Networks," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 282–286.

[85] H. K. Yea-Sul Kim, Ye-Eun Kim, "A Model Training Method for DDoS Detection Using CTGAN under 5GC Traffic," *Computer Systems Science and Engineering*, vol. 47, no. 1, pp. 1125–1147, 2023.

[86] M. Abdelaty, S. Scott-Hayward, R. Doriguzzi-Corin, and D. Siracusa, "GADoT: GAN-based Adversarial Training for Robust DDoS Attack Detection," in *2021 IEEE Conference on Communications and Network Security (CNS)*, 2021, pp. 119–127.

[87] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del Rincón, and D. Siracusa, "Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, 2020.

[88] ENISA, "Securing machine learning algorithms," Available at https://www.enisa.europa.eu/ (2023/10/23), European Union Agency for Cybersecurity., Tech. Rep., 2021.

[89] G. Apruzzese, R. Vladimirov, A. Tastemirova, and P. Laskov, "Wild Networks: Exposure of 5G Network Infrastructures to Adversarial Examples," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5312–5332, 2022.

[90] C. Benzaïd and T. Taleb, "AI for Beyond 5G Networks: A Cyber-Security Defense or Offense Enabler?" *IEEE Network*, vol. 34, no. 6, pp. 140–147, 2020.

[91] J. Śliwa and M. Suchański, "Security threats and countermeasures in military 5G systems," in *2022 24th International Microwave and Radar Conference (MIKON)*, 2022, pp. 1–6.

[92] M. Usama, I. Ilahi, J. Qadir, R. N. Mitra, and M. K. Marina, "Examining Machine Learning for 5G and Beyond Through an Adversarial Lens," *IEEE Internet Computing*, vol. 25, no. 2, pp. 26–34, 2021.

[93] K. Davaslioglu and Y. E. Sagduyu, "Trojan Attacks on Wireless Signal Classification with Adversarial Machine Learning," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.

[94] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep Learning for Launching and Mitigating Wireless Jamming Attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2–14, 2019.

[95] A. Omara and B. Kantarci, "Adversarial Machine Learning-Based Anticipation of Threats Against Vehicle-to-Microgrid Services," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 1844–1849.

[96] N. Naik, B. Kim, K. Chowdhury, and V. Shah, "Experimental Study of Adversarial Attacks on ML-based xApps in O-RAN," *arXiv preprint arXiv:2309.03844*, 2023.

[97] W. Guo, "Explainable Artificial Intelligence for 6G: Improving Trust between Human and Machine," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39–45, 2020.

[98] C. I. Nwakanma, L. A. C. Ahakonye, J. N. Njoku, J. C. Odirichukwu, S. A. Okolie, C. Uzondu, C. C. Ndubuisi Nweke, and D.-S. Kim, "Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review," *Applied Sciences*, vol. 13, no. 3, 2023.

[99] IETF, "SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol- IETF RFC 6668," 2012. [Online]. Available: https://tools.ietf.org/html/rfc6668

[100] K. Park, S. Sung, H. Kim, and J. il Jung, "Technology trends and challenges in SDN and service assurance for end-to-end network slicing," *Computer Networks*, vol. 234, p. 109908, 2023.

[101] M. Oqaily, Y. Jarraya, M. Mohammady, S. Majumdar, M. Pourzandi, L. Wang, and M. Debbabi, "SegGuard: Segmentation-Based Anonymization of Network Data in Clouds for Privacy-Preserving Security Auditing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2486–2505, 2021.

[102] P. Li, J. Thomas, X. Wang, A. Khalil, A. Ahmad, R. Inacio, S. Kapoor, A. Parekh, A. Doufexi, A. Shojaeifard, and R. J. Piechocki, "RLOps: Development Life-Cycle of Reinforcement Learning Aided Open RAN," *IEEE Access*, vol. 10, pp. 113 808–113 826, 2022.

[103] Z. Liu, P. Hong, K. Xue, and M. Peng, "Conflict Avoidance between Mobility Robustness Optimization and Mobility Load Balancing," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, 2010, pp. 1–5.

[104] C. Adamczyk and A. Kliks, "Conflict Mitigation Framework and Conflict Detection in O-RAN Near-RT RIC," *IEEE Communications Magazine*, pp. 1–7, 2023.

[105] M. Huang and J. Chen, "A Conflict Avoidance Scheme between Mobility Load Balancing and Mobility Robustness Optimization in Self-Organizing Networks," *Wireless Networks*, vol. 24, no. 1, p. 271–281, jan 2018.

[106] Viavi, "TeraVM RIC Test," 2024. [Online]. Available: https://www.viavisolutions.com/en-uk/products/teravm-ric-test

[107] Keysight, "P8828S RICtest — RAN Intelligent Controller Test Solutions," 2024. [Online]. Available: https://www.keysight.com/us/en/product/P8828S/rictest-ran-intelligent-controller-test-solutions.html

[108] Spirent, "Spirent O-RAN Test Solutions," 2024. [Online]. Available: https://www.spirent.com/assets/u/brief-spirent-o-ran-test-solutions