Enhancing URLLC Resiliency in Open RAN Access Networks via AI and Intelligent RIC Architectures

by **Ava Azadeh Arnaz**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

supervised by Professor Justin Lipman and Professor Mehran Abolhasan

School of Electrical and Data Engineering

Faculty of Engineering and IT

University of Technology Sydney

October 14, 2025

Certificate of Original Authorship

I, Ava Azadeh Arnaz, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220

Signature:	Production Note: Signature removed prior to publication.
Date:	October 14, 2025

Abstract

The emergence of Ultra-Reliable Low-Latency Communication (URLLC) applications has introduced unprecedented demands on wireless networks. Open Radio Access Network (Open RAN) architecture offers promising capabilities through its Near Real-Time RAN Intelligent Controller (Near-RT RIC), yet critical challenges remain in supporting URLLC reliability requirements. This thesis presents innovative solutions for enhancing Near-RT RIC for URLLC applications.

First, we develop a multi-objective optimization framework that enhances RAN control functions with Pareto-optimal decision making. Traditional approaches prioritize single objectives and handle failures through costly retries. We created HORLA (HandOver Reinforcement Learning Application) to simultaneously optimize multiple parameters for both reliability and performance. Experimental validation demonstrates a 40% reduction in handover failures compared to traditional approaches, while maintaining sub-second latency and reducing energy consumption by 57%. Results confirm that multi-objective controllers are essential for achieving necessary reliability within near-real-time constraints.

Second, we present a security study addressing vulnerabilities in AI-enabled Near-RT RIC systems through investigations of reward manipulation, last-layer distortion, and parameter tampering. Our work demonstrates how sophisticated attacks can compromise network performance while evading traditional monitoring systems.

Third, we propose PULSE (Predictive Ultra-reliable Low-latency System Engine), a Near-RT RIC xApp that redefines reliability solutions by incorporating semantic-aware processing. By extending Shannon's communication theory, PULSE leverages transformer-based understanding to reconstruct lost packets without retransmissions. Our implementation achieves 100% prediction accuracy for up to 10% packet loss and 93.96% accuracy with 10-50% loss, while delivering submillisecond processing times.

Finally, we introduce DANTE (Drone Adaptive Natural-to-Encoded Text Engine), a Near-RT RIC xApp that moves command standardization from endpoints to the centralized RAN edge. DANTE achieves 98.90% accuracy transforming natural language commands into standardized formats while maintaining strict latency requirements, improving efficiency and reliability across multiple devices.

These contributions establish a new paradigm for near-real-time applications' reliability in Open RAN. Our thesis demonstrates the necessity for innovative frameworks in Near-RT RIC that enhance reliability while maintaining strict latency requirements, providing a foundation for future research in intelligent wireless networks for next-generation URLLC applications.

Dedication

To innovation and the transformative power of ideas that push boundaries and create new possibilities.

Acknowledgements

I would like to express my sincere gratitude to my supervisors for their guidance and support throughout this research journey.

Ava Azadeh Arnaz

October 14, 2025

Sydney, Australia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Ultra Reliable Low Latency Communication (URLLC) use cases, including mission-critical communication, tactile internet, and autonomous vehicles, require reliability, resilience, and near-real-time latency thresholds. These requirements have challenged wireless communication infrastructure for decades. 3GPP added specifications to support ultra reliability and low latency in 5G. These features were originally added as separate requirements, but eventually in 2015, 3GPP specifications acknowledged ultra reliability and low latency as potential simultaneous requirements in some communications. These industrial movements, advised by research works, encouraged more research on URLLC applications.

This surge in URLLC related research coincides with major transformations in wireless network architectures. Over the past three decades, wireless communications, specifically Radio Access Network (RAN) architectures, have fundamentally transformed how society connects, works, and accesses information. The journey from voice-centric first-generation networks to today's digital systems illustrates remarkable technological progression. Research is now advancing toward sixth-generation (6G) technologies. With this advancement, the industry envisions new capabilities being integrated into societies, such as tactile internet and highly autonomous networks. In recent years, a new RAN architecture has emerged, called Open RAN. The O-RAN Alliance, founded in 2018, initiated and has been working on a global movement for designing and deploying the Open RAN architecture. Open RAN is designed on the foundation of 3GPP. This new architecture aims for an infrastructure that supports fast-paced innovation and lower

costs of building and deployment through a multi-vendor strategy. Furthermore, the proposed Open RAN architecture has dedicated components for AI applications.

Open RAN architectures include components dedicated to intelligent controllers, particularly the Near-Real-Time RAN Intelligent Controller (Near-RT RIC) framework. This presents an opportunity to explore novel theoretical approaches to network control and optimization, especially for URLLC use cases. The introduction of standardized near-real-time intelligence components in Open RAN establishes a unified and global paradigm for AI application deployment and execution. This standardization not only provides architectural clarity but also creates a foundation for focused theoretical research as the infrastructure matures. Previous research efforts in network intelligence explored diverse approaches regarding where AI applications should run within real-world network deployments. The standardization of intelligent controllers on RIC components has resolved this uncertainty. This architectural convergence now enables researchers to focus on exploring fundamental questions about network control and optimization.

1.2 Research Problems

3GPP has created directions for the industry to support URLLC applications. Open RAN has designed components dedicated to running near-real-time applications. The O-RAN specification for use cases has categorized some wireless communication controllers such as resource allocation, traffic steering, and Hand Over (HO) management as near-real-time use cases [1]. However, this thesis argues that several critical challenges remain unaddressed in the current literature and Open RAN specifications regarding URLLC use cases. First, existing use case specifications for near-RT RIC are not particularly URLLC oriented. Based on the Open RAN specifications, xApps are control applications that require near-real-time latency. But URLLC requirements are beyond low latency. The communication in URLLC should stay reliable while being near-real-time. However, the specification neither provides potential solutions nor emphasizes the ultra reliability required for URLLC applications for all use cases. Second, the security implications of introducing near-real-time intelligence in RAN architectures have not been thoroughly analyzed. Finally, the theoretical boundaries of RAN control need to be re-examined to accommodate emerging wireless communication needs. While wireless communication evolves, RAN methodologies to solve classic problems need to be reviewed. Furthermore, the RAN's

boundary of responsibilities need to be redefined and potentially revolutionized.
In summary, this thesis argues that Near-RT RIC holds great potential for URLLC applications. However, the current state of Near-RT RIC needs enhancements to make it a suitable infrastructure for URLLC and mission-critical use cases. This thesis addresses four critical gaps in the current Open RAN architecture, particularly concerning the Near-RT RIC (Near-Real-Time RAN Intelligent Controller) and its ability to support Ultra-Reliable Low-Latency Communication (URLLC) use cases. First, classic RAN responsibilities require enhancement to improve Near-RT RIC functionality for URLLC applications. Second, significant security vulnerabilities exist in the current Near-RT RIC specifications that may compromise URLLC AI applications. Third, conventional reliability methods in RAN systems have remained largely unchanged for decades despite evolving technological capabilities and requirements. Fourth, despite rapid advancements in edge computing and artificial intelligence, RAN's role in enhancing Quality of Experience (QoE) and Quality of Service (QoS) remains inadequately defined, particularly in the context of xApps deployment. To address these challenges, this thesis investigates the following research questions:

1. How can Near-RT RIC control functions be theoretically reformulated to support URLLC requirements?

2. Are Near-RT RIC security specifications sufficient to protect URLLC AI xApps? What are the security vulnerabilities and theoretical challenges introduced by AI xApps for URLLC use cases running on Near-RT RIC in Open RAN architectures?

3. Is there any classic reliability problem that can be redefined and enhanced using new technologies, particularly to support URLLC requirements?

4. Can RAN's responsibilities be expanded to enhance QoE and QoS for URLLC applications? How xApps can help with this redefinition of RAN?

1.2.1 Thesis Contributions

This thesis addresses the above-mentioned questions by developing novel theoretical frameworks for Near-RT RIC, validated through experimental analysis. To summarize, this thesis first examines potential enhancement opportunities in Near-RT RIC for URLLC applications. Upon establishing the feasibility of running AI xApps for URLLC, the research explores security vulnerabilities that could threaten these applications. Finally, after enhancing Near-RT

RIC functionalities to improve its reliability and resilience for URLLC applications, the thesis examines classic reliability solutions and expanding RAN boundaries to further enhance reliability and resilience in URLLC communication. The specific contributions of this Thesis are described below:

1. **HORLA (HandOver Reinforcement Learning Application):** This thesis proposed a multi-objective decision-making framework to enhance reliability of URLLC use cases. The framework proposes solutions that target multiple decision-making objectives simultaneously to reduce latency and improve the system's resilience and reliability. The thesis implemented one of Near-RT RIC use cases with the multi-objective decision making framework to experimentally validate the argument. The solution is implemented for Hand Over controllers and is called HORLA. HORLA directly addresses the first research question by verifying the comprehensive theoretical argument for URLLC related AI xApp design in Near-RT RIC platforms. The research begins with a mathematical formulation that unifies multiple objectives within low-latency, ultra-reliability, and resilience constraints, specifically combining reliability and signal strength requirements into a cohesive reinforcement learning model. HORLA establishes a methodology that can be adapted for various xApp implementations with multiple objectives to fulfill. The practical validation of this framework, achieving a 40% reduction in handover failures while maintaining stringent latency requirements, demonstrates both its theoretical soundness and real-world applicability. This thesis with HORLA implementation proposes para-optimal approaches rather than traditional single-objective optimization with failure recovery mechanisms. Para-optimal approaches are particularly well-suited for URLLC applications as they enhance reliability by simultaneously optimizing for both influential objectives, while maintaining the strict latency requirements - avoiding the additional delays typically introduced by traditional fail-and-retry mechanisms.

   HORLA not only addresses its core objective as explained above, but also extends beyond its primary goal to demonstrate how the framework and solution can save energy and enhance system security.

2. **AI xApp Vulnerability Research:** This contribution addresses the second research question by systematically exposing security vulnerabilities in AI-driven xApps within Near-RT RIC environments. Through rigorous experimental analysis of both software and hardware attack vectors, this research demonstrates how sophisticated adversaries

can compromise xApp integrity while circumventing traditional detection mechanisms. The findings reveal critical security gaps in current O-RAN specifications, particularly concerning AI applications supporting ultra-reliable low-latency communication use cases. By documenting how targeted manipulation of model parameters, reward functions, and computational resources can substantially degrade network performance without triggering conventional monitoring alerts, this work establishes the foundational evidence for developing specialized security frameworks tailored to the unique characteristics of AI components in Open RAN architectures. The research provides both theoretical insights and practical considerations for enhancing the resilience of next-generation wireless networks against emerging threat vectors.

3. **PULSE (Predictive Ultra-reliable Low-latency System Engine):** This thesis argues that to prepare the infrastructure for the large number of URLLC use cases in the near future, RAN methods to mitigate reliability problems need to be reviewed and potentially new methods to be introduced. PULSE addresses this argument and the third research question by establishing a theoretical framework that proposes an innovative packet loss framework particularly for URLLC use cases where reliability and low latency are simultaneously required. By extending Shannon's classical communication theory to incorporate semantic-aware processing, PULSE demonstrates how Near-RT RIC's packet recovery system can be enhanced to support emerging wireless paradigms. The framework introduces a novel method that combines transformer-based semantic understanding with traditional network control, achieving 100% prediction accuracy for up to 10% packet loss and 93.96% accuracy for 10-50% packet loss scenarios. Through the mathematical modeling of confidence-based decision mechanisms and the performance analysis against traditional approaches, this theoretical framework backed by the simulated experiment proposes a RAN transformation approach, enhancing reliability and resilience for URLLC use cases. The work further extends this framework to multi-robot coordination, demonstrating how leveraging Near-RT RIC capabilities and holistic knowledge can expand RAN network responsibilities and support complex URLLC applications. By consolidating edge device responsibilities within the RAN infrastructure, PULSE enhances decision-making, reliability and resilience through comprehensive device visibility, strengthens security by centralizing control away from potentially compromised edge devices, and optimizes energy efficiency across the network.

4. **DANTE (Drone Adaptive Natural-to-Encoded Text Engine):** This thesis addresses the fourth research question by demonstrating how Near-RT RIC responsibilities can be expanded beyond traditional RAN control to support emerging wireless paradigms. The framework establishes a theoretical foundation for semantic abstraction in URLLC communications, which provides flexibility and reliability to URLLC use cases. Moreover, this framework moves command standardization from edge devices or user equipment to the RAN infrastructure improving security and reducing overhead on URLLC endpoints. Through comparative analysis with rule-based approaches and comprehensive performance evaluation, DANTE achieves 98.90% accuracy in command standardization while maintaining sub-millisecond processing times. This novel approach not only transforms traditional command processing but also demonstrates how Near-RT RIC can assume additional responsibilities traditionally handled at edge devices and end users. By consolidating semantic communication within the RAN infrastructure, this thesis proposes a solution that enhances decision-making through centralized processing, strengthens security by eliminating reliance on edge devices, and optimizes system-wide efficiency. This work provides concrete evidence of how Near-RT RIC and consequently RAN control frameworks can be expanded to support emerging wireless communication needs while maintaining strict reliability and latency requirements.

1.3 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 3 proposes a multi-objective framework to enhance reliability of xApps and experimentally validates its theory with HORLA.

Chapter 4 provides a novel security analysis for Near-RT RIC AI applications to address vulnerabilities associated with future use cases.

Chapter 5 proposes innovation in classic reliability methods and proposes PULSE to advance Near-RT RIC capabilities and reliability.

Chapter 6 proposes an innovative approach to RAN responsibilities. This chapter introduces DANTE to enhance Near-RT RIC responsibilities through semantic communication.

Chapter 7 concludes the thesis, summarizing key contributions and suggesting future research

directions.

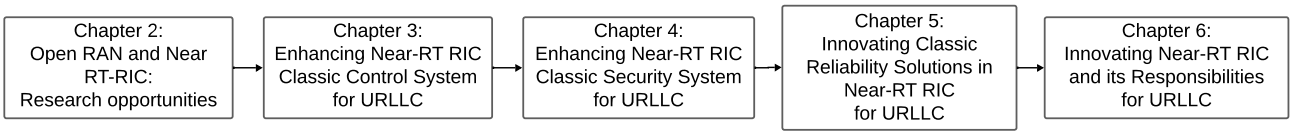


Figure 1.1: Thesis chapter organization and progression

Conclusion

This chapter has established the critical research context surrounding Ultra Reliable Low Latency Communication (URLLC) and the evolving Open RAN architecture. We have identified four key research problems in the current Open RAN architecture, particularly concerning the Near-RT RIC and its ability to support URLLC use cases. These include the need to enhance classic RAN responsibilities, address security vulnerabilities, modernize conventional reliability methods, and better define RAN's role in enhancing QoE and QoS.

To address these challenges, this thesis proposes four innovative solutions: HORLA, a multi-objective decision-making framework for handover control; a comprehensive study of security vulnerabilities for AI-enabled Near-RT RIC systems; PULSE, a predictive system for enhancing packet recovery in URLLC communications; and DANTE, a framework for expanding RAN responsibilities through semantic communication. These contributions collectively aim to transform Near-RT RIC capabilities to better support the stringent requirements of URLLC applications.

The next chapter provides a comprehensive review of relevant literature, focusing on Open RAN architecture and research opportunities.

Chapter 2

Literature Review

2.1 Introduction

The evolution of wireless communications has driven fundamental changes in network architecture as systems adapt to meet increasingly demanding requirements. Modern use cases like autonomous vehicles, remote surgery, and industrial automation demand unprecedented levels of reliability, latency, and flexibility from wireless networks. This chapter examines how Radio Access Networks (RAN) have evolved in response to these challenges, ultimately leading to Open RAN as a transformative architecture that enables the integration of artificial intelligence for network optimization and control.

The telecommunication Radio Access Network (RAN) forms a significant part of wireless communication and has evolved considerably in the past two decades. This thesis explores a new architecture of RAN named Open RAN and its role in the era of intelligent telecommunication. The RAN defines the logical group of components between the receiver and the core network in end-to-end telecommunication systems. These components communicate via interfaces. Each network generation has introduced new capabilities and possibilities to the public. As a result, the RAN evolved as a response to new requirements. Currently, the 5th Generation (5G) and Beyond 5th Generation (B5G) are in implementation and study. 5G and B5G have a much higher frequency than the previous generations. Therefore they are capable of transporting more data than the previous generations. However, high frequency limits signal attenuation through obstacles reducing non-line of sight coverage. 5G and B5G can enable more low-latency mission-critical and high data rate use cases despite challenges in coverage.

To achieve these requirements, the new RAN architecture should encourage a fast-paced and innovative approach in telecommunication. Considering the complexity and the latency of decision-making tasks in 5G and B5G, Artificial Intelligence (AI) will play a vital role in developing a new RAN architecture called "Open RAN". Consequently, AI is increasingly becoming relevant for 5G and B5G network deployment. Open RAN defines open interfaces to encourage a multi-vendor system. This intention is to introduce flexibility and boost innovation in a competitive market. Also, Open RAN has dedicated logical components for enabling intelligent applications that control RAN communications. This approach creates a global standard and framework for all 5G and B5G vendors and operators to implement AI solutions. Nevertheless, there are challenges to implementing this architecture in practice. For example, the number of components in the Open RAN and security or compatibility risks raised by this multi-vendor architecture has created research and engineering projects across different industries such as hardware, software, machine learning, and security. As a result, Open RAN has been the subject of significant research and standardization efforts over the last few years. This chapter is a comprehensive survey of Open RAN. This chapter's topics are focused on AI applications, the use of AI within Open RAN, deployment scenarios in artificial intelligence, and future opportunities in this area. The motivation for this work stems from the growing research interests in Open RAN and the impact it is expected to make for future wireless networks. Furthermore, the interaction of ML/AI with Open RAN introduces a tremendous opportunity to further enhance and optimise the performance of future wireless networks, in particular it opens up a pathway towards developing fully autonomous or self-driving networks. Hence, this chapter comprehensively surveys the current research and builds a pathway for future search directions over Open RAN. The main focus of this chapter is as follows:

- A holistic study of AI in telecommunication solutions, classified based on the Open RAN controllers classification is presented.
- ML deployment scenarios that are not covered in the standards are discussed.
- Future opportunities and challenges that the combination of AI and Open RAN in 5G and B5G can provide are presented.

The organization of this chapter is as follows. Section 2.2 provides a brief history of the RAN, while Section 2.3 introduces the Open RAN. Section 2.4 explores the proposed Open RAN architecture that is standardized. Section 2.5 explores the use of AI to solve problems within

Open RAN in telecommunications. Section 2.6 addresses an essential branch of machine learning: the deployment of machine learning pipelines in production, known as MLOps. Finally, Section 2.7 explores the opportunities and challenges for future research.

2.2 History of RAN

The end goal of any network infrastructure design is to provide seamless and secure communication between devices. Radio Access Network (RAN) is one of the critical concepts in telecommunication that facilitates this goal. Conceptually, RAN resides between a device such as a mobile phone, a computer, or any remotely controlled machine and the core network (CN)[2].

2.2.1 Networking generations

In [3] Henrik A. et al. describe the evolution of telecommunications networking known as "Networking Generations." The public started using the first generation of mobile phones (1G) in the early 1980s. Although in the early 1990s, the 2G Global System for Mobiles (GSM) telecommunication system created a pivot point in telecommunication, its functional architecture was static. Further, network functionalities were geographically localized, and all radio-related functionalities operated within the base station.

The next generation, 3G UMTS (Universal Mobile Telecommunication System) terrestrial RAN, split radio functionalities into two parts. One part was for functionalities such as transmission and reception, which were the responsibility of NodeB (a radio base station receiver in 3G). The other part included radio resource management and higher-layer RAN user processing in 3G running on RNC (Radio Network Controller). Splitting functionalities and running them on two separate parts created latency due to control processes between NodeB and RNC. But because RNC could execute resource allocation tasks faster than the previous generation, the overall latency in 3G became less than 2G.

Upgrading 3G to an LTE-advanced (Long-Term-Evolution-advanced) version led to 4G. 4G changed expectations on data transfer rate and security management. However, new use cases demand more and faster data transportation in wireless communication that exceeds the capacity of 4G. 5G and beyond 5G (B5G) are designed to meet these capacity requirements for future capacity requirements. The following section discusses the evolution of RAN from D-RAN to

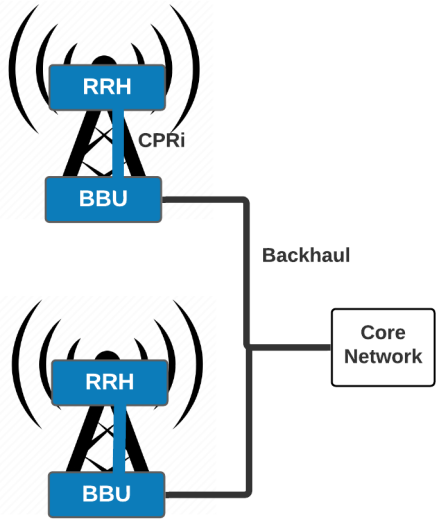


Figure 2.1: D-RAN

v-RAN, which led to the foundation of Open RAN.

2.2.2 D-RAN

Early generations of wireless networks had a Baseband Unit (BBU) and a Remote Radio Head (RRH) component, both physically located in Base Stations (BS). BBU and RRH were connected to the Radio Frequency (RF) antenna at the top of the tower through electrical cables. This design experienced RF signal loss. As a result, telecommunication experts designed Distributed RAN (D-RAN), in which BBU and RRH are separated. As Fig. 2.1 shows, in D-RAN, each Base Station(BS) includes a BBU and an RRH (Radio Remote Head), also called Remote Radio Unit (RRU). BBU connects to RRH through a Common Public Radio Interface (CPRI). BBU is responsible for baseband processing which includes processing calls and forwarding traffic. RRH is responsible for transmitting, receiving, and converting signals. Base stations are connected to the core network individually through a separated backhaul. One of the main limitations of this architecture is that the result of BBU processing can be shared only with the coupled RRH. The increase in demand for services disclosed other limitations of this architecture, such as low spectral efficiency, high cost of scaling this architecture, and inefficient use of resources.

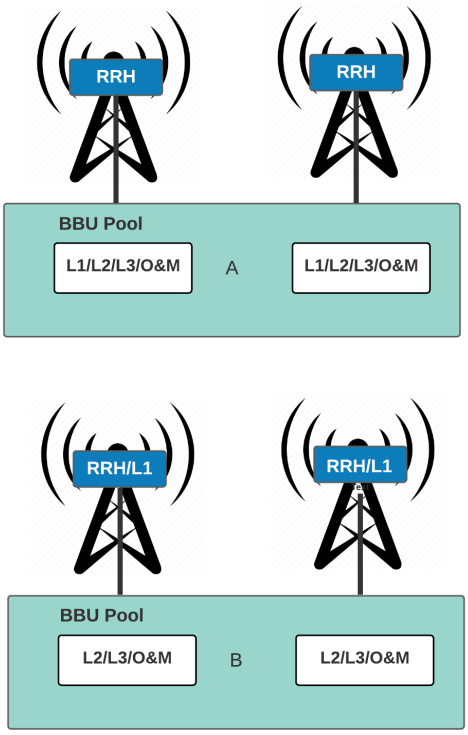


Figure 2.2: C-RAN. A: Fully Centralized, B: Partial Centralized

2.2.3 C-RAN

C-RAN (Cloud-RAN or Centralized Access Network) is a response to the limitations of D-RAN. It decouples RRUs and their corresponding BBUs. RRUs stay at the cell site in this architecture and connect to a centralized but shared and virtualized group of BBUs. Every RRU connects to a BBU pool via a fronthaul link. Each BBU pool can serve more than one RRU unit and connects to the Core network through a backhaul link. As shown in Fig. 2.2, C-RAN comes in two types: "Fully centralized" and "Partial centralized". The main difference between these two types is the functionalities related to layer one, such as sampling, modulation, resource block mapping, antenna mapping, and quantization. These functionalities happen in BBU pools in the fully centralized type, but they run on the RRU unit in the partially centralized type.

2.2.4 vRAN

Virtualized RAN (vRAN) is a revolutionary improvement in networking. vRAN is a disaggregated RAN architecture. In vRAN, the software is separated from the underlying hardware instead of running software on expensive proprietary hardware units. This modification

makes it possible to run networking functionalities dynamically and flexibly. It also reduces the cost of maintenance and operation because a modification of network functionalities does not require hardware-related amendments. Also, functions can run on common-off-the-shelf (COTS) hardware. Operators can use monitoring applications and manage load balancing and performance-related functionalities. Moreover, it creates an agile and scalable environment for Open RAN to achieve its goals.

2.3 Introduction of Open-RAN

In this section, we introduce Open RAN. We have divided this section into two parts. In the first part, we talk about the research history of Open RAN and examples of some published works on this topic. In the second part, we introduce communities and organizations in the industry that implement Open RAN for 5G and B5G telecommunication systems.

2.3.1 Open RAN in research

The idea of having a RAN with open interfaces is not very recent. In 2002, the Mobile Wireless Internet Forum IP in the RAN working group developed a version of Open RAN, in collaboration with researchers from Cisco Systems and DoCoMo Communication Lab. This collaborative team published a paper describing the need and requirements of Open RAN [4]. In that paper, the authors addressed the challenges of RAN architectures in scalability and reliability. For instance, the expansion of RAN is costly. Furthermore, the centralized control system becomes the *single point of failure*. The paper proposed an Open RAN architecture that included separate mobile node control functions; and supported multiple radio technologies, including 2G, 3G, Wireless LAN (Local Area Network), and any upcoming technologies in this area. The proposed architecture also supported the operation and administration of multi-vendor solutions, hence having open interfaces. In [4], the authors described the obstacles of implementing an Open RAN architecture, such as managing and orchestrating an Open RAN architecture. Existing RANs are not globally standardized, and different operators might have different administration procedures. As a result, making components from different vendors compatible with each other and the ecosystem might become an obstacle during the installation of a system. In addition, decommissioning a network generation and replacing it with a new one is costly and requires a significant amount of time. Operators need to upgrade their networking

generation alongside the running legacy network generations to avoid any negative impact on QoE for customers. A modular, multi-vendor architecture such as Open RAN should ensure that it can provide services to all current and future network generations such as 4G, 5G, and B5G.

Following on from [4], there has been a significant number of Open RAN-related publications. Virtualized Network (vRAN) created an environment to implement Open RAN architectures. Early research in Open RAN focused primarily on virtualization aspects. While [5] established important foundations for SDN/NFV integration in Open RAN, it didn't address the challenges of multi-vendor integration. This limitation was partially addressed by [6], which explored a multi-vendor ecosystem using open-source software. However, the practical deployment challenges remained unexplored until [7] examined xApp scalability and reliability in production environments. In [8] authors explored the role of AI in IoT (Internet of Things) and 5G within the context of Open RAN. This progression of research reveals a crucial gap: while individual technical components have been studied, a comprehensive framework for implementing AI applications in Open RAN while maintaining reliability and security requirements is still lacking. Research into Open RAN implementation strategies has evolved from basic architectural considerations to more specialized applications. While [9] provided a theoretical foundation for minimizing delays by co-locating core network and RAN components, their model's assumptions of same-datacenter deployment limit its applicability in distributed environments. Addressing different aspects of Open RAN deployment, [10] introduced blockchain technology to enhance trust and scalability. However, while their simulation results demonstrated improved security, the computational overhead of blockchain integration wasn't fully addressed in the context of latency-sensitive applications. The practical applications of Open RAN have been explored in emerging use cases. [11] demonstrated Open RAN's potential for drone communications, specifically focusing on high data rate streaming applications. While their work showed the architecture's flexibility, it didn't address the critical challenges of maintaining reliable connections with mobile aerial platforms. Building upon these implementation experiences, [12] provided a broader examination of deep learning integration in Open RAN, tackling multiple operational aspects including energy efficiency and security. However, their work, while comprehensive in scope, left open questions about how to balance these competing objectives in real-world deployments. In [13], the authors present an overview of Open RAN and its capabilities to solve potential networking problems. In [14], the author discusses the estimated cost of

Sample research	**Main Focus**
[4]	Initial proposal of Open RAN architecture addressing scalability and reliability challenges, supporting multiple radio technologies and multi-vendor solutions
[5]	Role of virtualized RAN and SDN/NFV in Open RAN implementation
[6]	Multi-vendor networking ecosystem using open-source software for Open RAN
[7]	Scalability and reliability analysis of xApps in production environments
[8]	Integration of AI in IoT and 5G within Open RAN context
[9]	Mathematical modeling for minimum delay problems in Open RAN systems
[10]	Integration of blockchain technology and network sharing in Open RAN
[11]	Open RAN architecture for high data rate cellular links in drone applications
[12]	Implementation of deep learning in Open RAN, focusing on energy efficiency and security
[13]	Overview of Open RAN capabilities and potential networking solutions
[14]	Cost analysis of network implementation and economic benefits of multi-vendor approach
[15]	Multi-agent AI systems in RAN disaggregation and virtualization
[16]	Overview of RIC architecture and intelligent controller in Open RAN
[17]	Open-source solution for closed-loop control of RAN slicing in Open RAN

Table 2.1: Some Open RAN research topics

$1 Trillion for implementing new generations of networking, specifically 5G. The article argues that opening RAN interfaces and deploying components from multiple vendors will create a competitive culture resulting in a lower product price. Implementing new network generations can take a few years to move from prototyping to a fully implemented solution in cities. There are standards such as 3GPP on networking for vendors and operators to facilitate this task. But these standards have many gaps. As a result, vendors have their internal specifications for implementations of networking architecture and design. For example, they use parameters from their internal specifications for connections to base stations. The goal of Open RAN should be standardizing interfaces to ensure compatibility among products made by different vendors. In addition, Open RAN implementations should be compatible with already implemented networks in any area. For instance, 5G or B5G RAN infrastructure should operate alongside the previous generations. The main reason is that replacing the entire networking infrastructure is costly and demands years of work. Also, not all user devices work with new network generations, so operators may need to provide backward compatibility with older devices (e.g., 4G) in addition to supporting newer networks (e.g., 5G or B5G). This approach has another benefit. It can let users switch between networks if any of them faces temporary problems.

As part of the Open RAN concept, operators should be able to mix components of different vendors. As a result, operators are not obliged to accept all the components from one vendor and can use and mix the best products from different vendors. Hardware disaggregation and virtualization in vRAN will help Open RAN to provide flexible environments that can welcome a multi-vendor architecture.

Open RAN's benefits of commercializing domains with multi-vendor inter-operable products will accelerate innovation. In this exciting innovation journey for telecommunication, AI applications play vital roles.

In [15], authors refer to a use case to discuss the role of AI in RAN disaggregation and virtualization. They consider components of CU (Control Unit), DU (Distributed Unit), and RU (Radio Unit) as multi actors or agents that can use intelligent programs to exchange information and collectively make a decision. The authors study challenges that implementing a multi-agent solution on Open RAN can introduce, such as information sharing and assigning the correct number of agents to a problem. [16] has an overview of RIC in Open RAN, its intelligent controller, and the overall architecture of RIC. In [17] researchers present an open source Open RAN solution which is a closed-loop control of RAN slicing for Open RAN. The solution

is top-to-bottom in the POWDER (the Platform for Open Wireless Data-driven Experimental Research [18]) mobile and wireless research platform .

2.3.2 Open RAN in action

As mentioned previously, Open RAN has several benefits, such as unlocking operators from single-only vendor options, decreasing cost, and increasing opportunities for innovation. However, these benefits come with some problems, such as end-to-end management of RAN. In a single-only vendor solution, that single vendor is responsible for maintaining and managing the whole system, which is not the case in Open RAN. Open RAN supports the multi-vendor implementation of RAN and requires compatibility between components built by different vendors. Interoperability and reliability in Open RAN demand a global standard that all vendors and operators can use. In 2016 a foundation named XRAN was formed by AT&T, Deutsche Telekom, SK Telecom, Intel Corporation, Texas Instruments, Radisys, Altran, and Stanford University. The XRAN Foundation aimed to create a software-based, modularised architecture for RAN. XRAN was created to standardize user plane silicon and software with open interfaces and logically centralize network intelligence and state. 3GPP has also created many networking standards that vendors and operators refer. Nevertheless, there are still gaps to address, especially regarding Open RAN requirements. That is why a team of members from the six operators, AT&T, China Mobile, Deutsche Telekom, NTT Docomo, and Orange, joined together to work on global specifications and fill the gaps in 3GPP standards. The collaboration is called O-RAN Alliance and started in 2018. The O-RAN Alliance has grown and added more active members from industry and academy involved in its projects.

The O-RAN Alliance group [19] creates standards for Open RAN based on 3GPP and in cooperation with ETSI. It also has partners who help them with implementing solutions. One of the most important partners is O-RAN Software Community [20], an Open Source Software community that works on building Open RAN components and creates frameworks for different use cases.

Other teams that work on solutions for Open RAN are TIP OpenRAN which works on disaggregated and interoperable 2G/3G/4G/5G NR Radio Access Network (RAN) solutions for Open RAN, and ONF (Open Network Foundation) that has created SD-RAN to implement near-real-time controller solutions for Open RAN. SD-RAN is a project under the Open Network

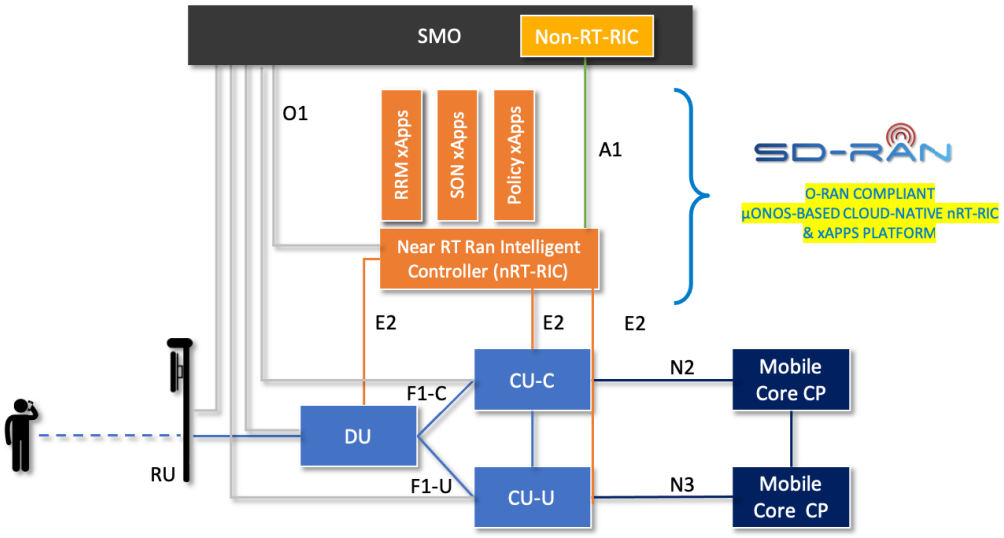


Figure 2.3: SD-RAN platform designed by ONF

Foundation (ONF) consortium hosted by Deutsche Telekom (DT) in Berlin. Fig. 2.3 present the suggested SD9-RAN platform by ONF. In [21] authors present a software development kit (SDK) that enables building specialized service-oriented controllers. The SDK has a modular architecture which makes an efficient product for SD-RAN. More and more consultants and software companies are getting involved in this technology as it becomes more evident that Open RAN is the future architecture of telecommunication.

2.3.3 Architectural Features

The key architectural factors in Open RAN are [22]:

- Disaggregation of RAN Hardware & Software on vendor neutral, 3GPP-based platforms.
- Open interfaces between components (e.g., RU/CU/DU/RIC) following universal specifications.
- Flexibility in separation or aggregation of components (DU, CU, RU, RIC, etc.) during implementation and installation.
- Adoption of new technologies such as Machine Learning and Artificial Intelligence (ML/AI) and Continuous Integration and Deployment (CICD).
- Multi-vendor approach for supplying components.

Open RAN has inherited many of its advantages from C-RAN and vRAN that are enablers of

Open RAN, such as seperation of BBU from RRU and interface virtualization.

2.4 Open RAN architecture and components

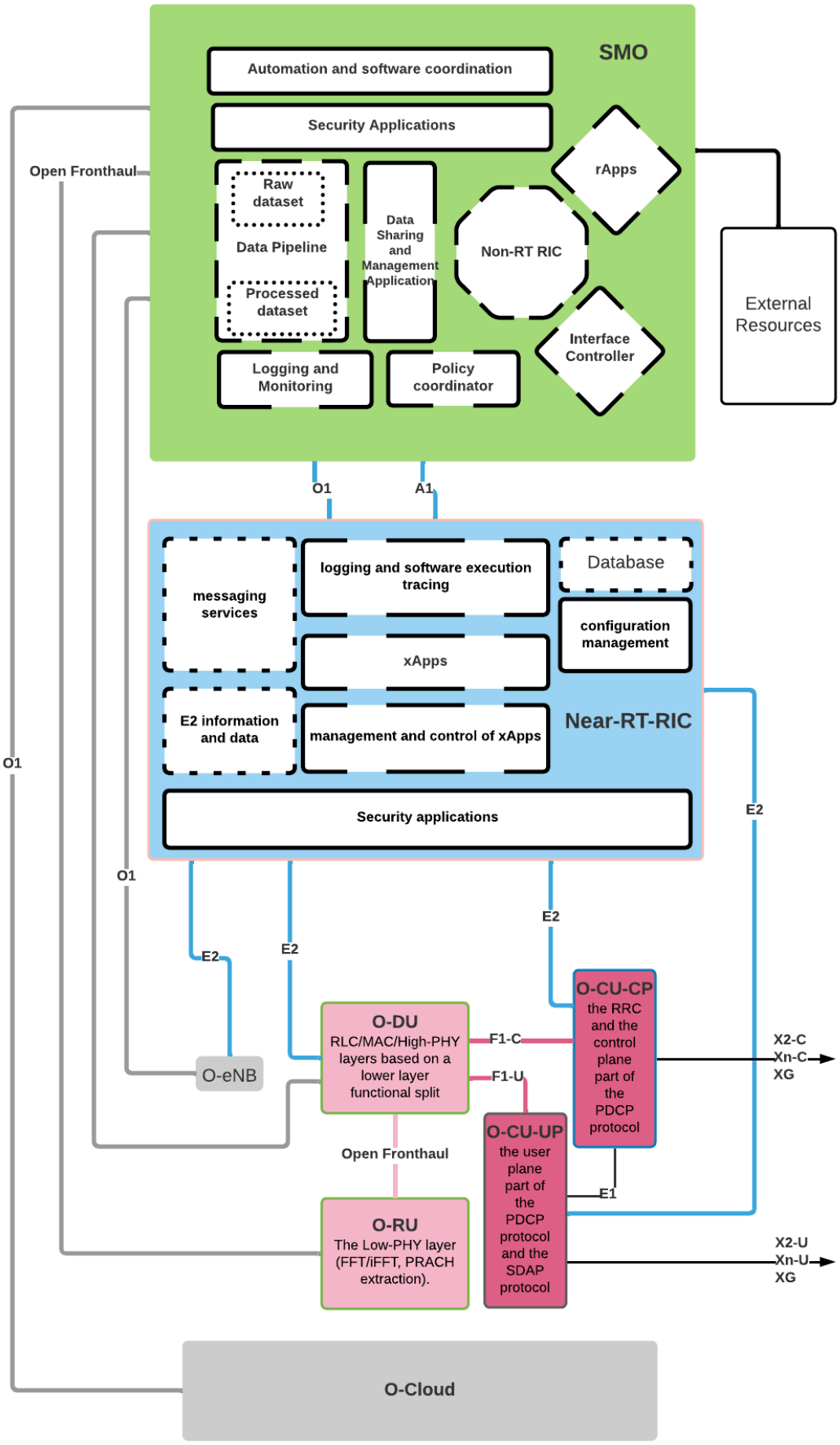
One of the main goals of Open RAN is enabling Operators to run multi-generation networking systems. Therefore, the Open RAN architecture should support both legacy generations and the new ones simultaneously. That is why observing LTE and 5G terminologies on the same Open RAN architectural diagrams. This section discusses terms and components specific to Open RAN in addition to LTE and 5G components and terminology. O-RAN Alliance has prepared and continues to issue specifications for different aspects of Open RAN. The architectural content of this paper is based on O-RAN Alliance specifications defined in [23]. The overall architecture of the Open RAN and its components are shown in Fig. 2.4. This section describes these components in the architectural point of view.

2.4.1 Non Open RAN specific components

This chapter starts with components that Open RAN has in common with other networking architectures. Some terms are initially from 3GGP TR 21.905 but overwritten by the O-RAN Alliance.

O-Cloud: O-Cloud is a cloud computing platform that is inherited from the concept of virtualization and cloud-native RAN.
O-Cloud is a collection of physical infrastructure nodes that host the O-RAN functionalities (such as Near-RT RIC, O-CU-CP, O-CU-UP, and O-DU), the management and orchestration related functions, and the underlying software components (such as Operating System, Virtual Machine Monitor, and Container Runtime). O-Cloud architecture and deployment architecture are affected by the use case, location, and the operators' preference. The current specification of O-Cloud approves both containerization and NFV (Network functions virtualization), but operators might decide to accept only one of the approaches from vendors in the future, to reduce the complexity of management and operational tasks.
Networking use cases define which combination of components should run on the same cloud platform. Fig. 2.5 presents different combinations of deploying Open RAN components. The

Figure 2.4: Overall Open RAN Architecture



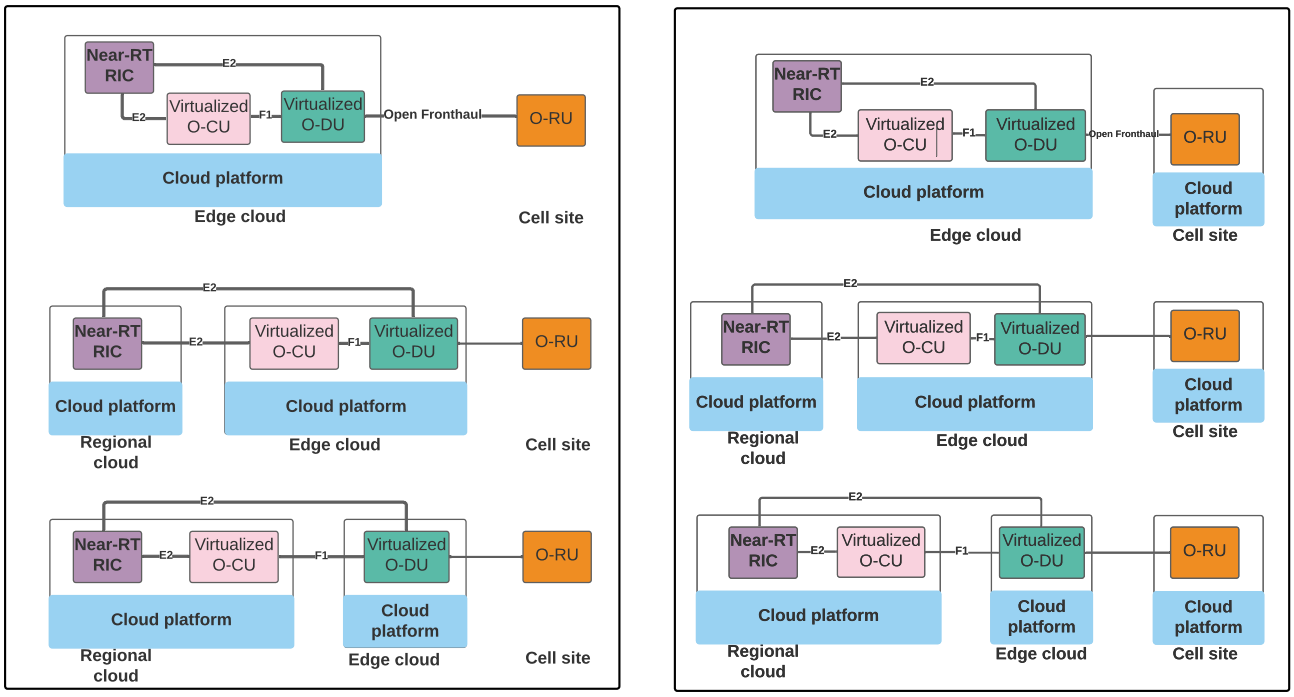


Figure 2.5: Deployment Scenarios of O-Cloud

figure shows all the different combinations of deploying RAN components on cloud environments.

NMS (Network Management System): A Network Management System for the O-RU to support legacy Open Fronthaul M-Plane deployments.

O-eNB: It is the hardware aspect of a 4G RAN that communicates with Near_RT RIC over E2.

gNB: The gNB (Next Generation NodeB) is introduced in 5G and is the succeeding Node of NodeB in 3G and eNB in 4G. The New Radio (NR) RAN has designed two diverse types of gNB, including en-gNB (the 4G Evolved Packet Core (EPC) as the core connected to the 4G LTE base station which is connected to a 5G NR base station) and ng-eNB (the 5G NG core is deployed with connection to the 4G LTE base station) to make NR and LTE compatible with each other [24]. In 5G, gNB includes CU (Central Unit) and DU (Distributed Unit).

O-CU, O-DU: In 5G, BBU is split into two functional units, centralized Unit (CU), and Distributed Unit (DU). DU is responsible for real-time L1 and L2 scheduling functions. CU is responsible for non-real-time higher L2 and L3 functions.

O-DU (O-RAN Distributed Unit): A logical node hosting RLC/MAC/High-PHY (Radio Link Control/Media Access Control/High Physical layer) layers based on a lower layer functional split.

O-RU (O-RAN Radio Unit): A logical node hosting Low-PHY layer and RF (Radio Frequency) processing based on a lower layer functional split. This is similar to 3GPP's "TRP" (Total Radiated Power) or "RRH" (Remote Radio Head) but more specific in including the Low-PHY layer (FFT/iFFT (Fast Fourier Transform/Inverse Fast Fourier Transform), Physical Random-Access Channel extraction).

O-CU (O-RAN Centralized Unit): Same as CU is a part of gNB in 5G, and it is responsible for functionalities such as Transferring User Data, Mobility Control, RAN sharing, Positioning, and Session management. CU communicates with DU over the F1 interface. In 3GPP, CU is split into CU-CP and CU-UP.

- **O-CU-CP (O-RAN Central Unit–Control Plane)**: a logical node hosting the Radio Resource Control (RRC) and the control plane part of the Packet Data Convergence Protocol (PDCP) protocol.
- **O-CU-UP (O-RAN Central Unit–User Plane)**: a logical node hosting the user plane part of the PDCP protocol and the SDAP (Service Data Application Protocol) protocol.

2.4.2 Open RAN specific components

This section reviews the architecture with components and terminologies exclusive to Open RAN, also presented in Fig. 2.4. The Open RAN architecture comprises two groups of components that are exclusive to Open RAN. The first group is those components that Open RAN originated. The second group is Open RAN components inherited from other network designs but modified to suit its needs.
5G and B5G networking generations promise use cases that require intelligent networking system to proactively manage networking tasks. As a result, the Open RAN includes intelligent

and proactive management elements as part of its base framework. The Open RAN specification [25] categorizes applications that facilitate this purpose into three groups, based on their expected latency. The first category is those real-time controlling applications that run on the DUs. The latency for these controllers is Time To Interact (TTI) and is expected to be less than 10ms. The second controller category is near-real-time applications with a latency of less than 1s. The third category is non-real-time intelligent controllers with more than 1s latency. Given the above expected latencies, the next question will be where is the best place within the Open RAN architecture to execute these applications.

RAN architectures currently include RU and DU to run real-time intelligent applications. The Open RAN specification has added two logical components to the RAN: Near-Real-Time Intelligent Controller (Near-RT RIC) and Non-Real-Time Intelligent Controller (Non-RT-RIC). These components are the primary hosts of near-real-time and non-real-time intelligent applications. Dedicating specific components for running AI applications and creating operational specifications and standards for these components is an invaluable benefit of O-RAN Alliance Open RAN architecture. It creates a universally agreed solution for running AI applications or federated learning programs in wireless communication systems.

SMO: SMO stands for Service Management and Orchestration Framework. It is not specific to Open RAN, but its role is customized in Open RAN.

The 3GPP standard number TS 28.533 has covered many aspects of management and orchestration in networking. It covers components such as operation and notification, entity-specific information management, performance management, and any combination of these components. In Open RAN, SMO includes Non-RT RIC, which we discussed before. In addition, SMO is responsible for any operation and management tasks related to O1, A1, R1, O2, and any communication with external resources. SMO also offers APIs (Application Programming Interfaces) for DU, CU, RU, and RIC configurations. The interaction between SMO and O-Cloud, which hosts RAN functionalities such as RU, DU, and CU, will continuously manage and support the ongoing activities of Open RAN. In the design and implementation, the large capacity for collecting, storing, and processing a large amount of data on SMO is a primary requirement.

SMO in Open RAN can reduce security risks and improve the management of applications and their versions by being the source of truth for both rApps and xApps. It can be responsible

for onboarding applications and keeping a catalog of deployed applications and their status. In this way, if any unknown or less known application is running on Near-RT RIC, it can send an alarm to operators. Nevertheless, implementing this logic needs enough maturity and capacity in SMO and a tendency from Operators to create the catalog of ML/AI applications in SMO. Otherwise, Near-RT RIC should maintain a trustworthy catalog of xApps. In any case, the overall system should be consistent on where to save the details of already evaluated applications that run on Near-RT RIC hosts.

Non-RT RIC: Non-Real Time RAN Intelligent Controller, or Non-RT RIC, is a logical function made of many software products within SMO (Service Management and Orchestration). It drives the contents carried across the A1 interface and comprises Non-RT RIC frameworks and Non-RT RIC applications (rApps). Non-RT RIC is responsible for operating "content transfer" via interfaces to Near-RT RIC. Non-RT RIC influences content for the O1 interface and generates *enrichment information* for Non-RT applications. For model monitoring, the ML developer will provide metrics of the model in the form of a contract so Non-RT RIC can log and present those metrics.

Non-RT RIC Apps (rApps): rApps are modular functions that leverage the functionality exposed via the R1 interface. They provide services relative to RAN, such as driving the A1 interface, recommending values and actions applied over O1/O2 interfaces. The rApps functionalities enable non-real-time control and optimization of RAN elements and resources and policy-based guidance to the applications on Near-RT RIC.

Non-RT RIC framework: This framework addresses the functionality internal to the SMO. The framework logically terminates in the A1 interface with the Near-RT RIC and is connected to rApps via its R1 interface. The Non-RT RIC Framework functionality within the Non-RT RIC provides ML/AI workflow, including model training, inference, and required updates for rApps. As shown in Fig. 2.6 Non-RT RIC is responsible for the security and data management of its functionalities.

Near-RT RIC: Near Real-Time RAN Intelligent Controller or Near-RT RIC is one of the logical functionalities to control and optimize RAN elements and resources. The control happens via fine-grained data collection and actions over the E2 interface. This component may

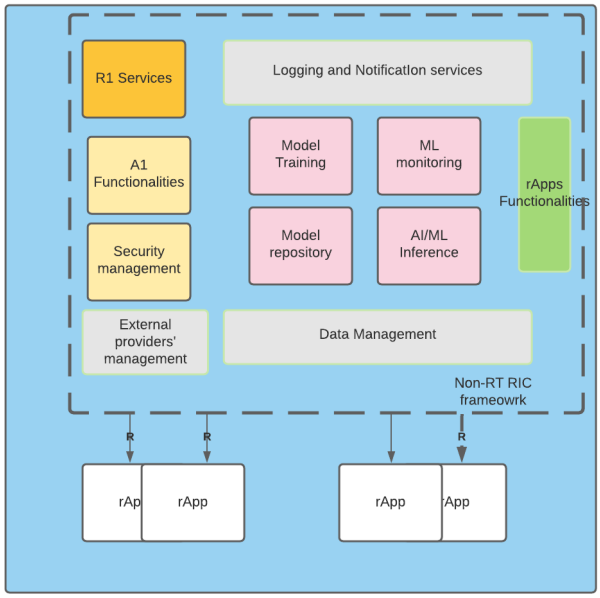


Figure 2.6: Non-RT RIC

include AI and ML pipelines, including training, inference, and updates.

Near-RT RIC is primarily responsible for control activities with less than 1s latency. These use cases do not necessarily include real-time application scenarios. However, to run intelligent applications, Near-RT RIC facilitates many functionalities that are presented in Fig. 2.7, and some of them are described below:

- Database: The database collects data from UEs and other components of RAN. It will be used as an input for Near-RT applications to decide and take action. Applications also can update or put data into that database if they need to store information.
- Management components: These components are for application management tasks such as onboarding, data sharing, and response collection.
- Messaging and notification component: This component is responsible for alerting or informing operators and vendors on the status of applications and/or platform.
- Logging and monitoring facilities for applications.
- Interface management: This component is added to manage and communicate with interfaces.

xApp: xApp is an application designed to run on the near-RT RIC. This application consists of containers with input and outputs. xApps can be provided by an approved vendor. The E2

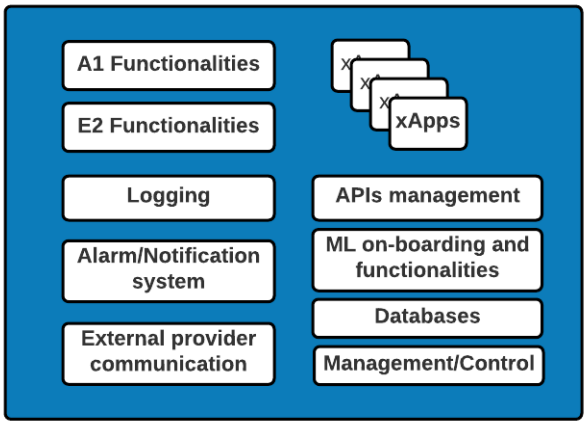


Figure 2.7: Near-RT RIC Platform

enables a direct association between the xApp and RAN functionalities.

2.4.3 Open RAN Interfaces

Three major groups of interfaces are SMO interfaces, Near-RT RIC interfaces and Nodes interfaces. Table 2.3 summarises these interfaces with their functionalities and their origin.
SMO communicates with other components of the Open RAN via O1. SMO uses O2 to communicate with the cloud and support the execution of functionalities that run on the cloud. However, Non-RT RIC which is a part of SMO uses A1 to send information regarding use cases and EI Jobs to Near-RT RIC.
Near-RT RIC in addition to A1 and O1 for interaction with SMO, uses E2 to communicate with managed elements such as O-CU, O-DU, and O-eNB. Other components that are inherited from previous generations of RAN use the same interfaces as the other RAN architectures. For example, E1 between O-CU-UP and O-CU-CP, or F1 between C-CU and O-DU. Some interfaces such as front-haul between RU and DU is also inherited from previous architecture designs but is modified by being an open interface, now being called open front-haul.
Because Open RAN is a multi generation framework, both 4G and 5G can run on this design. As a result 5G interfaces that help with this multi-generation architecture are also adopted in the Open RAN architecture. For example, X2, and Xn help with interoperability of nodes from both networking generations and NG connects 5G nodes to the core network in a standalone operation.

2.4.4 Communication details in A1:

A1 plays a crucial role in Open RAN and communication between Near-RT RIC and Non-RT RIC. Therefore, we dedicate a section to this interface and the life cycle of data transfer via A1.

Enrichment information

SMO collects information from internal and external resources. This information which is called Enrichment Information (EI), is being used by both SMO functionalities including Non-RT RIC and Near-RT RIC.

A1 policy

Non-RT RIC uses a declarative policy to lead Near-RT RIC functionalities via A1. In a declarative policy, statements express the goals of the policy but not how to accomplish those goals. This policy is called the A1 policy and is non-persistent. In other words, it cannot survive if Near-RT RIC restarts. Non-RT RIC provides EI to Near-RT RIC via A1, using A1 policy.
A1 Enrichment Information is a phrase to address the information which is collected or derived at SMO and Non-RT RIC. Sources of EI can be external or internal Open RAN resources. EI is either not directly available to Near-RT RIC or cannot be derived inside Near-RT RIC from network data due to processing or storage constraints. Near-RT RIC requests for delivering this information by using formal statements, called *EI jobs* created, modified, and deleted by Non-RT RIC.

EI Functions

EI functions are functions that generate and manage the delivery of the A1 EI. This function is also responsible for publishing Ids for different types of EI as EiTypeIds. Fig. 2.8 presents the EI jobs' lifecycle. It starts with *Registration* when Non-RT RIC generates a new EI Job from collected input information by SMO. Then in the stage of *Discovery* Near-RT RIC discovers new EI jobs and requests the detailed description of the EI. Near-RT RIC can send a request for EI to Non-RT RIC (The *Request* stage). The stage of *Request* generates an EI Job and an Id named EiJobId that will be assigned to the corresponding the EI Job. In the last stage, the stage of *Delivery*, Non-RT RIC sets up the connection and delivers the EI job via A1 on a push

delivery basis.

2.5 Artificial Intelligence and Open RAN

In this section, we focus on how intelligent applications can improve the efficiency of RAN, specifically with the architecture of Open RAN. This section starts with a brief review of previous implementations of AI algorithms in telecommunication. Following the O-RAN Alliance specifications [25], we categorize intelligent applications that will control RAN in Open RAN into three categories based on their expected latency. These categories are real-time controllers that run on DU, near-real-time applications, and non-real-time applications. [26] has reviewed the employment of AI in Open RAN based on different network levels. In this research work, we look at applications of AI in terms of latency requirements with more details about their background, implementation in production, and challenges.

2.5.1 AI in Telecommunication

ML/AI algorithms, in general, are being categorized into supervised, unsupervised, and reinforcement learning. Fig. 2.9 has summarised common telecommunication problems and ML/AI algorithms in each category that researchers have used to solve those problems. In this section we discuss the benefits and challenges in each one three major categories of ML models.

Supervised learning

Supervised learning algorithms are AI algorithms that take labeled data as input for training. In this learning method, humans control what the algorithm learns. Nevertheless, this learning method needs a large amount of correctly labeled data which leads to the human capital cost and any potential human error in labeling. Nevertheless, some common supervised algorithms have been used in telecommunication extensively. In telecommunication, Support Vector Machine (SVM) is used in security and time-series forecasting, and Naive Bayes for intrusion detection, TCP (Transmission Control Protocol) enhancement [27], DDOS (Distributed Denial Of Service) attack, and localization. Algorithms such as Logistic regression, Random Forest, and decision trees have been used to solve security, intrusion, and DDOS attacks problems.

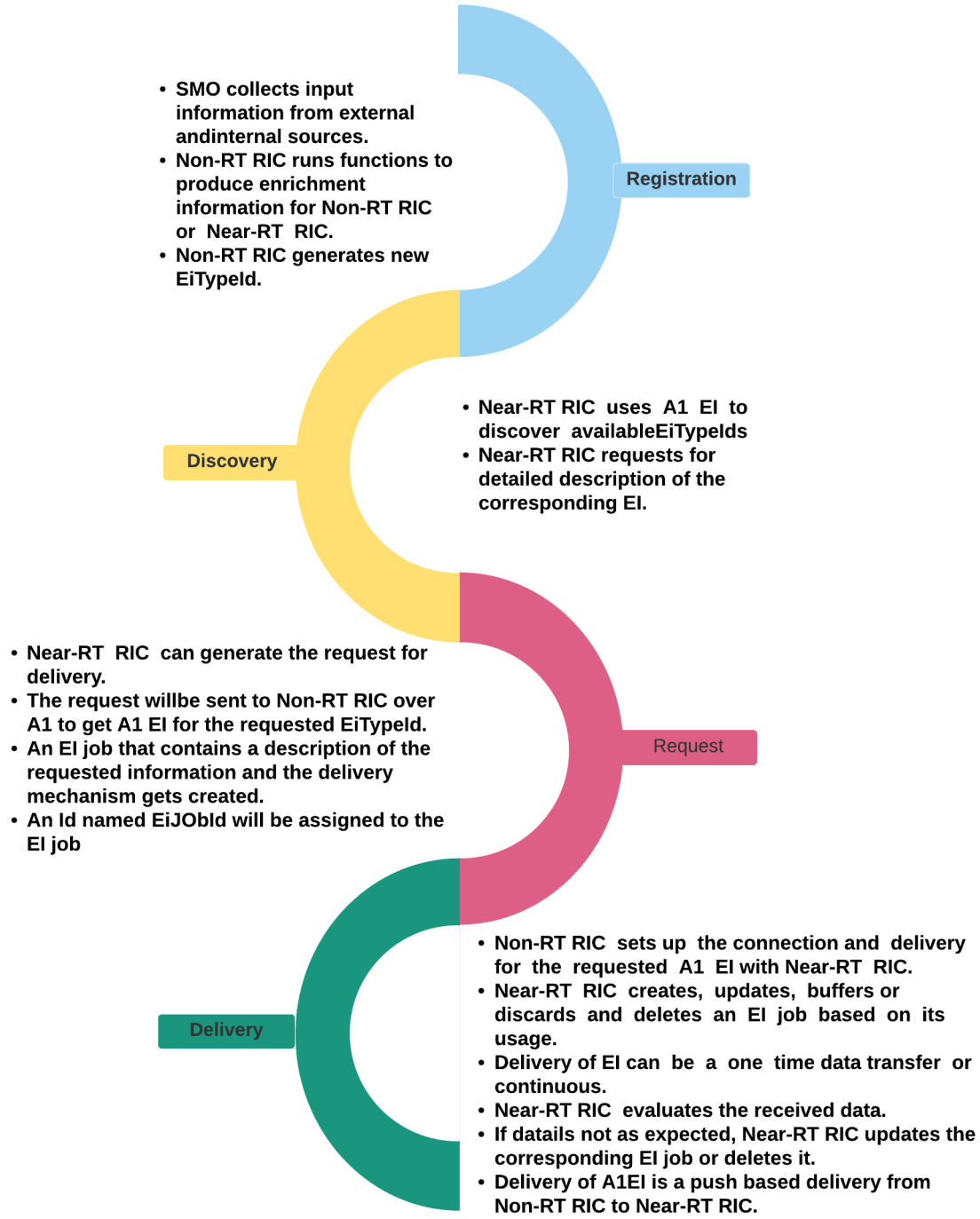


Figure 2.8: EI Job Lifecycle

Supervised learning algorithms have been extensively employed in wireless security applications, particularly in threat detection and prevention. For example, SVM classifiers have been effectively used to detect rogue access points and unauthorized wireless devices by learning patterns of normal versus suspicious network behavior [28]. In wireless intrusion detection systems, supervised learning helps identify various attack patterns such as MAC spoofing [29] and deauthentication attacks [30]. Furthermore, supervised learning algorithms have demonstrated success in wireless jamming detection and mitigation [31]. Random Forest classifiers have been trained to differentiate between legitimate signal interference and intentional jamming attacks by analyzing features like signal strength patterns, packet delivery ratios, and channel utilization metrics [32]. Logistic regression models have been employed to predict potential security breaches in wireless networks by learning from historical security incident data and network behavior patterns [33]. Decision trees have proven particularly effective in real-time wireless security applications, as they can quickly classify incoming traffic patterns and make rapid decisions about potential threats [34]. These algorithms have been successfully implemented in detecting man-in-the-middle attacks in wireless networks by analyzing patterns in authentication requests and network traffic behavior.

Unsupervised learning

Unsupervised learning is a learning method that does not require labeled input data. As a result, it reduces the cost of pre-processing steps in machine learning. However, the range of problems that this learning method can solve is minimal compared to the supervised learning method. Some of the unsupervised learning methods that have been used in telecommunication are clustering algorithms for sensor networks and locating controllers in Software Defined Network (SDN) systems or data mining algorithms to ensure the reliability and accuracy of solutions. Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) are common unsupervised algorithms in data processing. Authors in [35] demonstrated the importance of SVD for MIMO. Researchers in [36] used PCA to detect anomalies in wireless mesh network.

Reinforcement learning

Reinforcement learning (RL) is a method that an agent learns patterns and decision-making strategies by interacting with its environment. It does not need labeled data like supervised learning. But it requires accurate environment modeling and sometimes more iteration than

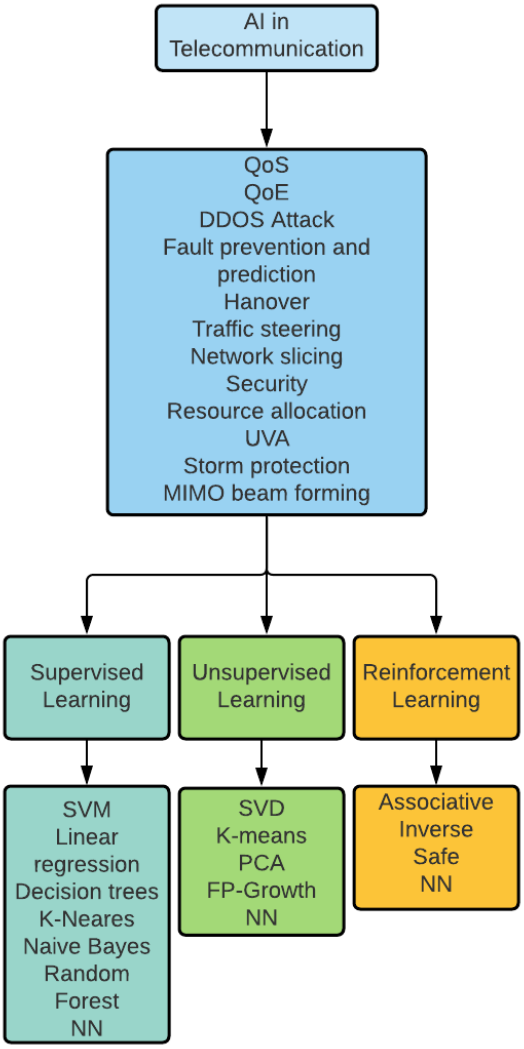


Figure 2.9: AI in Telecommunication

supervised learning to achieve the required accuracy. RL has been used in telecommunication for many use cases such as packet routing, beamforming, Hand Over-optimization, and other use cases.

In telecommunication, any of these learning methods or a combination of them can be used to solve problems or improve the efficiency of tasks. Table 2.4 has summarised learning methods with some of the research papers on each topic.

2.5.2 Running ML/AI on OpenRAN

ML/AI algorithms can solve many problems in a telecommunication system. Fig. 2.11 shows some telecommunication problems that AI algorithms can solve. Intelligent functionalities that manage the RAN (Radio Access Network) can be classified based on their learning approach, such as supervised or unsupervised learning. They can also be categorized according to their latency requirements. Real-time applications need to operate with latency under 10ms, while Near-Real-Time (Near-RT) applications must respond within 1 second. Non-Real-Time (Non-RT) applications can tolerate latencies longer than 1 second. Real-time MAC (Medium Access Control) and PHY (Physical) layer intelligent applications run on the Distributed Unit (DU), which is why they are referred to as DU applications. Fig. 2.10 presents these three intelligent controllers. Although researchers have been working on enhancing RAN with AI applications, O-RAN Alliance created specifications and designed an AI enabler architecture. The new architecture will help researchers and engineers follow the same standard for deploying AI applications. Following Open RAN specifications, we call Near-RT and Non-RT intelligent applications Near-RT RIC and Non-RT RIC, respectively.

One of the most important parts of any intelligent system is its data pipeline. Collecting, processing, and passing data to applications in a secure, robust, reliable, and efficient is critical. The quality of output from machine learning models and analysis programs is highly dependent on the quality of data pipelines. Therefore, we review data-related applications as part of intelligent controllers. In the following paragraphs, we start with real-time applications and move to the other two groups of RAN Intelligent Controllers (RIC). It is worth mentioning that solving many problems needs collaboration among applications from different families. Also, Non-RT RIC and SMO might include applications that respond in less than 1s, or training algorithms can happen on a Near-RT RIC host to meet some KPIs, although they usually take more than 1s to complete.

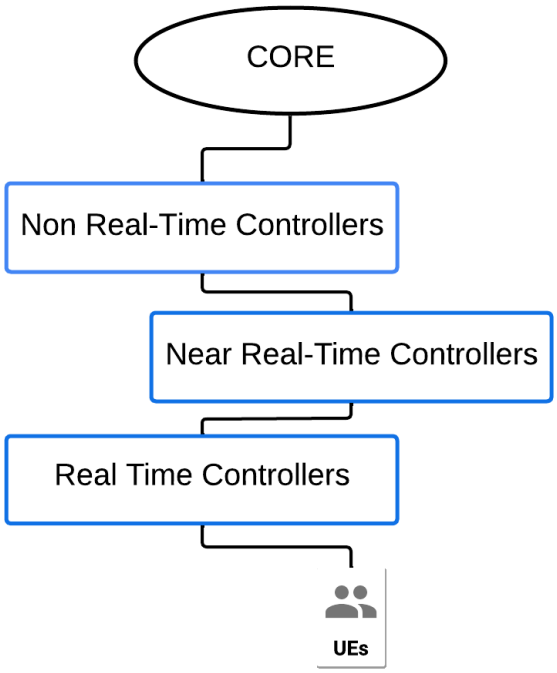


Figure 2.10: Three Intelligent Controllers in Open RAN design

2.5.3 Real-time intelligent controllers

Running intelligent applications close to UEs has started before Open RAN. Researchers in [68] developed an AI solution to maximize throughput running on BBU. Researchers in [69] used LSTM for traffic prediction at the edge, running on BBU. Light ML algorithms and small size but powerful processors have shifted the expectation for ML/AI applications. When an *application latency* is defined as real-time, the program should run on components as close as possible to users. DU is one of the hosts for real-time applications in wireless networks. DU is responsible for PHY and MAC layer scheduling functions.

MAC Layer functions

MAC schedulers, first introduced in LTE networks, have evolved and transitioned into 5G technology. These schedulers operate at the MAC layer, managing resource allocation to user devices while ensuring Quality of Service (QoS) requirements are met. As networks become more complex, Machine Learning and Artificial Intelligence techniques have emerged as valuable tools for enhancing MAC scheduler capabilities, particularly in areas such as Link Adaptation, Massive MIMO, and multi-user MIMO operations.

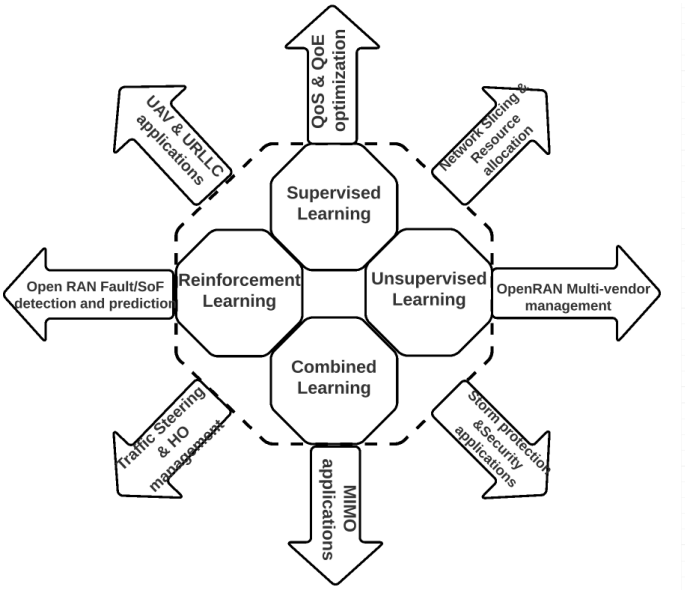


Figure 2.11: ML/AI in Open RAN

In exploring these AI-enhanced capabilities, an industrial project by Capgemini [70] investigated the development of a cognitive MAC layer focused on predicting user equipment (UE) mobility patterns. However, their experimental results revealed a critical challenge: the computational demands of both standard MAC layer operations and ML model predictions created resource contention. This competition for resources made it difficult to maintain the required 10ms response time threshold, highlighting a key implementation challenge for AI-enhanced MAC schedulers.

In [71] a real-time application focused on predicting encrypted packets in video streaming use cases. They combined an unsupervised clustering model with an adaptive classification approach to classify frames and used a time series forecasting algorithm to predict the streaming frames.

2.5.4 Near-RT RIC

Near RT RIC is connected to Non-RT RIC to access data collected from internal and external resources. It is also connected to CU and DU that are closer to UEs.

As we mentioned before, one of the considerable benefits of Near-RT RIC is helping engineers and researchers have a universally agreed platform for running many edge applications or federated learning programs.

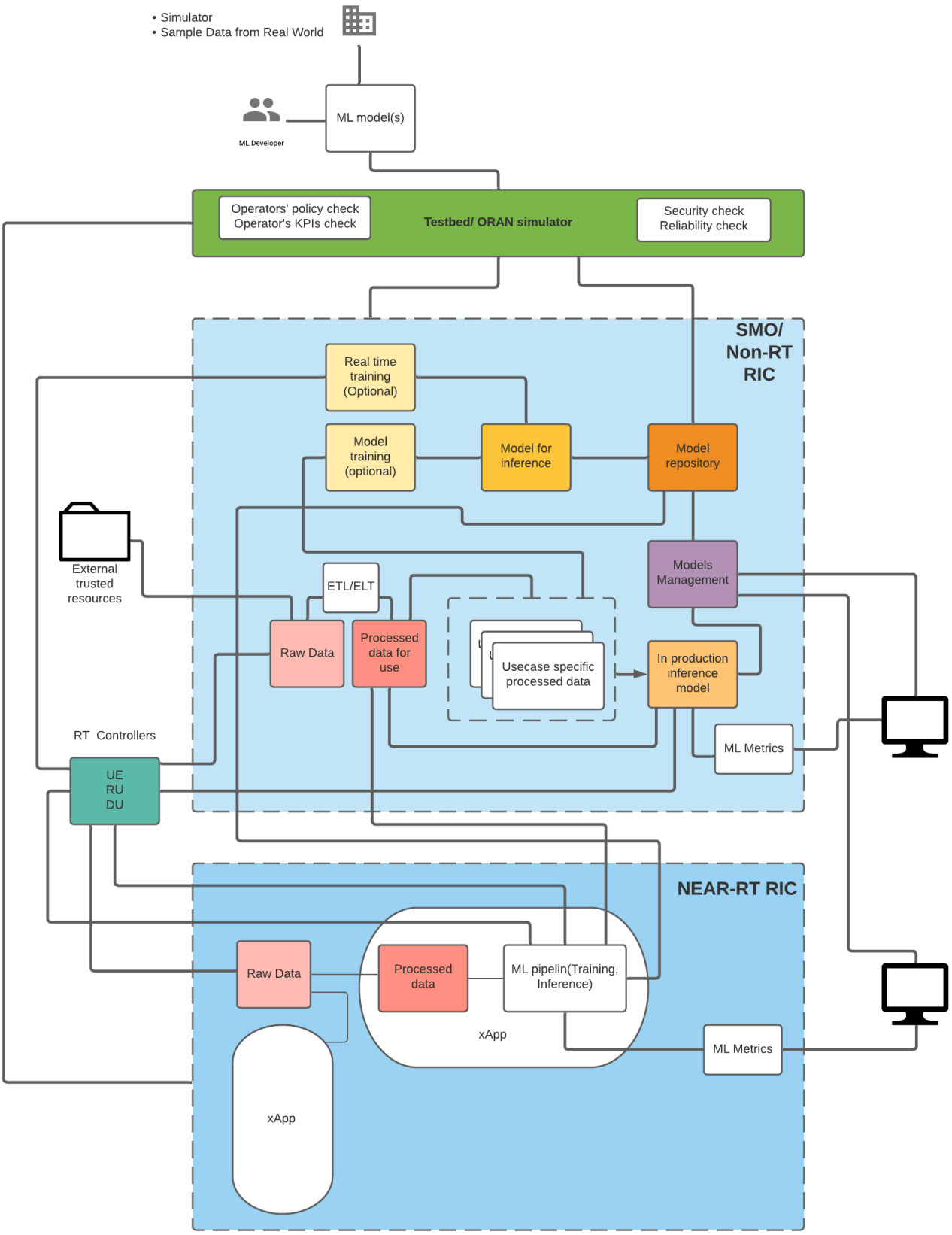


Figure 2.12: Overall ML/AI workflow

Dynamic HO Management

In wireless communication the process of transferring an ongoing call or data session from one channel connected to the core network to another channel is called Handover (HO). To optimize HO and perform near-real-time optimization, we need ML models which run on Near-RT RIC. In this solution, the xApp, an AI model, will monitor the device-specific mobility, predict or detect unexpected HO events, and generate HO sequences to prevent anomalies. One of them is Vehicle to Everything, known as V2X. It is about communication between a vehicle and surrounding items that can receive or send any information from or to the vehicle. The main goal of this topic is to improve road safety and traffic efficiency and reduce energy consumption. The 3GPP standards include standards regarding V2X and V2V (Vehicle to Vehicle). Also, IEEE 802.11 has an amendment, IEEE 802.11p, for wireless access in vehicular environments. Using Open RAN on top of these standards makes some of the goals for V2X communication achievable. The IEEE standard of 802.11p supports direct communication between vehicles and their environment. On the other hand, 3GPP covers cellular V2X (C-V2X) in communication, specifically for 5G. C-V2X includes both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I), as well as Vehicle-to-Network (V2N) [72]. In V2X, the communication happens between the V2X application server (V2X AS) and the V2X device attached to the vehicle. The V2X AS handles services such as data delivery data V2X devices. The application also provisions the 5G core and the V2X device with parameters. At the sub-optimal HO, some anomalies might happen, such as short stay, ping-pong, and the remote cell. An xApp should update the database maintained by V2X AS to improve the V2X user experience. This use case can run on Near-RT RIC as an xApp. Non-RT RIC can also support this use case which we will address later in 2.5.5.

Traffic Steering

Near-RT RIC can constantly monitor users and ensure KPIs of QoE are met. Running traffic steering relies on the historical data of available cells and networks. Near-RT RIC can access this information from SMO and Non-RT RIC or if the resources are not limited, use the same host to collect and use this data. BY using this information, it can optimize UEs Received Signal Strength(RSS) and energy consumption of batteries.

In some papers such as [73],[74],[75],[76], the researchers suggested a reinforcement learning algorithm that runs on UEs and, by learning from the past experiences of the device in an envi-

ronment, improves its QoS and reduces its energy consummations. This approach that suggests running the model on UEs has some problems. Firstly, it threatens the network system's security or is unavailable to all users. Secondly, the underlying system and the networking system can change dynamically. Updating the model running on user devices is not always possible. A solution can be running a model as a xApp on the Near RT RIC platform, using AI algorithms to improve UEs QoS and QoE. It can also be used as part of a federated learning solution among UEs so they continuously will be updated on the current status of their environment or the latest version of the model.

2.5.5 SMO and Non-Real-Time Controller

This section explores applications that do not necessarily need a near real-time or real-time response. However, in some scenarios, these applications collaborate with Near RT RIC or support it to solve some problems.

Data Governance and Analysis

Data Governance and analysis are crucial tasks for making intelligent decisions in any business. We discuss these applications in the category of Non-RT controllers because SMO is where data from all external and internal resources are collected. Therefore, SMO and Non-RT RIC platforms can implement all sorts of data-related logics and tasks, regardless of latency constraints. Nevertheless, the other two categories of ran intelligent controllers also can include such applications.

To improve QoE, operators need to understand and monitor their users in the most intelligent way. Data analysis and visualization are systematic approaches for this purpose. Engineers and scientists collect, store and process data to prepare it for analysis or ML/AI applications. Handling data pipelines securely and efficiently while complying with data governance rules and standards is crucial in Non-RT RIC and SMO.

Without a reliable data pipeline achieving a reliable, scalable, and intelligent networking system is impossible.

Management and Orchestration

SMO is responsible for the management and orchestration of the Open RAN environment. However, managing and orchestrating an environment such as Open RAN, with different stakeholders, various components, multi-vendor solutions, a large number of parameters, and many different data flows to or from components, is complicated and challenging. Communications between elements of Open RAN are potential targets of malicious actors. Data collection and processing from internal and external sources require continuous quality and safety policies. Incident prevention and quick response to alerts without ML/AI solutions are almost impossible. End-to-end network slicing and resource allocation, especially in dense urban areas, are complex challenges for operators to overcome. These problems are not new to researchers.
[77] used SDN principles to orchestrate and coordinate resources in a 5G RAN infrastructure. Another published work [78] proposed a functional split orchestration scheme for running 5G on Cloud-RAN. The authors evaluated the results from running the proposed design on a 5G experimental prototype and claimed the suggested design reduces energy consumption and the deployment cost.

Network Storm Protection

Storm protection and control prevent LAN interfaces from interruptions caused by a broadcast storm. A broadcast storm happens when network packets flood the subnet, creating excessive traffic and degrading network performance [79]. This action degrades the network performance and interrupts crucial communications such as health-related applications. As technology advances, more mission-critical use cases use wireless networking, where a loss of connection may have life-threatening consequences. Broadcast storms can be a consequence of errors in network design or installation. However, there can be some intentional attacks causing this disturbance. In these scenarios, an attacker usually finds vulnerable devices connected to the internet and manipulates them by aggressively sending many packets to create a storm and cause an outage of an extensive network. At the moment, the most common reaction is rejecting both benign and malicious services. However, intelligent algorithms can detect the compromised device, preventing malicious actions while serving benign service requests. A good solution should include both detection and mitigation capabilities. Although detection and prevention of the applications can run on Near-RT RIC, some solutions can better run on SMO and Non-RT

RIC. Information such as device type, International Mobile Equipment Identity (IMEI), and Public Land Mobile Networks (PLMN) is available on SMO. SMO also has the scheme of the attack. Non-RT RIC can feed Near-RT with this data and help it detect and stop the threat on edge. In this way, Non-RT RIC will do heavy computation tasks and send input data to near-RT xApps and help them make more intelligent and faster actions.

Dynamic Hand Over (HO) management

Efficient Handover management in networking has always been a challenging task, but new use cases such as V2X make this challenge an urgent research problem. In general, The HO sequences happen mainly based on Neighborhood Relation Tables (NRTs), which are maintained by xNB, so devices themselves do not have much information to optimize them.
ML/AI applications can optimize HO in different ways, such as navigation and radio statistics history. In this solution, data gets sent from devices through O1 to SMO, where the Non-RT RIC applications run, and databases collect data. An ML/AI application can find anomalies and suggest resolutions accordingly. In this solution, The received data from devices will be an input for the AI algorithm to learn and find anomalies. The V2X AS is responsible for data maintenance and providing information to AI algorithms. The algorithm's output should be finding the anomalies and the resolutions accordingly. [80] worked on 3G/4G cellular networks to maintain QoE for a UE that sends RSRP (Reference Signals Received Power) and RSRQ (Reference Signal Received Quality) of its serving cell. If the received power and quality are less than a defined threshold, the serving eNB selects another target cell and triggers the HO process by sending the message to MME (Mobile Management Entity), and consequently, a message will be sent from MME to NB if enough resources are available. If eNB gets the message from MME, it switches the target cell to the new one, updates the table, and releases resources from the old cell. The paper used regression and neural networks to predict handovers based on their data analytic results. Another research [81] used ML/AI in SON to identify and improve faulty cells and reduce packet loss. They used the KNN algorithm to detect abnormal access points (AP), classify access points, and find HO delay. They compared their results with other algorithms such as SVM, Random forest, and K-mean. Another research team used RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory) to learn latency and cost associated with service requests. They used a real dataset of real-world vehicle movements to create a simulation environment for training their algorithm. Their goal was optimizing

vehicular fog computing by HO optimization [82]. In another research [83], authors focused on anomaly detection using a semi-supervised algorithm. They used this method to detect two types of cells that cause abnormal behavior in a system. The two types were the sleeping cell caused by very low user requests and the too busy cells caused by too many requests that cause more demand for resource allocation. In [84], researchers proposed a method that combines fuzzy logic and multiple attribute decision algorithm (MADM) to use historical data and find the optimum target. As a result, the generated model triggers the HO at the right time rather than being late or dismissed. In sum, using datasets from real-world scenarios, combined with one of the supervised or unsupervised methods, can help create an AI model that can detect abnormal cells or anomalies in the system. This detection can trigger further actions or be transferred to the Near-RT RIC platform to find anomalies in near real-time.

UAV Applications

An Uncrewed Aerial Vehicle (UAV) is a component of an Uncrewed Aerial System (UAS) that can fly and move without any human pilot on board. There are many benefits to using UAS in telecommunication. RAN architectures can benefit from UAS and UAV to overcome data traffics' high load and variability. Telecommunication systems can use them as additional stations, acting either as a base (BS) or relay stations (RS), especially in unexpected, natural disasters, or special events. UAVs can operate as BS or RS to increase the coverage area, balance traffic load, and enhance network capacity. The benefits of using UAVs are not limited to their mobility and flexibility in geolocation. The flexibility in deploying various products and providing line-of-site connectivity are other benefits of this technology for telecommunication, specifically RAN. Although UAS individually has had significant progress in providing services to users, there are still many telecommunication challenges to resolve that ML/AI can come to the rescue. One scenario that needs HO is Radio Resource Allocation (RRA). Based on parameters and features such as data traffic rate, latency tolerance, and reliability that define user group, an ML/AI program can find a pattern and efficiently provide the best suggestion on resource allocation and other decision-making tasks. The selection of ML/AI algorithms depends on the nature of the business question. Predicting future behavior is better executed by a supervised learning algorithm that can learn very well from the past and make a model that can more accurately predict the subsequent requests. At the same time, an online decision in a dynamic environment might suit a Reinforcement learning algorithm trained in similar

dynamic environments and can take quick actions in response to changes, incoming requests, or events. When it comes to UAVs, we cannot underestimate the impact of weather conditions on their operation. ML/AI algorithms can combine weather conditions and telecommunication parameters to generate models that can operate UAVs better than a manual controller or a hardcoded program [85],[86].

Traffic steering

The increasing number of telecommunication users and imbalances in traffic caused by their UEs or various bandwidths available to users have challenged telecommunication operators. The 3GPP Self-Organizing Network (SON) function includes Mobility Load Balancing (MLB). MLB balances the load by optimizing the handover triggers and handover decisions using load information shared between neighboring cells. However, it treats different user groups equally. In addition, since 5G, the networking system can support different combinations of access technologies such as LTE (Licensed band), NR (licensed band), NR_U (unlicensed band), and WI-FI (unlicensed band). As a result, finding a solution to prioritize and provide services according to the request type becomes inevitable. Traffic Steering addresses this challenge. Based on the traffic type (e.g., HTTPS (Hypertext Transfer Protocol Secure) or HTTP), user membership profile, or the priority level of the service request, and many other factors, the request can be served by different operational activities or a combination of access technologies.
Traffic management policies with flexible configuration can help proactively manage user traffic across different technologies. SMO can collect information from UEs and, in collaboration with Non-RT RIC, can monitor user experience by measuring the UE performance and resource utilization on the cell level. Where the service requirements are not acceptable, it can locate the target cell and implement a solution such as switching cells for the affected user or offloading that cell, or increasing bandwidth if it is possible. Moreover, in multi-access systems, traffic steering can happen between different environments. Non-RT RIC can create traffic management policies specific for any UE and based on the priority of cells for each UE. Non-RT RIC sends policies to the Near-RT RIC to enforce the radio resource control. In addition, Non-RT RIC, by exploring the historical data, can extract radio fingerprint EI. By monitoring the inter-frequency measurement and passing information to Near-RT RIC, the system can predict the inter-frequency measurement. This prediction can reduce the unnecessary inter-frequencies to boost traffic optimization and network performance. There are many academic works on MLB

algorithms to improve QoE. Although those works consider MLB as a solution, their logic and algorithms belong to Non-RT RIC applications. The shortcomings of this solution can be used as challenges for improving QoE. For example, authors in a research [87], after studying the previously suggested algorithms for MLB and addressing the unfairness of those algorithms, provided a QoE-aware algorithm for LTE systems. They use a fuzzy logic controller to tune handover parameters to reduce QoE differences across cells and services. They argue that this method will provide a fair service across all users. They validated their algorithm in a dynamic system-level simulator of a macro-cellular LTE scenario. In [34], authors exclusively studied data analytics with Radio Access Technology (RAT) selection scheme and discussed acquiring contextual information and minimizing control signaling exchange. As we mentioned in Near-RT RIC, monitoring users and checking KPIs (Key Performance Indicator) can be executed by Near-RT RIC.

Optimization of Massive MIMO beam forming

Massive multiple-input multiple-output (MIMO) wireless communications refers to the concept of implementing a large number of antennas in a cellular base station [88]. [89] describes the benefits of using MIMO to increase the capacity on the scale of ten times or more and improve the radiated energy at the same time. Massive MIMO enhances energy efficiency by precisely focusing signal energy into targeted spatial regions while simultaneously reducing latency in user device communications. As 5G and B5G are in the range of mmWave and THz and are the critical path for improving networking communications, the importance of Massive MIMO becomes apparent. One of many papers in this area is [90] which proposes an ML/AI influenced design for Ultra Massive MIMO and intelligent surfaces. Although many research works are on MIMO and intelligent surfaces, designing an end-to-end solution from core networking to user devices is still a massive challenge for researchers and operators to overcome. In this use case, AI can help with the optimization of parameters and locations.

Network Slicing

Network slicing has been a topic of interest for improving communication performance for many years [91], [92]. Because of technology advancements and making concepts such as NFV (Network Functions Virtualization) and SDN (Software Defined Networking) applicable, network slicing became a key technology for 5G and B5G.

"Network slicing is a paradigm where logical networks/partitions are created, with appropriate isolation, resources, and optimized topology to serve a purpose or service category (e.g., use case/traffic category, or for MNO internal reasons) or customers (a logical system created on-demand)" [93]. A network slice is a virtual network with a group of allocated services over a shared network infrastructure.
However, network slicing is a challenging task. Services which send requests to RAN for resources are dynamic and various. They are different in terms of accepted latency, required computation, time duration, and priority in terms of emergency. A utility request related to autonomous vehicle alarms can tolerate much less latency than a device reporting the customer traffic into a shopping center. Fortunately, machine learning algorithms can be used practically in network slicing and help manage the network communication's life cycle.
The provisioning of network slicing includes the four phases: preparation, commissioning, operation, and decommissioning. The NSI/NSSI (Network Slice Instance/Network Slice Sub-net Instance) provisioning operations include: Create an NSI/NSSI, Activate an NSI/NSSI, Deactive an NSI/NSSI, Modify an NSI/NSSI, Terminate an NSI/NSSI. Some of important use cases of network Slicing are:

1. RAN Slice SLA (Service Level Agreement) Assurance: In this use case, the flexibility of Open RAN combined with network slicing will allow multiple vendors to request resources from RAN and serve their clients with different applications.

2. NSSI (Network Slice Subnet Instances) Resource Allocation Optimization. ML/AI algorithms and methods have been used for these scenarios and implemented in Open RAN.

[94] presented a unified model which included the network slices, data, and slice managers run by ML, communicating with SDN and the rest of the network over APIs. In Open RAN, the data and slice manager applications can be deployed as xApps on the Near RT RIC platform but can also be deployed on Non-RT RIC. In another research, [95] the authors discussed the challenges that network slicing faces, such as Heterogeneous QoS (Quality of Service) requirements of various services, the fluctuating status of networks, overhead costs caused by network slicing. The authors proposed a network slicing framework and a neural network algorithm to predict the resource allocation requirement. The authors of [95] suggested a multi-layer control level for network slicing, such as gNodeB-level and package scheduling level. However, as data

is scarce, they suggest using transfer learning to overcome this challenge and generate synthetic data to train and build highly accurate models.

One of the crucial tasks in network slicing is to allocate resources adequately based on the plausible usages in the region. For example, IoT devices might need resources during peak times or events, such as some sport matches that increase the number of requests considerably in a short period, demand for intelligent methods for prediction, and allocation of required resources.

Open challenges in network slicing include NSSI resource allocation, multi-vendor NS, indoor positioning, congestion prediction and management, IIoT (Industrial Internet of Things) optimization, and dynamic spectrum sharing.

Finding optimum solutions for each topic, especially for running 5G and B5G applications on Open RAN, is a complex but crucial task. Smart grids are also another new technology that can improve resource allocation policies. In [96] the authors describe the use of smart grid technology in 5G and B5G. The integration of AI application with IoT and smart grids in 5G and B5G, can help with the resource allocation.

Use cases that we discussed represent only a subset of the numerous potential opportunities for AI applications within Open RAN and the broader telecommunications ecosystem.

2.6 MLOps in Open RAN

The phrase MLOps stands for Machine Learning in Operations and refers to the efficient deployment and application of machine learning solutions in production. A ML/AI pipeline in production should be scalable, reproducible, and maintainable. Moreover, ML/AI pipeline should have a robust monitoring and versioning system for data, models, and features. In MLOps, an artificial orchestrator automates and maintains these principles. Engineers from open source communities to major cloud providers have created some frameworks and products to help data scientists run their ML/AI solutions efficiently in production based on MLOps principles. Fig. 2.12 shows the main concept and workflow of ML applications in production. However, these solutions are for general purpose applications where ML/AI pipelines can run on a protected data center at any location. Nevertheless, Open RAN is an emerging technology to explore. Near-real-time and real-time ML/AI solutions are less likely to run on the currently available data centers. Low latency applications that need to be close to user devices cannot

run on traditional cloud and data centers. They need to run on edge devices close to base stations and UEs (User Equipment) but have limited resources for computation and data-related activities. Also, executing all the pipeline steps on one platform might not be possible, and we might need to split pipelines to run on Non-RT RIC and Near-RT RIC platforms. Therefore, maintaining the performance of ML/AI pipelines and maintaining security in communications become inevitable challenges to manage.

This section is continued by studying commonly used ML/AI pipelines required in production which can be used in Open RAN. This section does not include different scenarios for data pipelines. Data pipelines which include data storage, data processing, and data privacy procedures, might require a large amount of memory or processing resources and can affect the location and the strategy of data pipelines. They can affect or be affected by the corresponding ML pipeline. O-RAN Alliance has published a specification on AI/ML workflow [25]. It includes multiple deployment scenarios. However, this chapter covers standard deployment practices in addition to practical scenarios such as the need to run multiple versions of a model in Near-RT RIC that the specification does not cover. Fig. 2.13, 2.14, 2.15, and 2.16 present some of common workflows in MLOps that are discussed in this section.

2.6.1 Single Data Single Model

After selecting the algorithm and defining the data source in this pipeline, the algorithm will be trained on the collected and processed dataset. The first time training of the model is usually on batch datasets, even if the inference and prediction run on a data stream. For example, a defect detection model can be trained on batches of defective items and then receive a stream of scanned images for identifying the problem on a real-time basis. There are exceptions for this practice, such as reinforcement learning algorithms that run in a simulated environment. The output of training algorithms is a generated model. This model will be deployed to production, usually as a container. The container accepts inputs based on the model requirements. Therefore the production pipeline includes a model with input data from stored data or as a stream from an external source. The outputs and metrics are stored and presented to the end-user via a monitoring system. Most of the time, new types of data over time degrades the model performance. In this case, the engineer or data scientist will add new sample data to training datasets or change training parameters to retrain the algorithm and redeploy the model.

Fig. 2.13 presents this logic. Data from datasets flows to an algorithm that is running offline.

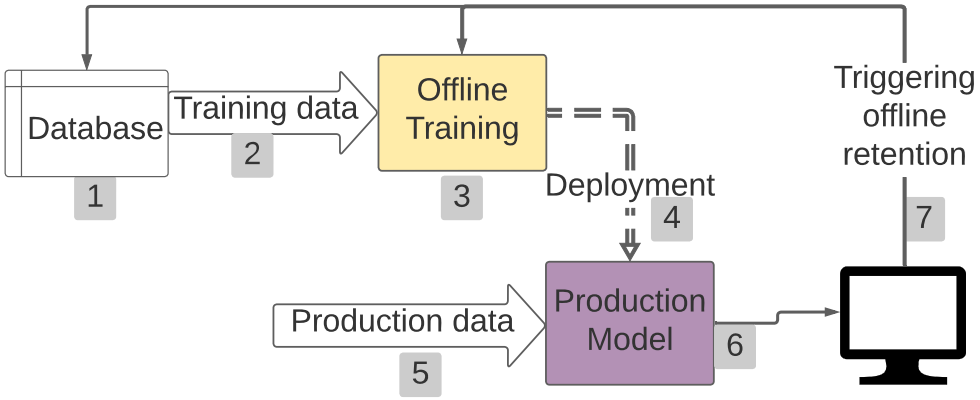


Figure 2.13: Single Model ML Pipeline

The output of this training is a model that meets the required metrics and can be used in production. The model gets deployed to production to generate inferences for the production data. This pipeline can run on any three controllers' platforms, depending on the data source and availability of computing resources for the production model. Running training activities on Near-RT RIC is possible if, firstly, the production model runs on the RAN edge and enough hardware resources are available for this purpose. Secondly, SMO and Non-RT RIC are not responsible for maintaining a model repository for the Open RAN ecosystem.

2.6.2 Chain of models

In this pipeline, the ML/AI solution is a combination of more than one model, that run either pure sequentially or partly asynchronous and partly sequentially. The chain of models can be treated as one model and map the previous solution to this one, with one input data source. However, there are scenarios in which, one of the models is a multi-input model. A multi-input model requires some considerations on providing the required input data. Fig. 2.14 shows a group of models that either generate output for another model in the same group, or use the output of one of models as input. A common example of this case is using AI models for pre-processing or post-processing of the main inference logic.

2.6.3 Champion challenger and Online training

Champion challengers are trained models supposed to have a higher performance than the model currently running in the production. Online training refers to using the production stream of data to incrementally train the model and update the model to improve its performance.

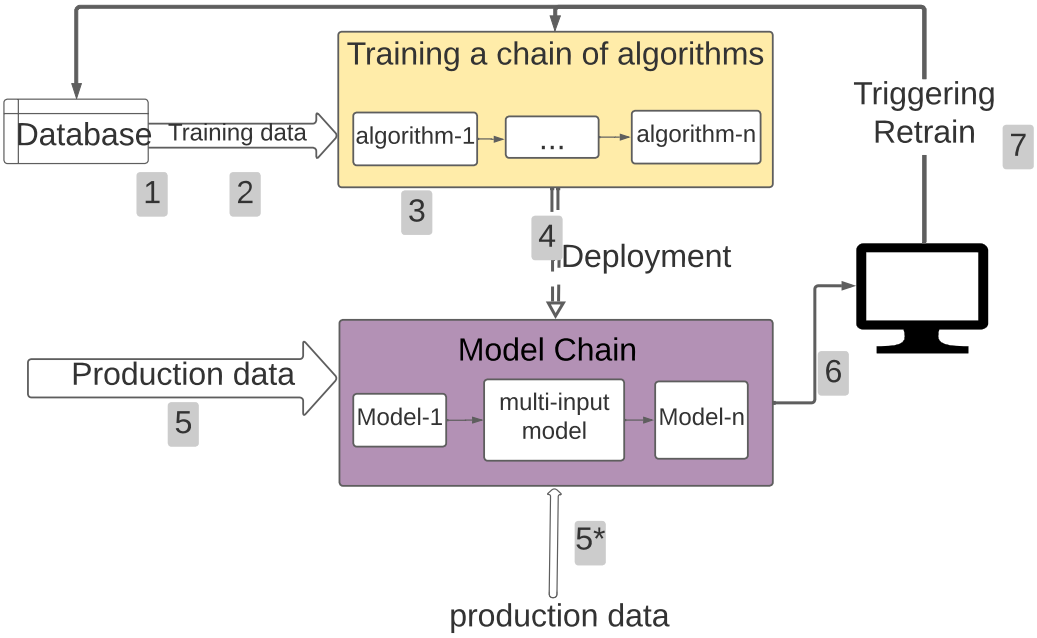


Figure 2.14: Model chain

Fig. 2.15 shows a diagram of online training combined with the concept of the champion challenger. Not all champion challengers are built from online training, but an online trained model is usually deployed as a champion challenger with the aim of not to adversely impacting the QoE. The traffic is split between the currently in production and the newly built models. After monitoring metrics and the new model's performance, the traffic will be routed entirely to the new model. The previously running model will be stored in an archive for future reference or any roll-back requirement.

2.6.4 A/B or Canary testing

In A/B or canary testing (Fig. 2.16) the goal is to compare the performance of two versions of the same model. These versions solve the same business problem with the same KPIs but are different, for instance in parameters, or the impact of a feature in the performance. In A/B testing, the ML pipeline splits the traffic between these two models (usually equally), and compares their performance in production. Then the traffic will be increased to 100% towards the superior one. The champion challenger scenario also can be called a case of A/B testing.

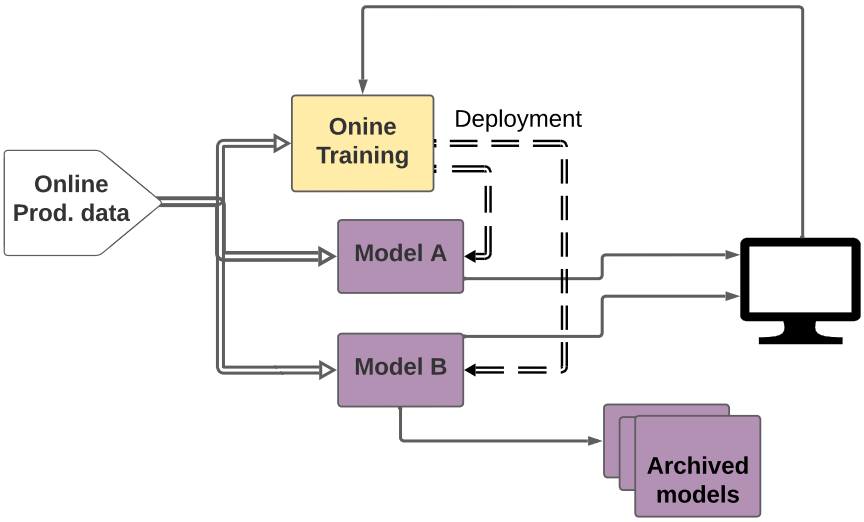


Figure 2.15: Online training with champion challenger

2.6.5 MLOPs and ORAN

As we can see in each of the pipelines, the ML orchestration system is responsible for maintaining an ongoing performance of the ML/AI model and securing all communications between the items in the process. Splitting a pipeline to run on more than one platform will introduce security risks, but running the whole pipeline on one platform requires enough resources to store and process data besides running intelligent applications.

The architecture of Open RAN, the life cycle of data in this architecture, and the diversity of ML applications in telecommunication make MLOps solutions on Open RAN different from other industries. Automation and deployment of ML models require specific considerations, especially regarding security, availability, and reliability. This requirement emphasizes a new branch in the telecommunication industry that focuses on MLOps.

2.7 Challenges and Opportunities

In this section we explore opportunities and challenges in 5G and B5G that using AI in an Open RAN architecture can solve.

2.7.1 Architectural Opportunities

I discussed the architecture of Open RAN and the variety of use cases that it supports. Open RAN introduces new interfaces such as A1 and new components such as Near-RT RIC. Moreover, combining virtualization and cloud-native architecture with Open RAN components gen-

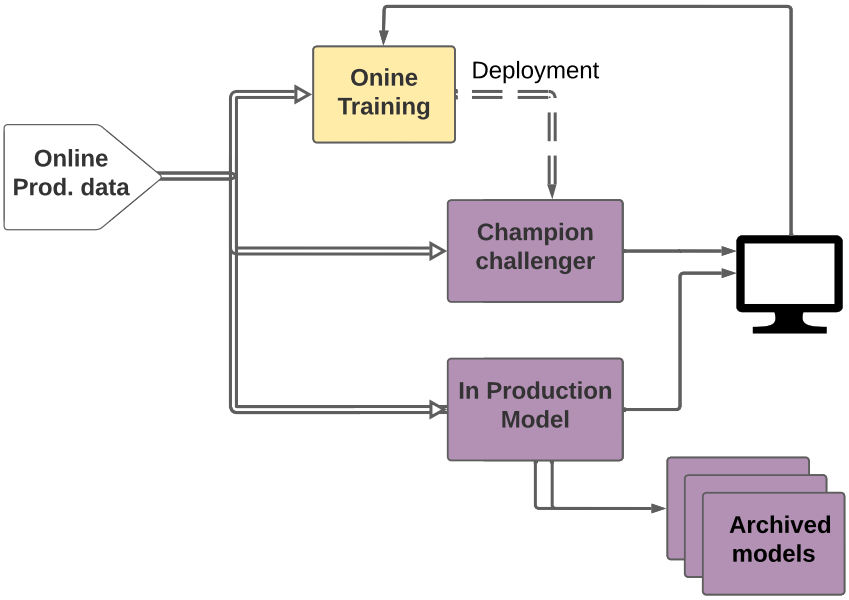


Figure 2.16: A/B and Canary testing

erates many different scenarios to deploy discussed in Section 2.4.2. As a result, there are many different architectural options to install an end-to-end Open RAN scenario. Even though cloud deployment has been in mainstream use for sometime, designers and operators face new challenges when implementing a cloud-based platform for Open RAN. One of the challenges is the deployment of functions on the cloud platform. The current specifications and standards allow virtual functions and container deployment on cloud platforms. Although it might add flexibility to specifications, it can reduce compatibility between vendors' components and operators' available platforms. The other challenge will be connecting components. Wired or wireless, with interfaces or via API calls, components communicate with each other and their environment. Designing a robust, reliable, and scalable system requires a detailed and thorough study of requirements. Especially the requirements of different regions are not the same. For instance, the countryside with camping facilities is quite different from an urban region, considering these regions also need to be connected. In a multi-vendor system such as Open RAN, products made by different vendors have to be deployed, maintained, and secured by operators. Although standards work on Open RAN interfaces, each operator can have different requirements and configurations, such as namespaces for Kubernetes deployment. Vendors who build components and products for one operator might face difficulties in offering the same products to other operators.

2.7.2 Non-terrestrial use cases

Many use cases need very low latency or are in geographical locations where a solid urban network system does not exist. Utilizing mobile networking components such as aerial and vehicle-based components can be helpful in these situations.

For example, in a use case where an accident has happened, but paramedics cannot access the injured people, drones can fly over and carry resources such as robotic hands and a mobile cloud platform to run applications. Another opportunity to use a non-terrestrial implementation of Open RAN is where operators temporarily face hot spots. They can use UAS to ensure an acceptable level of QoS or QoE (Quality of Experience) with a low cost.

The main controversial challenge for non-terrestrial solutions might be providing them with enough power supply. The current enhancement in solar panels on space platforms such as satellites can enhance the efficiency of non-terrestrial networks in terms of energy [97]. Vehicles also can play the role of RAN components, and even base stations in the future of cities [98].

2.7.3 Security

In Open RAN, splitting RAN components and opening interfaces in addition to accepting multi-vendor solutions will introduce vulnerabilities to the system [99]. Consequently, security experts need to focus on three major areas.

The first is the general telecommunication security problems, regardless of the RAN implementation. This one is a classic threat. In this security concern, network intrusion happens between UEs and RAN components. The DDOS attack, in which malicious actors target RAN and networking infrastructure through UEs, is a well-known example.

Secondly, security attacks are caused by Open RAN vulnerabilities. In Open RAN, many interfaces are open. Therefore, components made by different vendors can communicate. This feature makes a malicious actor capable of breaking into that communication if the interface is not managed correctly.

Thirdly, contaminated applications or components or faulty products can create intentional or unintentional threats to the system. In Open RAN, unlike previous versions of RAN, one vendor is not responsible for the end-to-end implementation and maintenance of the installed RAN. All vendors should follow the same security standards and tests before submitting any product for operation. Before deployment, operators can test products in a test environment to ensure products meet safety and quality requirements. In particular, implementing intelligent

solutions that can identify anomalies and unexpected activities in real-time is vital.
Especially for cross-component communications, there is a need to create algorithms, write programs and generate models which can monitor, identify and prevent malicious activities. This opportunity is open for researchers and security engineers to find creative and efficient solutions and methods. Using certified protocols can improve security but can increase latency. Networking operators also can cause shortcomings in the system if they do not apply strict measures to Open RAN implementation and engage vendors and third parties in deploying their products. This problem is preventable by adding processes to Open RAN installation specifications.
In summary, operators need a systematic and proactive approach for security threats in Open RAN. Also, they need to engage their vendors and monitor products actively.

2.7.4 Implementing ORAN in urban regions

In combination with 5G and B5G, Open RAN will generate opportunities for serving diverse use cases, from mobile broadband to URLLC, and it will profoundly impact the future of technologies. However, more diversity in requests, more complicated to manage communications. In particular, autonomous vehicles and mission-critical applications are inseparable parts of future cities. The biggest challenge of these use cases is that they cannot tolerate interruption. Building a robust and reliable RAN creates unlimited project topics for researchers and engineers. These projects should focus on creating communications that follow the highest standard and are secure and reliable with efficient resource management.
Moreover, any of those projects should be environmentally friendly. Achieving these requirements without compromising latency and other QoS KPIs is another continuous challenge to explore.
Another crucial problem that Open RAN systems in urban areas have to overcome is attenuation in 5G and B5G. In 5G and B5G wireless generations, wireless communication will be in the range of mmWave and THz. Transmission in these band can be significantly impacted by buildings, wall, and vehicles can create challenges for designing and implementing a reliable system in an urban environment, leading to signal blockage and link failures. Weather conditions also act as blockers and interrupt mmWave and THz communication, which add to this complexity. These are topics of research that researchers have been working on since the inception of 5G. The survey conducted in [100] discussed how some weather conditions such as

heavy rain, temperature, and fog could affect the quality of mmWave communications. [101] also has studied the impact of rain on 5G communication. Considering that rain is a common phenomenon in many places and some cities even experience heavy rainfall for months, its severe impact on millimeter-wave propagation and signal loss cannot be overlooked.
Another atmospheric condition that can severely impact a telecommunication system is a hurricane. In [102] hurricanes and rain are mentioned as severe threats for 5G communication.
Future cities and people living in them require a wide range of telecommunication use cases, and any interruption can cause severe damages and huge costs, not only a financial cost but also loss of lives. Therefore, building and optimizing Open RAN in an urban area requires ongoing innovation and optimization research and engineering works. The end goal is to run a reliable, scalable, and secure telecommunication system with robust contingency plans that can serve all requests with a high level of QoS and QoE in any weather condition.

2.7.5 Zero-touch Networks

Zero-touch networking represents a paradigm shift in how communication networks are managed and operated. It leverages artificial intelligence, machine learning, and closed-loop automation to create fully autonomous network systems that can configure, monitor, optimize, and heal themselves with minimal human intervention. The European Telecommunications Standards Institute (ETSI) defines zero-touch management as a fully autonomous network management solution with human oversight, where networks can reason about their current state, interpret information, and provide reconfiguration recommendations. This approach addresses the unprecedented complexity of next-generation wireless networks by automating critical operational processes including planning, deployment, provisioning, and monitoring. Through data analytics, predictive capabilities, and adaptive decision-making, zero-touch networks can continuously self-optimize to meet changing service requirements, reduce operational costs, enhance reliability, and accelerate service delivery. As 5G evolves toward 6G, this automation becomes essential to manage the increasingly heterogeneous and dynamic network environments supporting mission-critical applications [103].

Autonomous Networking

Over the past decades, networking controllers have evolved from hardware controllers to software-defined networks controllers and will soon enter the era of AI-controlled networks. The role

of AI is expanding the flexibility that SDN provides and adds elasticity, self-maintenance, and higher performance. *Autonomous networking* is a phrase that started with the introduction of SDN as a means to use software products to manage and orchestrate networking in a region or for a use case. The other term common in mobile networking to address a network system without predefined network infrastructure and parameters is Self-Organizing Network (SON), widely used in mobile networking. 3GPP has explained SON in detail, in TS 32.500 [104]. Autonomous Network and SON are not new concepts.The idea of Autonomous networking was introduced more than 20 years ago. As a result of advancements in hardware and ML/AI algorithms, both SON and Autonomous networking are taking advantage of AI algorithms [105],[106].

The ETSI organization has published a white paper [107], dedicated explicitly to end-to-end autonomous networks and more focused on the operation and maintenance of telecommunication. It defines four levels for autonomous networking. From level one to four, it gradually decreases the role of humans in tasks and increases the role of artificial intelligence. Artificial intelligence's responsibilities in the system grow at each level from being assistive tools and failure recovery systems based on logs and alerts to failure diagnosis applications. Eventually, in the fourth level, being intelligent decision-makers and predictors. In the fourth level, the system autonomously can take precautionary actions and be in charge of failure and fault prediction. Using Open RAN in operation, levels three and four will become more crucial. In Open RAN, there will not be any specific vendor responsible for the end-to-end implementation of RAN. Therefore, it is vital to have a system that accurately diagnoses the problem and maps it to the source. Also, incorporating an application that can predict faults and failures can give potentially affected operators and vendors time to take action in a reasonable time. Once more, ML/AI applications become inevitable tools of future networking systems.

2.7.6 Management

Each of the individual topics mentioned before is a management challenge and opportunity. Operators and vendors can solve many of these problems using products and research results that researchers and engineers have created for other RAN implementations. Nevertheless, some problems are unique to Open RAN. One of the unique problems is managing different vendors and their products. The multi-vendor concept helps operators take advantage of the

competitive market and negotiate price and quality with their providers. However, it introduces many challenges.

Firstly, ensuring that products meet the required quality, pass security measures, and are compatible with other operators that run products in their system. Experimentations and trials, using testbeds before deployment, universal specifications and standards, and engaging stakeholders and vendors in discussions can resolve or minimize this problem.

Secondly, open interfaces generate opportunities for malicious actors, and operators maintain communications' security intact. In addition to in-depth investigation and test on purchased components, they need to implement ML/AI applications and other intelligent systems to predict and detect flaws or malicious activities in real-time and react in real-time.

Thirdly, a RAN system's end-to-end maintenance and responsibility is not the vendors' responsibility anymore. Operators should establish procedures covering all parameters, communications, and activities for a test and monitoring. In addition, they need an intelligent application that can quickly map any flaw or problem to its possible sources. In this way, operators can quickly bring the related vendor on board and resolve the issue.

Also, in Open RAN, three intelligent controllers generate massive opportunities to optimize and innovate for telecommunication applications and use cases. Its orchestration has many use cases for software engineers to implement the best practices on deploying zero-touch data or ML pipelines and set up the best practices for deployment, continuous delivery, reliability, and scalability of software products in Open RAN.

Moreover, conflict resolution scenarios will become challenging as more applications for different telecommunication use cases run on the RIC platforms. Applications can have compatibility issues with the running infrastructure, or have conflicts on optimization and adjustment of parameters in their target network. If two AI applications directly or indirectly try to change the same parameter or change QoE of the same UE, there needs to be an intelligent system to prevent or resolve this issue.

As we can see, managing an Open RAN system requires a strong collaboration between many sectors in engineering and business. This efficient collaboration becomes more crucial, considering that an operator may not be able to implement what they build in one area to another because of many factors such as populations, diversity of use cases, type of user requests, different regulations, and different weather conditions.

2.7.7 Digital Twin for Test and Improvement

Operators need to create a test environment to monitor the quality and reliability of products before integrating them into a running system. The current test environments are based on historical data. The ideal test could be monitoring the applications in the currently running environment while not deploying them into the actual production environment. A digital twin environment can assist with this problem. Based on [108] "A digital twin is a virtual representation of an object or system that spans its lifecycle, is updated from real-time data, and uses simulation, machine learning, and reasoning to help decision-making.". As a result, applications can run in a digital twin environment and show how they behave in different circumstances.
The other benefit of creating a digital twin will be troubleshooting and fault finding in applications or components. If the outcome of a running application is not as expected, the digital twin environment can repeat the scenario, and operators can find the source of the issue or optimize parameters to prevent the same poor outcomes.
Creating digital twins increases the required resources, but by advancing hardware design and implementing data lifecycle policies, the benefits of utilizing digital twins, such as enhancing security and troubleshooting activities, can outweigh its cost for operators.

2.7.8 Energy conservation

In end-to-end telecommunication, the RAN consumes more than 80% of the wireless network power [109]. As a result of the growth in networking systems and their traffic which generate a large amount of data, energy consumption increases. Therefore, reducing the energy consumption in the RAN is a challenge in 5G and B5G that AI can address. For instance, intelligently allocating tasks to servers can economize energy consumption in the RAN. In [110] authors proposed an algorithm to optimize the offloading selection, radio resource allocation, and computational resource allocation in Mobile Edge Computing (MEC). In [111] researchers designed an autonomous control method to reduce the energy consumption of networking systems. [112] proposes a green cellular network by using Markov chain and modeling possible load variation. This information is used to select the most efficient base station. In sum, AI applications can help operators use the available resources efficiently with total capacity. Also, they can reduce redundancy in the system, which consequently can contribute to creating a green wireless networking system. Radio Frequency devices, the RAN components, and AI algorithms can be improved to conserve energy.

2.8 Conclusion

This literature review has provided a comprehensive overview of the evolution of Radio Access Networks, the emergence of Open RAN architecture, and the increasingly important role of artificial intelligence in modern wireless networks. It has highlighted the significant potential for AI applications, particularly in the form of xApps running on the Near-RT RIC component, to optimize network performance and enable new capabilities

However, while the opportunities for AI in Open RAN are substantial, this thesis argues that the current state of the Near-RT RIC is not sufficient to fully support the demanding requirements of Ultra-Reliable Low-Latency Communication (URLLC) use cases. The stringent reliability, latency, and resilience needs of applications like autonomous vehicles, remote surgery, and industrial automation push the boundaries of what traditional RAN architectures and control paradigms can achieve.

This thesis posits that by innovating across different areas the Near-RT RIC can evolve into a true enabler for URLLC within the Open RAN paradigm. The subsequent chapters will delve into each of the enhancement dimensions, proposing novel frameworks and experimental validations to substantiate this central argument.

As 5G and beyond networks continue to push the envelope of wireless capabilities, rising to the challenge of URLLC support will be critical. By critiquing the current state of the art and charting a path forward, this thesis aims to contribute meaningfully to the realization of AI-enabled, URLLC-ready Open RAN architectures. The fusion of AI and RAN holds immense promise, but realizing that potential for the most demanding use cases will require a concerted effort to enhance and evolve the underlying control structures and mechanisms. It is this transformation that the remainder of this thesis will explore in depth.

Table 2.2: Non Open RAN Specific Components in Open RAN Architecture

Component	**Description**	**Origin**	**Key Functions**
NMS	Network Management System for O-RU	Legacy RAN	Support legacy Open Fronthaul M-Plane deployments
O-eNB	Hardware aspect of 4G RAN	4G/LTE	Communicates with Near RT RIC over E2
gNB	Next Generation NodeB	5G	Base station in 5G; includes CU and DU functionalities
O-CU	O-RAN Centralized Unit	5G	• RRC and PDCP protocols • Mobility Control • RAN sharing • Session management
O-DU	O-RAN Distributed Unit	5G	• RLC/MAC/High-PHY layers • Real-time scheduling • Based on lower layer functional split
O-RU	O-RAN Radio Unit	5G	• Low-PHY layer and RF processing • Similar to RRH but includes Low-PHY layer

Interfaces			
Interface	Components	Origin	Role
A1	Non-RT, Near-RT RIC	O-RAN Alliance	Application deployment, Policy control and management
E2	Near-RT, Node	O-RAN Alliance	Communication between edge and Near-RT
Fronthaul	O-RU, O-DU, SMO	O-RAN Alliance	Communication between O-RU, and O-DU/SMO
O1	SMO	O-RAN Alliance	Between SMO and other ORAN components
O2	SMO, Cloud	O-RAN Alliance	Communication between Core and SMO
E1	CU-CP,CU-UP	3GPP	Communication between CU-CP and CU-UP
X2/Xn/NG	Node components	3GPP	Legacy responsibilities in previous RANs

Table 2.3: List of interfaces in Open RAN

Learning method	Algorithm	Sample papers
Supervised Learning	SVM security and time-series	[37],[38],[39],[40]
	Logistic regression	[41],[42],[43]
	Naive Bayes	[44],[45],[46],[47]
	Decision trees	[48],[49],[50]
	K-Nearest neighbor	[51]
	Neural networks	[52],[53]
	Random Forest	[54],[55]
Unsupervised Learning	Clustering algorithms	[56],[57],[58],[59]
	Data mining algorithms	[60],[61]
Reinforcement Learning	Associative, Inverse, Safe	[62],[63],[64],[65],[66],[67]

Table 2.4: ML/AI in telecommunication

Chapter 3

Optimizing RAN's Reliability with Multi-Objective xApps

3.1 Introduction

This chapter addresses the first research question of this thesis. This chapter explores how classic control functions running on Near-RT RIC can be theoretically reformulated to support URLLC requirements.

Communication control actions often face failures for various reasons. Traditional approaches typically optimize a single parameter and handle failures through retry mechanisms - attempting the action again with either the same or a modified decision until success is achieved. Hand Over (HO), one of Near-RT RIC's key use cases, exemplifies this challenge. While the HO process aims to enhance user experience by transferring an active user equipment (UE) from one cell to another to improve received signal strength, the current HO approach becomes problematic for URLLC applications. The current HO methods adhere to the prescribed procedure outlined in 3GPP documentation [113], primarily focusing on improving the RSS. However, the omission of metrics like outage probability can result in costly disruptions. Given that reliability is a fundamental requirement for URLLC use cases, the conventional concept of retrying failed handovers becomes unacceptable for these scenarios, necessitating a new approach that ensures reliability from the initial attempt.

This thesis proposes Pareto-optimal decisions as one solution to this challenge. Pareto-optimal decisions aim to optimize multiple parameters simultaneously, rather than maximizing a single

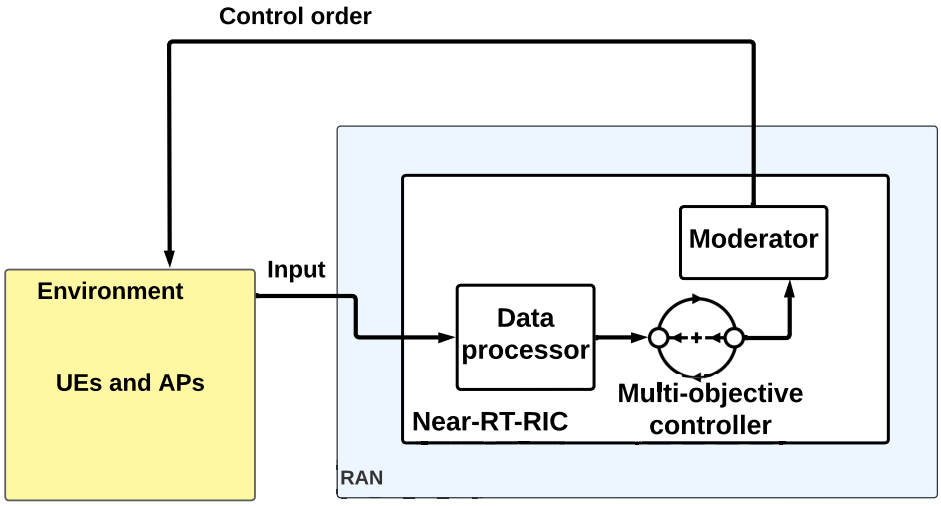


Figure 3.1: A Multi-Objective Framework to Enhance Reliability

primary parameter and resorting to retry actions when secondary parameters cause failures. In the context of URLLC applications, where reliability is as crucial as performance, this means considering both signal strength and reliability in the initial decision-making process, rather than optimizing signal strength alone and handling reliability failures through retries. Fig. 3.1 presents a framework with the multi-objective controller accompanied by its pre-processor and post-decision making moderator. To achieve the multi-objective decision making goal, this thesis proposes a framework that uses Reinforcement learning (RL) algorithms. RL is a learning paradigm that allows agents to learn from experience and maximize rewards [114]. However, in some cases, agents must simultaneously achieve multiple objectives, leading to Pareto optimal policies in multi-objective reinforcement learning (MORL) [115].

While previous research has used deep RL to optimize time and frequency allocation in near-real-time use cases [116], these approaches typically focus on single objectives like RSS maximization or employ agents on UEs. This chapter proposes a novel approach using Pareto-optimal decisions at the network level to simultaneously optimize multiple objectives critical for URLLC applications. This approach becomes particularly critical in 5G/6G wireless networks, where high frequencies affect signal propagation, reducing coverage and increasing vulnerability to radio frequency (RF) blockers. These characteristics necessitate micro-cell deployments, often with multiple overlapping cells providing multiple access points (APs) for UEs to connect to, making handover decisions more complex and reliability more crucial. These characteristics make the HO controller a great candidate with which to validate our framework.

3.2 Handover, Reliability and Quality of Experience

Handover (HO) is the process of transferring a call and data session between cells as a user moves within a cellular system's coverage area [117]. HO algorithm design and optimization have been researched extensively [118]–[121], with approaches falling into categories like Maximum Received Power (MRP), Context-aware, Cost-function, and Fuzzy-logic [122]. 5G networks use measurement reports for HO actions [117], but 3GPP recommends considering traffic criteria [123], especially for critical applications like remote surgery. AI methods have been explored to address HO challenges, but they often require extensive data [124]–[127]. This chapter introduces HORLA (HandOver Reinforcement Learning Application), an xApp running on the Near-RT-RIC component in the Open RAN architecture. HORLA aims to enhance Reference Signal Receive Power (RSRP) and handle potential outages using UE measurement reports and interaction with access points. The first Key Performance Indicator (KPI) is the difference between the target RSRP and the serving RSRP, $RSRP_{ik} - RSRP_{ij}$. UE with the index i experiences $RSRP_{ik}$ as the received power from micro-cell k, and $RSRP_{ij}$ as the received power from micro-cell j.

$$P_r = P_T - PL \tag{3.1}$$

P_r is the received power in dBm after handover from the target cell, P_T is the power in the source, and PL is the lost power before reaching the UE. The experiment uses the Hata model path loss for urban environments [128], Eq.(3.2)

$$\begin{aligned} PL_j(dBm) = 69.66 + 26.16log_{10}f - 13.82 \\ log_{10}h_B - C_H + [44.9 - 6.55log_{10}h_B]log_{10}d \end{aligned} \tag{3.2}$$

C_H is an empirical coefficient based on the UE density and the transmit frequency. The calculations in this chapter use the following value for C_H related to the dense urban regions and high-frequency communications.

$$C_H = 3.2(log_{10}(11.75h_M))^2 - 4.97 \tag{3.3}$$

Where h_M is the height of the device. h_B is the height of the power base station. As a result, the goal is to maximize the difference between the target RSRP and the received RSRP. Eq. 3.4 shows this goal's success rate. n is the number of APs observable by a UE being served by an AP with power P_s.

$$\frac{P_{target-cell} - P_s}{Max[(P_0 - P_s), ...(P_n - Ps)]} \tag{3.4}$$

The probability of experiencing errorless communication is related to the value of Eq. 3.4. This experiment uses the Shannon-Hartley theorem, Eq. 3.5, to address the outage problem.

$$C = B \log_2(1 + \frac{S}{N}) \tag{3.5}$$

In Eq. 3.5, C is the channel capacity, B is the signal bandwidth, S is the signal power, and N represents noise. $\frac{C}{B}$ is the maximum data rate that a device can experience considering the signal power and noise in that state (Eq. 3.6).

$$R_{max} = \log_2(1 + \frac{S}{N}) \tag{3.6}$$

Therefore, to guarantee the QoE for a device, the maximum data rate should be more than the required data rate R_r by a device. In Eq.3.6, $\frac{S}{N}$ is SNR. This chapter uses SINR (Signal to Interference and Noise Ratio), which is analogous to SNR. In SINR, the received power is divided by I, the sum of interference power, and noise, N (Eq. 3.7).

$$SINR = \frac{SignalPower}{I + N} \tag{3.7}$$

or more specifically [129] :

$$SINR_i = \frac{h_{ki} P_i}{\sum[h_{kj} Pj] + N_k} \tag{3.8}$$

$$N_k = -174 + 10 * \log 10(B) + N_o \tag{3.9}$$

In Eq. 3.8, $SINR_i$ is the value of $h_{ki}P_i$, faded received power from AP_i by user k, divided by the sum of faded received power from any other transmitter to the user k and any source of noise experienced by user k. In Eq. 3.9, B refers to bandwidth and N_o refers to other sources of Noise. To achieve successful communication, the value of Eq. 3.6 must be more than the UE threshold.

$$\log_2(1 + SINR) > R_r \tag{3.10}$$

Then

$$(1 + SINR) - 2^{R_r} > 0 \tag{3.11}$$

So to avoid an HO failure, the decision maker needs to ensure the Eq. 3.11 is met while improving the value of RSRP. In other words, the agent is solving the problem with a pareto-optimal solution.

In Eq. 3.12, H_{uij} is the controller's objective that has to be maximized for the UE u at the time of t after transferring the device u from the serving cell i to the serving cell j. Also, $P_{Max(dif)}$

is $Max[(P_0 - P_s), ...(P_n - Ps)]$, the maximum possible improvement in RSRP by taking the handover action in this state.

$$H_{uij} = \frac{(RSRP_j - RSRP_i)}{RSRP_{Max(dif)_i}}((1 + SINR) - 2^{R_r}) \quad (3.12)$$

$$max \int_{t=0}^{\infty} \sum_{u=0}^{W} \sum_{j=0}^{M} \sum_{i=0}^{N} = \frac{(RSRP_j - RSRP_i)}{RSRP_{Max(dif)}}((1 + SINR) - 2^{R_r}) \quad (3.13)$$

This experiment combines the rewards of two independent objectives by multiplying them. HORLA's main goal is to select a target cell with better power, and the second objective is to ensure system reliability and availability. The MORL agent is trained to maximize the value of Eq. 3.12. Fig. 3.2 presents a sample area with micro-cells as static black circles and UE as red triangles in different time steps. The environment transitions from the state A to the state B by taking a HO action on this environment.

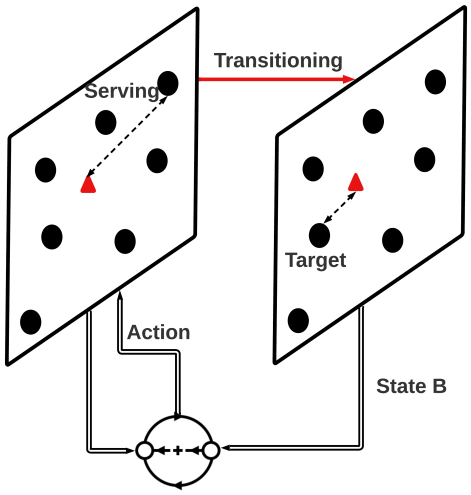


Figure 3.2: MDP Modeling of Handover in Wireless Communication

3.3 The big picture of HORLA

In our proposed model, multiple HO agents collaborate to manage wireless communication between devices and access points. Each agent oversees a specific area and communicates during conflicts or emergencies. These agents operate on the RAN side, avoiding issues with compromised user equipment and hardware-software compatibility. See Fig. 3.3 for a high-level diagram of the multi-agent model, the parent of HORLA.

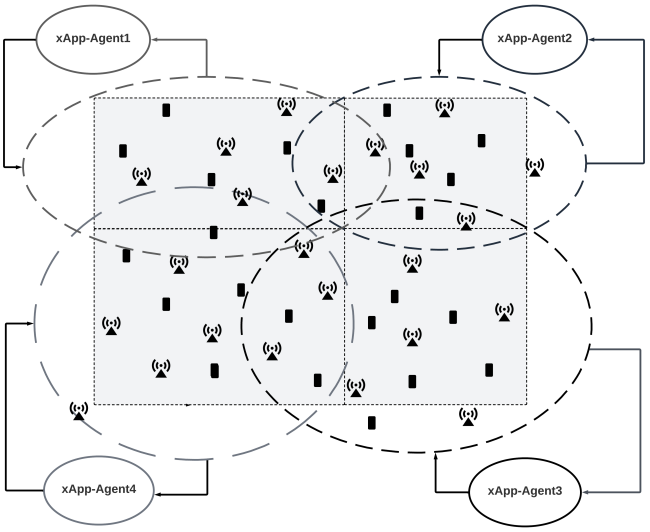


Figure 3.3: A schematic arrangement of APs and HO agents

3.4 The architecture of HORLA

RL has been explored in previous studies for HO optimization, considering aspects such as energy consumption, satellite stations as APs, and UAVs as UEs [65], [130], [131]. For instance, one study [122] compared Q-Learning and Sarsa, two RL algorithms, to maximize received signal strength (RSS). In another study [132], RL techniques were employed to identify the maximum RSS value. Another paper [133] used a Bandit Arm algorithm to optimize RSS within a wireless communication system. Additionally, a multi-objective approach was utilized in [65] to optimize HO allocation and power control with RL agents residing on UEs. In [134], a predictive approach was proposed for the handover process in wireless communication systems. Lastly, the study in [67] introduced a competitive multi-agent approach by deploying agents on UEs. However, this thesis leverages multi-objective RL capabilities for enhancing Near-RT RIC xApps.

HORLA's HO process introduces a novel approach for analyzing UE measurement reports. Unlike traditional methods, it doesn't process reports directly at the serving base station. Instead, they are sent to the xApp on the Near-RT RIC platform, with HORLA as the core module. The RL model in HORLA analyzes these reports to select the target cell for HO. The data processor handles scanning and message preparation, and the RL agent (HORLA) uses UE-reported data like $RSRP$, $SINR$, and R for decision-making. A "Moderator" component communicates this decision to APs. Fig. 3.4 illustrates HORLA's architecture, including the data processor, HORLA agent, and moderator components. The base of HORLA is a Deep

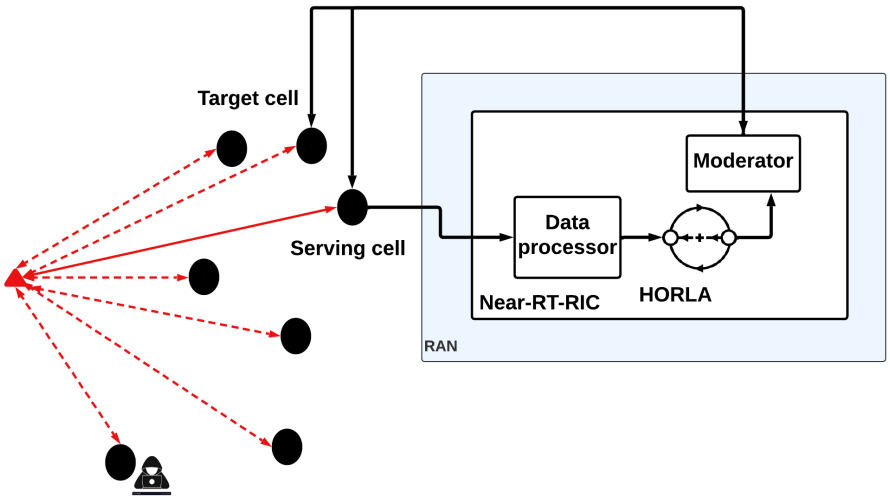


Figure 3.4: HORLA running on a Near-RT RIC platform controlling the HO process

networking RL model that takes $2n + 2$ inputs for the input layer, where n represents the number of micro-cells within the agent's designated territory. The output layer has a size of n. The design of the state, action, and reward is as follows.

- The state consists of RSRP values and SINR values reported by the UE for multiple base stations, the serving base station, and the required data rate threshold for that UE. For each UE ue_u, the state is defined as $S_u = (P_{u1}, ..., P_{un}, M_{u1}, ..., M_{un}, b_{uj}, R_u)$. Here, n is the total number of accessible base stations, P_{uk} and M_{uk} are RSRP and SINR for base station k, b_{uj} is the serving base station, and R_u is the required data rate threshold for UE ue_u.

- Action: An action is the selection of the target AP for UE handover.

- The agent's goal is to optimize target cell selection success in handover. The reward design considers decision risks and incorporates probabilities for positive rewards despite potential negative outcomes like lower RSRP. The reward is conditional: if the selected RSRP is lower than the serving RSRP, the agent receives a fixed reward of -10, encouraging decisions that improve signal strength, which is the main reason for handover requests.

$$Reward^{t}_{uij} = \begin{cases} -10 & \text{if } (RSRP_j - RSRP_i) < 0 \\ Huij & \text{if } (RSRP_j - RSRP_i) >= 0 \end{cases} \tag{3.14}$$

HORLA's integration into existing Open RAN systems leverages the standard xApp deployment framework within the Near-RT RIC platform. The system interfaces with the broader

RAN environment through established O-RAN protocols, specifically utilizing O1, A1, and E2 interfaces. UE measurement reports are received through the E2 interface, while handover commands are transmitted back to the O-CU/O-DU components through the same channel, ensuring seamless integration with existing network flows.

The experimental version of HORLA is responsible for managing 19 access points. Based on its design explained above, the network will have 40 input and 19 output nodes. From a resource requirement perspective, HORLA's neural network architecture is deliberately designed to be lightweight. The version used for this experiment requires minimal storage space of approximately 100KB. The runtime memory footprint remains within bounds suitable for edge deployment, typically requiring around 500MB of memory. This efficient design ensures that the system can operate within the Near-RT RIC's sub-second latency requirements without introducing additional signaling overhead, as it utilizes existing measurement reports.

The operational workflow maintains consistency with standard RAN procedures. When measurement reports arrive from UEs through established RAN interfaces, HORLA processes these inputs through its pre-trained neural network. The system generates handover decisions within the strict latency constraints of the Near-RT RIC (less than 1 second), and communicates these decisions through standard O-RAN interfaces. This integration approach ensures that HORLA enhances existing network operations without disrupting established protocols or requiring significant infrastructure modifications.

Regarding scalability, HORLA's architecture supports network growth through parallel deployment options. The system's resource utilization demonstrates linear scaling with respect to the number of managed cells, allowing for efficient resource allocation based on network demands. The implementation supports standard containerization practices, enabling flexible deployment across varying network configurations and loads.

This integration strategy which includes containerization with compatible inputs and outputs ensures that HORLA can enhance handover operations while maintaining compatibility with existing Open RAN deployments and meeting the stringent performance requirements of modern wireless networks.

3.4.1 Algorithm Selection Analysis

The selection of an appropriate reinforcement learning algorithm for handover optimization in near real-time environments requires careful consideration of both the problem characteristics

and the strengths and limitations of various algorithms. After thorough analysis, Deep Q-Network (DQN) emerged as the most suitable choice for this specific use case. This subsection presents a detailed analysis of why DQN was selected and why alternative approaches were deemed less appropriate.

Advantages of DQN for Handover Optimization

The handover optimization problem fundamentally involves selecting a target cell from a finite set of available access points. This discrete action space aligns perfectly with DQN's architecture, which excels at problems requiring selection from a fixed set of actions. While algorithms like Deep Deterministic Policy Gradient (DDPG) or Soft Actor-Critic (SAC) are designed for continuous action spaces, they would introduce unnecessary complexity for our discrete selection problem.

The state space in handover optimization encompasses multiple critical variables including RSRP values, SINR measurements, serving base station information, and UE thresholds. DQN's neural network architecture effectively handles this high-dimensional state space through function approximation, capturing complex relationships between variables that would be difficult to model using traditional Q-learning approaches.

In handover scenarios, the success or failure of a decision isn't immediately apparent - there's a temporal delay between making a handover decision and observing its outcomes regarding outages or signal strength improvements. DQN's experience replay mechanism is particularly well-suited for handling these delayed rewards, effectively correlating actions with their long-term consequences.

Limitations of Alternative Algorithms

SARSA's on-policy nature presents significant limitations for handover optimization. As an on-policy algorithm that evaluates the same policy it follows, SARSA would be inherently less exploratory in discovering optimal handover opportunities. This characteristic is particularly problematic in near real-time environments where we need to carefully balance exploration and exploitation.

The Asynchronous Advantage Actor-Critic (A3C) algorithm introduces unnecessary complexity for handover optimization. Its architecture, which employs multiple agents training in parallel,

would create excessive overhead in the Near-RT RIC component of our system. Moreover, A3C's strengths in handling continuous action spaces are irrelevant for our discrete handover decisions.

Proximal Policy Optimization (PPO) presents several significant drawbacks for our use case. Its sample inefficiency is particularly problematic in near real-time communication scenarios, where we cannot afford to collect extensive handover failure data before achieving reliable performance. Additionally, PPO's higher computational resource requirements make it less suitable for real-time handover decisions in RAN infrastructure.

Implementation and Multi-objective Considerations

DQN's implementation advantages extend beyond its theoretical strengths. The algorithm's experience replay buffer proves invaluable for handover optimization by enabling learning from rare failure cases and making efficient use of historical data. The separate target network ensures stability that's crucial for near real-time communication reliability, while the overall architecture remains computationally efficient for RAN deployment.

While DQN is traditionally single-objective, its adaptation to multi-objective scenarios is straightforward through careful reward function design. Our implementation demonstrates this flexibility by combining signal strength and outage probability objectives into a unified reward structure. This adaptability is crucial for handover optimization, where multiple performance metrics must be balanced simultaneously.

From a practical deployment perspective, DQN offers significant advantages. Once trained, the model can execute efficiently, making it suitable for real-time handover decisions. Its architecture integrates well with the Near-RT RIC component in Open RAN, and the model can be continuously updated with new experiences while maintaining performance stability.

The selection of DQN for handover optimization in near real-time environments represents a careful balance of theoretical capabilities and practical requirements. While alternatives like SARSA, A3C, and PPO offer certain advantages, they each present limitations that make them less suitable for this specific use case. DQN's combination of discrete action handling, stability features, and implementation efficiency, along with its adaptability to multi-objective optimization, makes it the optimal choice for reliable handover optimization in next-generation wireless networks.

frequency	1800 MHz
Power	46 dBm
Access points Height	10 m
No. of access points	19
R	[0.3,0.6,1]
Antenna pattern	Hexagonal
Noise	7dB
Learning rate	1e4
End eps	0.1
eps decay	0.99

Table 3.1: Parameters and variable values used in the experiment

3.5 Training HORLA

The values of variables used in the experiment are presented in Table 3.1. The simulation parameters were chosen to reflect typical urban cellular network deployments. The frequency of 1800 MHz represents a common band used in urban cellular networks, offering a good balance between coverage and capacity. The hexagonal antenna pattern follows standard cellular planning practices, while the 46 dBm transmission power aligns with typical micro-cell base station specifications. The 19 access points were arranged to provide overlapping coverage, simulating the dense deployments characteristic of urban environments where handover optimization is most critical. The learning rate and epsilon decay values were selected through empirical testing to balance exploration and exploitation in the reinforcement learning process, while the noise figure of 7dB represents typical environmental interference levels in urban settings. The simulated environment is made of 19 access points arranged in a hexagonal pattern based on Report ITU-R M.2135-1 [135]. For this experiment, the transmit power from all access points is the same. However, in practical scenarios, micro-cells may have varying transmit power. Reinforcement learning faces the challenge of forgetfulness when a model's performance declines in subsequent episodes after optimization. This phenomenon, also known as catastrophic for-

getting, occurs when neural networks lose previously learned information as they acquire new knowledge. In the context of handover optimization, this means that while the model might learn to handle certain network scenarios effectively, it could gradually lose this capability as it adapts to new situations. This is particularly problematic in wireless networks where maintaining consistent performance across all scenarios is crucial for reliable operation. To address this, this chapter modifies the Deep Q-Learning (DQN) algorithm to periodically assess the model's accuracy. If accuracy exceeds a threshold, the program saves parameters as checkpoints and tests them in a separate environment. High-performing checkpoints are stored, and the best one is selected for production at the end of training episodes. This approach ensures the final model maintains consistent performance across the full range of network scenarios it was trained on. See Fig. 3.5 for the algorithm.

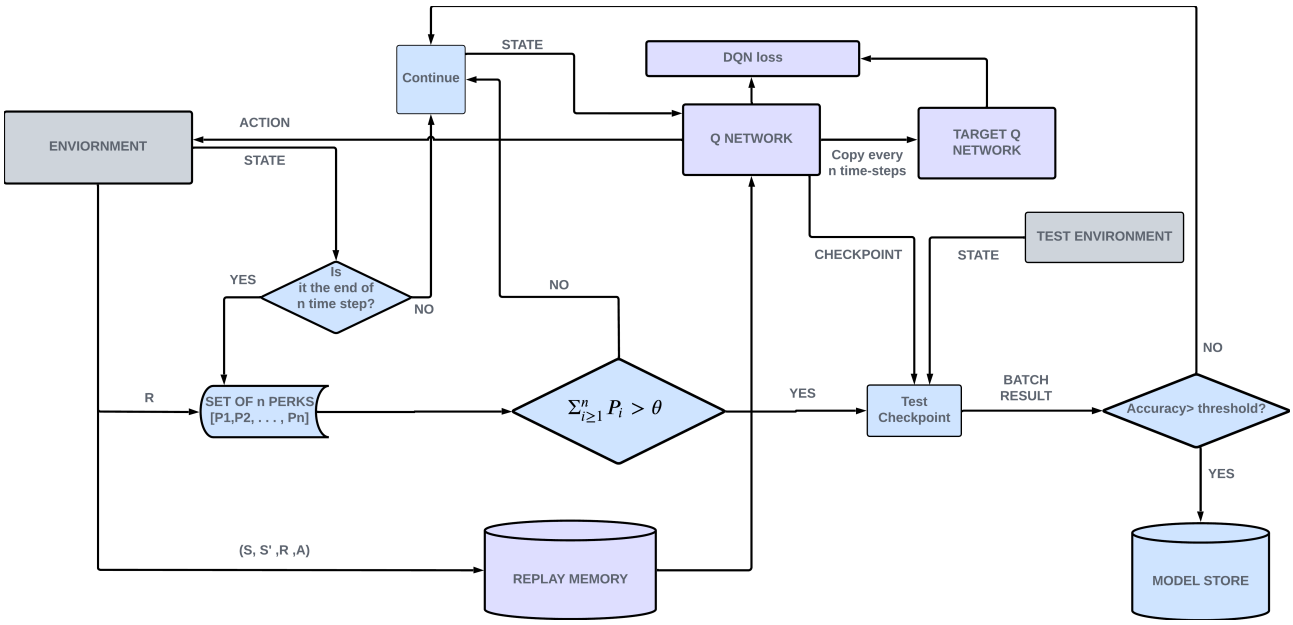


Figure 3.5: HORLA Training process to combat forgetfulness

3.6 Results

In the experiment, the testing environment differs from the training environment, and two baselines are used for benchmarking. The first baseline, referred to as MRP, follows 3GPP documentation triggers based on received power values for handovers. The second baseline, referred to as LOP (Least Outage Probability), selects the access point with the best unlikelihood of outage events value. MRP was selected as it represents the current industry standard

approach defined in 3GPP specifications, making it a crucial benchmark for any proposed handover optimization solution. LOP was chosen to represent approaches that prioritize reliability over signal strength, providing a contrasting baseline that emphasizes different network objectives. MRP's primary limitation is its single-objective focus on received power, which can lead to suboptimal decisions when reliability is also critical. While MRP performs well in scenarios with clear signal strength differentials, it may trigger unnecessary handovers in situations where the reliability impact outweighs marginal RSRP improvements. LOP, conversely, may be overly conservative in its handover decisions, potentially missing opportunities to improve user experience when signal strength gains could be achieved with minimal reliability risk. In the MRP baseline, any HO causing an outage is marked as a failure, while in the LOP baseline, HO attempts not improving received power are considered failures. HORLA must meet both criteria to succeed. In other words, a handover attempt is considered a failure under the following conditions:

- For MRP baseline: When an outage occurs after handover, defined as when SINR falls below the threshold required for the UE's data rate (when (1 + SINR) - $2_r^R \leq 0$)
- For LOP baseline: When the target cell's RSRP is not higher than the serving cell's RSRP (when $RSRP_target \leq RSRP_serving$)
- For HORLA: When either of the above conditions occur, as HORLA must simultaneously maintain both signal strength and reliability requirements.

The experiment generated batches of UEs with different data rate thresholds for each testing episode, running each episode at least 10 times with different seed values. The simulation followed a Monte Carlo approach with ten runs using different seed values, where UEs with varying data rate thresholds sent HO requests to their serving AP. Fig. 3.6 presents a bar chart showing mean failed HO attempts for different UE batch sizes, considering both controllers. HORLA aims to improve both RSS and system reliability, while MRP primarily focuses on RSS. HORLA outperforms MRP, reducing failures by over 40%. In summary, HORLA's consideration of multiple objectives leads to superior performance compared to MRP, which primarily prioritizes RSS improvement. This experiment and the results shown in Figs. 3.6, 3.7a and 3.7b conclude that the proposed MORL agent running on the Near-RT RIC component of the Open RAN improved reliability in all scenarios that MRP found challenging. The considerable difference between MRP and HORLA puts HORLA ahead of the competition in near real-time

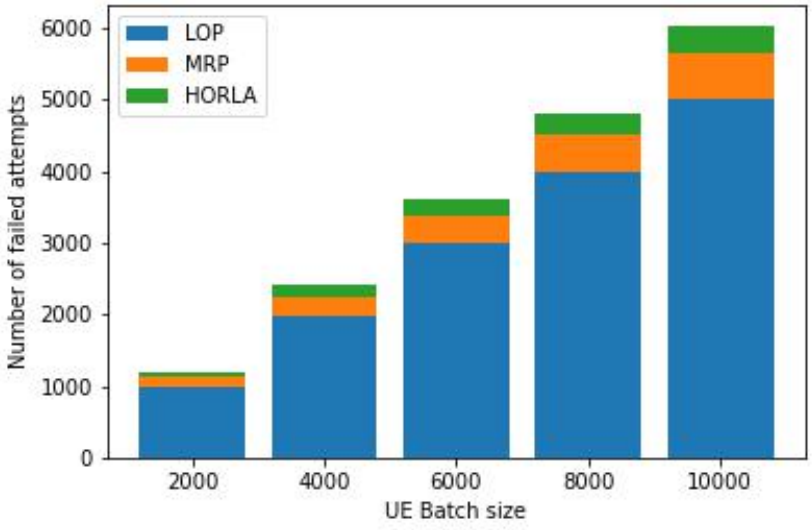


Figure 3.6: Failure comparison for LOP, MRP and HORLA

use cases and any use cases with a low tolerance for failure in communication.

3.7 Further discussion

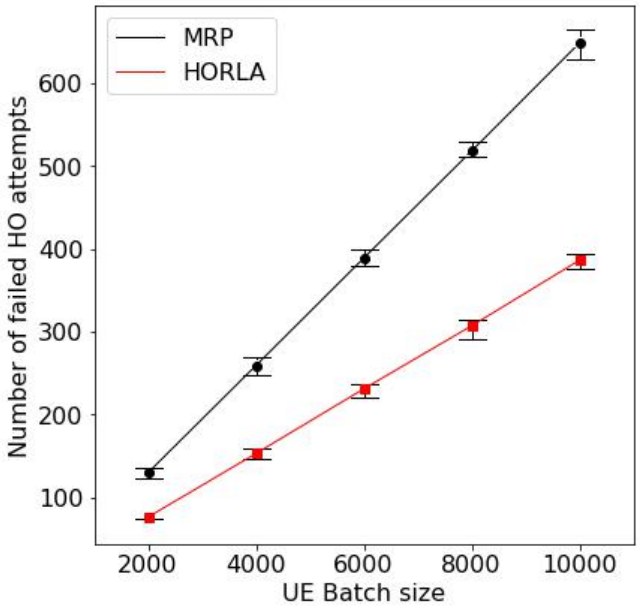
This section discusses an added benefit of the proposed HO solution and practical implementation considerations.

3.7.1 Preserving energy

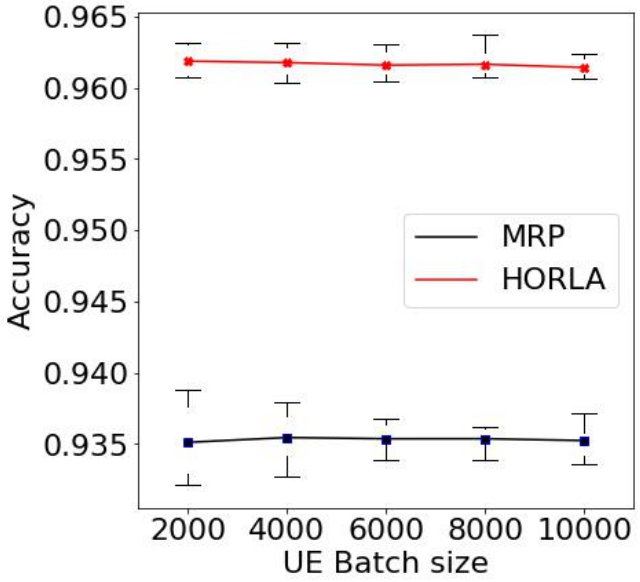
The proposed solution reduced energy consumption per time step by minimizing failed handover (HO) processes. Each failed HO attempt consumes additional energy due to the need to resend measurement reports and requests to the serving cell. Let m_t represent the number of failed requests at time step t, and $PU_{\delta T}$ be the energy consumed for processing each request. Based on Fig. 3.7b, MRP had an average failure rate of 7%, while HORLA had a lower rate of 4%, indicating 0.57 times fewer failures for HORLA. Consequently, the total energy consumed for request processing with HORLA and MRP can be expressed as:

$$\sum_{i=1}^{m_t} PU^{s}_{i\delta T}|_{HORLA} \approx 0.57 * \sum_{i=1}^{m_t} PU^{s}_{i\delta T}|_{MRP} \tag{3.15}$$

Where s = {Measurement request, HO Communication, HO Confirmed}. If we designate the unit of energy used for each device as "s," the cumulative wasted energy over time at each point



(a)



(b)

Figure 3.7: Comparison between HORLA and MRP.(a)Failed attempts, (b)Successful attempts

in time of T can be calculated as follows:

$$\sum_{t=1}^{T} \sum_{i=1}^{m_t} PU_{it}^{s} \tag{3.16}$$

Fig. 3.8 shows the cumulative difference in wasted energy between MRP and HORLA over 10 time steps. Each step involves processing HO requests from a random number of UEs (100 to 1000) to APs, excluding retries from previous failed attempts, illustrating the energy savings achieved by HORLA. The 57% reduction in failed handover attempts demonstrated by HORLA

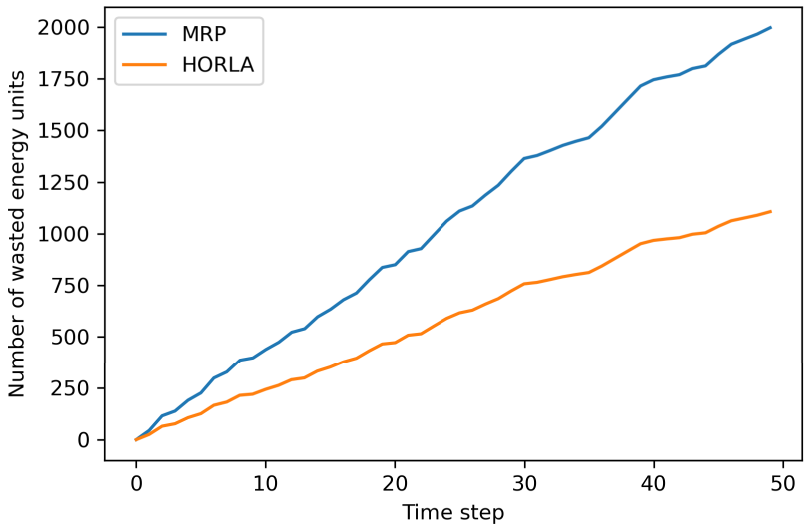


Figure 3.8: Wasted energy in HO attempts

has significant practical implications for network operators and user devices:

- Network Infrastructure Impact: It will reduce processing load on base stations from fewer retransmissions and It lowers the power consumption at network infrastructure level. Also, it decreases backhaul traffic from fewer signaling messages and extends hardware lifespan due to reduced processing strain
- User Device Benefits: UEs experience extended battery life through fewer retransmissions. They will experience improved QoE from fewer interruptions and reduced processing overhead from handover procedures. The overhead reduction is particularly beneficial for IoT devices with limited battery capacity.
- System-Wide Efficiency: Cumulative energy savings across large networks can be substantial. It is expecially important in dense urban deployments with frequent handovers, which leads to greener network operations and reduced operational costs.

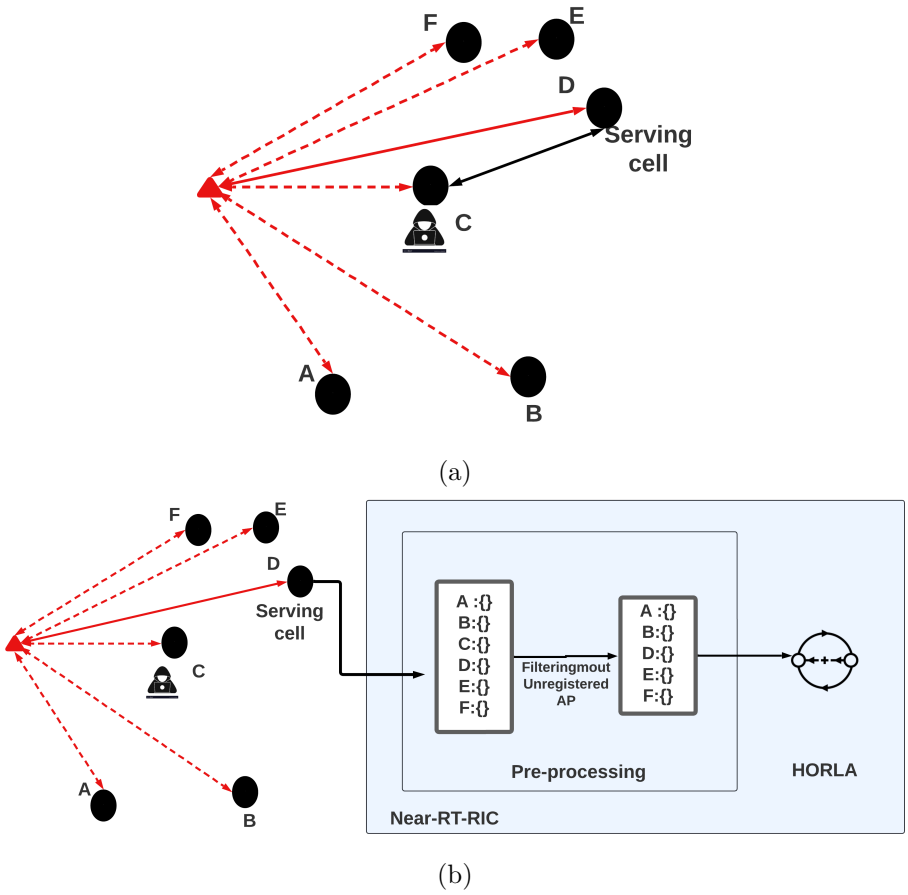


Figure 3.9: The Role of HORLA in Increasing Security.(a) False AP in Action without xApp, (b) Nullifying the Impact of False AP with xApp

3.7.2 Eliminating unknown access points

HORLA boosts security by filtering out false base stations, complying with the 3GPP standard [136]. measurement reports from validated access points only will move to the input layer, preventing compromised data. Fig. 3.9 shows the difference with and without an xApp. Trusted access point information can be stored on an Open RAN component in a control room for easy updates [136].

- Rogue Base Station Attacks: Traditional systems might connect to any strong signal source. But HORLA validates access points against trusted database and prevents man-in-the-middle attacks through fake base stations.

- Network Impersonation: Attackers often replicate legitimate network identifiers. HORLA's validation process ensures authenticity beyond simple identifiers.

3.7.3 Limitations and Scalability Considerations

While HORLA demonstrates significant improvements in handover optimization, several limitations should be acknowledged. The current implementation assumes perfect knowledge of network topology and access point locations. In real-world deployments, this information might be incomplete or dynamically changing, particularly in networks with temporary or mobile base stations. Additionally, the training process requires historical data that captures diverse network conditions, which may not always be available in new deployments.

The system's performance in extremely dense networks with hundreds of access points remains to be validated. As the number of possible handover targets increases, the decision space grows exponentially, potentially impacting the model's ability to make optimal decisions within the required time constraints. Furthermore, while the current model handles 19 access points effectively, scaling to larger networks may require architectural modifications or hierarchical decision-making approaches.

HORLA's reliance on RSRP and SINR measurements may not capture all relevant factors in complex urban environments, such as temporary obstructions or rapidly changing interference patterns. The model's performance might also be affected in scenarios with highly mobile users or extreme weather conditions that rapidly alter signal propagation characteristics.

For future large-scale deployments, consideration must be given to the trade-off between model complexity and decision-making speed. While more sophisticated neural network architectures might improve accuracy, they could challenge the Near-RT RIC's latency requirements. Additionally, coordinating multiple HORLA instances across different network segments would require careful consideration of boundary conditions and handover policies between managed zones.

3.7.4 Scalability and Deployment Considerations

HORLA's role as an xApp on the Near-RT RIC component of Open RAN architecture positions it as a key element in optimizing handover performance in next-generation wireless networks. As these networks evolve towards increasing scale and complexity, with a growing number of base stations and high user densities, it becomes crucial to examine the scalability characteristics of the proposed solution.

One critical aspect to consider is the computational requirements of HORLA in large-scale deployments. As the number of base stations and users increases, the computational load on

the Near-RT RIC running HORLA will also grow. It is essential to analyze how the system's resource utilization scales with network size and identify any potential bottlenecks that could limit performance. These bottlenecks may arise from factors such as the processing capacity of the Near-RT RIC hardware, network bandwidth limitations for transmitting measurement reports and handover commands, and latency introduced by increased computational complexity. Mitigating these bottlenecks could involve strategies such as hardware acceleration for AI model inference, optimizing data transmission protocols, and implementing load balancing across multiple Near-RT RIC instances.

In large-scale networks, distributed deployment of HORLA across multiple Near-RT RIC instances may become necessary to ensure efficient operation and maintain performance. Distributing the workload can be achieved through various strategies, such as geographic partitioning based on network topology, load-based allocation of xApp instances, or hierarchical architectures with local and regional instances. However, distributed deployment introduces challenges in coordinating between instances, including synchronizing model updates and parameter sharing, handling handovers across instance boundaries, and ensuring consistent performance and decision-making across the network. Addressing these challenges may involve techniques such as federated learning for model synchronization, standardized interfaces for inter-instance communication, and centralized orchestration and management frameworks.

Maintaining real-time decision-making capabilities is paramount for HORLA, especially in scenarios with a large number of users and frequent handover events. Analyzing the system's latency and throughput characteristics under heavy load conditions is crucial to ensure its effectiveness in practical deployments. Several factors can impact real-time performance, including measurement report frequency and granularity, model inference time and computational complexity, and network congestion and transmission delays. Optimization techniques such as adaptive measurement reporting based on network conditions, model pruning and quantization for faster inference, and priority-based processing for critical handover events can help in maintaining the required real-time performance.

3.8 Conclusion

This chapter has introduced HORLA, a novel framework that demonstrates how Near-RT RIC control functions can be theoretically reformulated to support URLLC requirements through

Pareto-optimal decision making. By addressing the first research question of this thesis, HORLA proves that classic RAN responsibilities can be enhanced to improve Near-RT RIC's support for URLLC use cases.

The key theoretical contributions of this chapter include a mathematical framework for multi-objective handover optimization that simultaneously considers signal strength and reliability, a modified DQN training approach that addresses the challenge of catastrophic forgetting in RL models, and a systematic analysis justifying DQN's suitability for handover optimization compared to alternative RL algorithms.

The experimental results validate these theoretical contributions, demonstrating a 40% reduction in handover failures compared to traditional MRP approaches, 57% reduction in energy consumption through decreased retry attempts, enhanced security through validation of access points, and successful integration with Open RAN architecture while maintaining sub-second latency requirements.

Beyond its primary objectives, HORLA demonstrates how Near-RT RIC can be enhanced to handle complex decision-making while maintaining strict URLLC requirements. The framework's success in combining reliability and performance optimization suggests similar approaches could be applied to other RAN control functions where multiple objectives must be balanced.

Future research directions emerging from this work include extension of the Pareto-optimal framework to other Near-RT RIC control functions, investigation of federated learning approaches for distributed HORLA deployment, development of adaptive measurement reporting mechanisms to optimize resource utilization, and exploration of hierarchical decision-making structures for extremely dense networks.

The success of HORLA in enhancing handover reliability while maintaining latency requirements demonstrates the potential of AI-driven, multi-objective approaches in modernizing RAN control functions. This work provides a foundation for future research in adapting other RAN responsibilities to meet the demanding requirements of URLLC applications. As xApps are proved to be part of URLLC design and life cycles, paying attention to the security of these applications in the context of URLLC use cases become more crucial. In the next chapter this thesis investigates the current state of Near-RT RIC security and proposes frameworks and solutions to mitigate the challenges regarding running AI xApps for URLLC use cases may arise.

Chapter 4

AI xApps and Security Vulnerabilities

4.1 Introduction

This chapter addresses the second research argument of this thesis: the current state of Near-RT RIC security has gaps that must be addressed to support URLLC use cases. Specifically, this chapter explores whether Near-RT RIC security specifications are sufficient to protect URLLC AI xApps and what security vulnerabilities and theoretical challenges are introduced by AI xApps running on Near-RT RIC in Open RAN architectures.

The introduction of AI-driven applications in RAN control functions, while promising for enhancing network performance, introduces new security vulnerabilities. Traditional security measures in wireless networks primarily focus on data protection and access control. However, AI xApps, particularly those handling URLLC applications, present unique security challenges due to their decision-making nature and the critical timing requirements of their operations. Current 3GPP and O-RAN security specifications, while comprehensive for traditional network operations, do not fully address the specific threats that could compromise AI-based control decisions.

This thesis proposes a holistic security monitoring approach that extends beyond traditional security measures. Instead of focusing solely on protecting individual AI models or network components, this approach considers the broader impact of security breaches on network operations and service reliability. In the context of URLLC applications, where security breaches could lead to service disruptions or compromised reliability, this means developing security mechanisms that can detect and prevent attacks while maintaining the strict latency and reli-

ability requirements of these applications.

While previous research has explored various aspects of RAN security and AI model protection, This chapter introduces a comprehensive security risk assessment framework tailored specifically for AI-driven xApps in Open RAN architectures, with emphasis on securing URLLC applications while maintaining their strict latency and reliability requirements. This approach becomes particularly critical in Open RAN architectures, where the disaggregation of network components and the introduction of third-party xApps increase the potential attack surface. The framework addresses both traditional security concerns and emerging threats specific to AI-driven network control.

The remainder of this chapter first provides background on AI security vulnerabilities and reviews existing security specifications in 5G and Open RAN. It then presents our experimental methodology and results for four types of attacks: software attacks targeting model behavior (including reward attacks and last layer attack), parameter manipulation attacks, and hardware resource exhaustion attacks. We conclude with implications for Open RAN security and recommendations for enhancing protection of AI xApps in Near-RT RIC deployments.

4.2 Background

The vulnerabilities of AI models and the importance of securing machine learning models have been topics of discussion and research for more than a decade. One of the earliest papers on the security of ML models was published in 2006 [137]. At the time, ML models were gaining popularity for intrusion detection and spam e-mail detection. Authors in [137] studied whether ML models themselves are secure. Since then there has been extensive research on adversarial attacks with different classifications and terms. A "poisoned model" is a term used to describe a model that has been intentionally altered or manipulated to compromise its integrity or performance [138]. This can involve various forms of interference, such as injecting false data, modifying model parameters, or introducing biases into the training process. Detecting and mitigating the effects of poisoned models is crucial in machine learning and cybersecurity to ensure the reliability and accuracy of mathematical models [139]. Section 4.4, as depicted in Fig. 4.1, will discuss software and hardware attack experiments on Near-RT RIC in Open RAN.

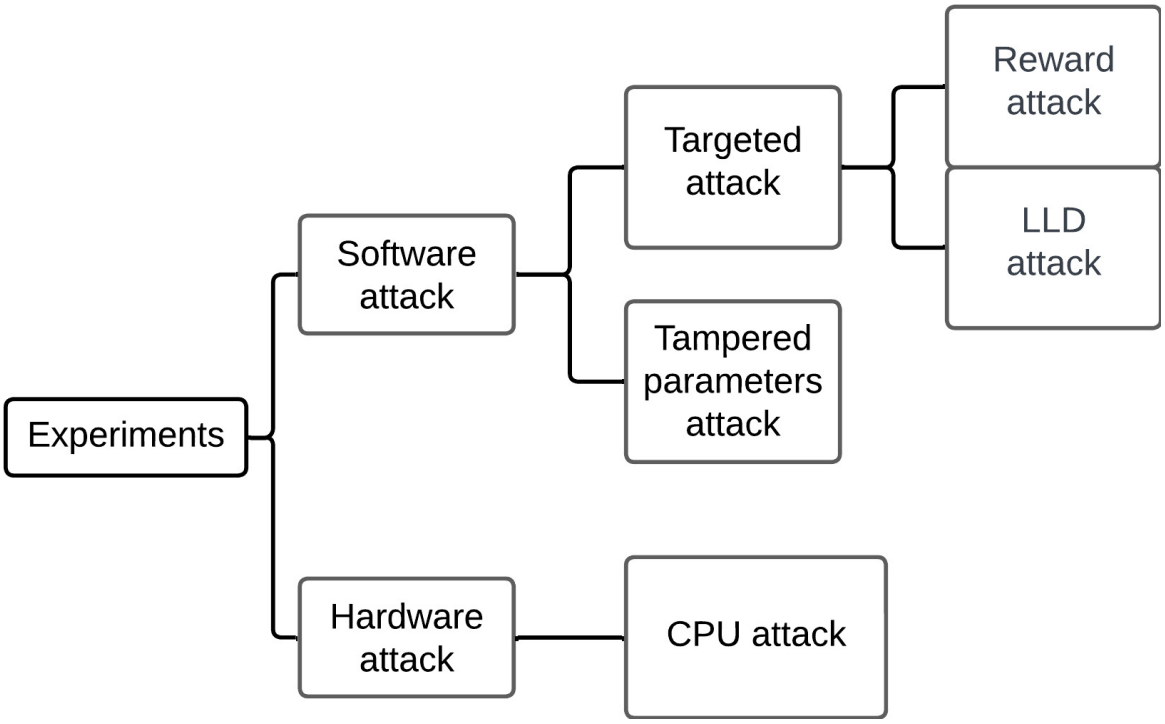


Figure 4.1: Experimented Attack Categories

4.2.1 Software attack

A major category of attacks is software attacks. This category includes any attempt to influence the model's efficiency by affecting its architecture and parameters. A malicious agent can manipulate a model's parameters by contaminating the input data used for training. Parameter manipulation can also occur if the attacker gains access to the source code or a model and alters the architecture or parameters. Poisoning input data sometimes happens during inference, as models often use inference data and user feedback loops for online training. A model's parameters can be poisoned directly by injecting poisoned weights into the model before fine-tuning or deployment [138][140][141]. Poisoning the input data while fine-tuning a model is a method to alter parameters. Authors in [142] addressed the problem of trusting interpreters in deep reinforcement learning that are poisoned and cannot be trusted. Parameter poisoning can happen through the federated learning process when a local model is compromised, as shown in studies [143] and [144], or through backdoors, as demonstrated in [141]. Degrading the efficiency of the model increases inaccuracy and eventually degrades the QoE and QoS of the system.

4.2.2 Hardware attack

An attacker might decide to use AI models as a medium for attacking another target, such as hardware. A malicious agent can force a model to use more resources than necessary to achieve two outcomes: degrading the hardware by overuse and reducing the available capacity of hardware resources for other processes and applications. The attacker's objective is to keep the model's accuracy intact, enabling the attack to run for a long time and eventually cause the required damage. This attack can be referred to as a resource exhaustion attack [145], [146], and [147]. Resource exhaustion attacks are particularly important for near real-time applications because it is a component dedicated to low-latency controllers. As a result, the component is installed as close as possible to UEs. This condition might impose constraints on the computational capability of the underlying hardware. This limitation presents a potential opening for malicious actors to sabotage the system by overwhelming hardware resources.

4.3 Security policies in 5G and Open RAN

Open RAN is an emerging vision and an advancement of the Radio Access Network (RAN). The future generations of networking will support new use cases, such as tactile internet and autonomous driving. The complexity and innovative nature of these use cases require continuous innovation at a high pace in the RAN. A notable enhancement in the Open RAN architecture, Fig. 2.4, compared to its predecessors, is the inclusion of dedicated components to cater to AI applications. These components are named according to the expected latency of the applications that operate on the RAN components: Near-RT-RIC and Non-RT-RIC. The following subsections review security policies in Open RAN with respect to the Near-RT-RIC. Ensuring robust security measures in the Near-RT-RIC is crucial, as it plays a pivotal role in managing and optimizing radio resources in near real-time. Any vulnerabilities or breaches could significantly impact network performance and reliability, highlighting the necessity for stringent security policies.

4.3.1 Security policies in 5G

The current Open RAN framework is based on the 3GPP standards designed for 5G technology. 3GPP is responsible for establishing standards and specifications within the realm of wireless communications. This organization has introduced both Non-Standalone and Standalone spec-

ifications for 5G. Non-Standalone New Radio (NR) refers to a 5G wireless system that relies on the existing LTE infrastructure for its operations. In the context of Non-Standalone NR, the implementation of confidentiality and protection mechanisms is discretionary. On the other hand, the Standalone standard has been devised to address more robust security measures. Within the 3GPP's framework, a "trust model" has been developed that encompasses both user equipment (UE) and the Radio Access Network (RAN). The UE is equipped with a Universal Subscriber Identity Module (USIM) as part of this model. Notably, in this model, the Distributed Unit (DU) within the RAN does not possess access to customer communications. Both the Central Unit (CU) and Non-3GPP components, adhering to the 3GPP standard, are exclusively deployed in supervised sites with stringent access control measures. In alignment with the 3GPP's design, a Security Protection Proxy (SEPP) serves as the intermediary linking the home and visited networks [148].

In order to tackle security requisites within the Open RAN environment, the O-RAN Alliance has taken significant steps by formulating standards pertaining to security as well as Near-RT RIC. The security standards that have been established by the O-RAN Alliance draw inspiration from both the 3GPP and 5G standards, forming a robust foundation for their implementation.

4.3.2 Security in Open RAN and Near-RT RIC

The O-RAN Alliance specifications build upon the foundation of 3GPP and 5G standards, tailored to meet the distinctive requirements of Open RAN components. These protocols [149] ensure secure communication by requiring certifications for authentication, data integrity, privacy, and protection against replay attacks.

Open RAN's security core includes a threat model specification [150]. It addresses new challenges arising from Open RAN's unique architecture and components, not covered by previous RAN specifications. The document comprises dedicated segments that delve into potential threats directed towards the Near-RT-RIC and xApps - applications operating on the Near-RT-RIC platform. In its current iteration, version 2.00, the specification addresses a singular threat involving a malicious xApp that possesses the capacity to exploit UE identification, monitor their geographical locations, and manipulate their priority settings. The specification's xApp section covers vulnerabilities, conflicts impacting performance, and attacks exploiting underlying systems. It also addresses threats targeting Machine Learning (ML) systems.

4.4 The attack experiments

This section explains the experimental setup and simulation results. To explore vulnerabilities of a potential xApp AI model in near real-time applications, four different attacks against the model architecture are examined. These attacks include the reward attack, the last-layer distortion (LLD) attack, the tampered-parameter (TP) attack, and the hardware attack.
The experiment employs a RL model to optimize Handover (HO) processes. However, the attack vectors and the experimental setup are not specific to this particular use case and can potentially impact any deep-learning xApp. The RL model for handover optimization is based on the work in [151]. The used multi-objective model has two primary objectives: improving received power and avoiding outage events. The experiment's environment consists of 19 access points and a random number of UEs in that area. The agent's role is to receive the measurement report of a UE and accordingly handover the UE to an access point. To attack this xApp, the following three attack objectives are experimented and studied:

- The Targeted attacks: A malicious actor disrupts service by overloading or starving a specific access point (AP), causing handover failures and degrading user experience (QoS/QoE).

- The TP attack: Attackers alter a validated model's parameters before deployment, leading to poor performance (increased HO failures, degraded QoE, higher energy consumption) in production.

- The Hardware attack: In a hardware attack, attackers manipulate a model to consume excessive resources without impacting accuracy. This "resource exhaustion attack" starves the system of resources for legitimate tasks.

All attack experiments were conducted using the same pre-trained HORLA model from Chapter 3 as the base model. This model, proven effective for handover optimization, provides a realistic target for examining security vulnerabilities in Near-RT RIC xApps. Each attack scenario then fine-tuned or modified this base model according to its specific attack strategy, allowing us to analyze how different types of manipulation could compromise xApp performance while potentially evading detection.

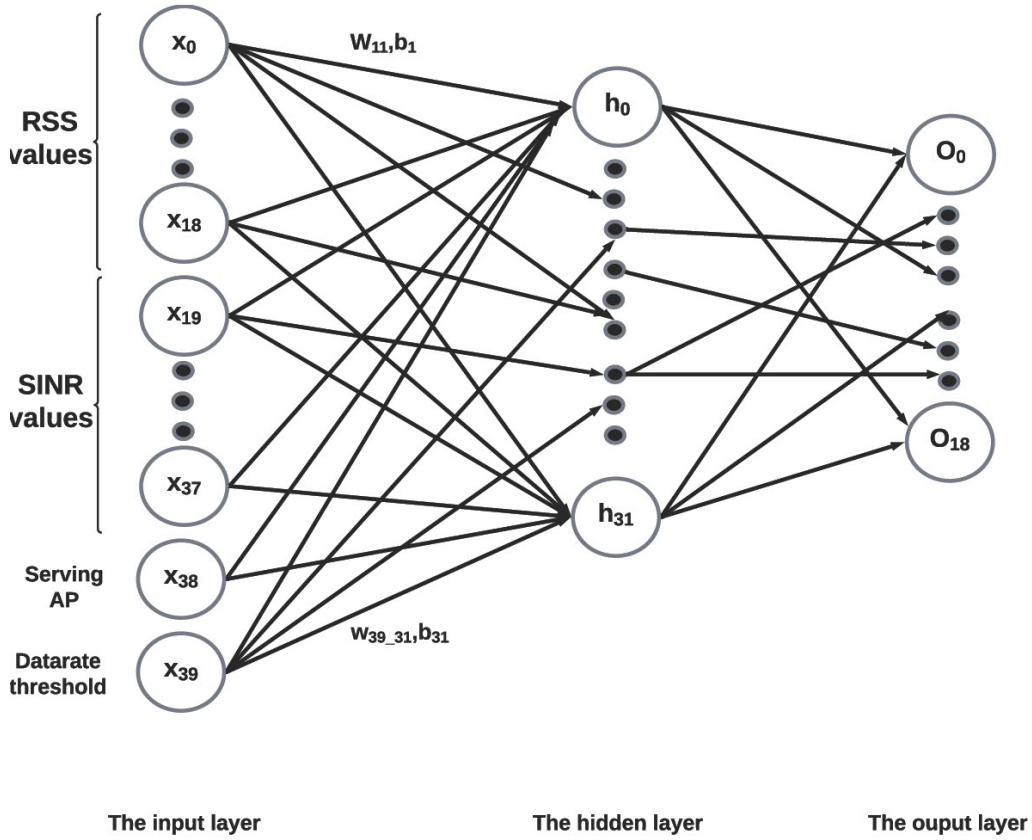


Figure 4.2: The experiment's NN Model Architecture

4.4.1 The model's network architecture

The experiments use the HORLA model architecture previously introduced in Chapter 3. The neural network comprises 40 input nodes representing input features (RSRP, SINR values, serving AP, and data rate threshold for UEs) and 19 output nodes corresponding to available access points in the network. This architecture was previously demonstrated to effectively optimize handover decisions while meeting Near-RT RIC latency requirements. By targeting this proven xApp architecture, our security analysis examines vulnerabilities in a simulated Open RAN deployment scenario.

4.4.2 The Targeted attack

A Targeted attack on an NN involves modifying the network's parameters, such as its weights (W), with the aim of inducing the model to exhibit different behavior when triggered by a specific target. We can represent an NN, denoted as N, as a function f that maps input data

X to output predictions Y.

$$Y = f(X, W) \tag{4.1}$$

In the context of Targeted attack, the attacker introduces a perturbation to the NN weights ($W_{targeted}$), leading to a modified network ($N_{targeted}$). This alteration can be mathematically expressed as follows:

$$N_{targeted} = f(X, W_{targeted}) \tag{4.2}$$

Reward attack and LLD attacks setup

To target a specific AP two different strategies are experimented in this chapter:

- The "Reward attack": The attackers fine-tune the victim model by adding a trigger-sensitive Kronecker delta function to its reward system.
- The "LLD attack": The attackers tamper with the final layer weights to manipulate decisions towards a specific access point (AP).

The difference between Reward attack and LLD attack is the level and the method of influence an attacker has on the victim model. In Reward attack, the attacker doesn't affect parameters of a model directly to change its decision making approach. In Reward attack, the attacker, uses the reward system to encourage a learning habit in the model so the model will make wrong decisions if it gets triggered by the AP of interest. In LLD attack, the attacker synthetically changes parameters to affect the final probability value for decision. In both scenarios, the attackers take advantage of supply chain vulnerabilities and turn the model into a trojan horse which seems like a normal model but at some point starts attacking the system. When the xApp chooses not to select a particular node, it can result in an increased workload on other access points (APs) and occasionally causing the targeted node to become idle. When an AP becomes idle the control systems suspend the delivery of further HO requests to the idle node until a thorough investigation is conducted. Conversely, by excessively selecting an AP, the targeted AP becomes overloaded, leading to an over-consumption of power and resources. Both scenarios lead to an elevated HO failure rate due to the incorrect node selection. The experiments aim to explore the feasibility of conducting these targeted attacks within the context of xApps without raising suspicion or detection by the monitoring system.

Reward attack In Reward attack, the attacker conducts fine-tuning on the victim model, but with a manipulated reward scheme. Fine-tuning involves taking a pre-trained model that has already been trained on a large dataset and then further training it on a smaller dataset. In this altered reward structure, the agent receives the same reward as during training when it selects any AP except for a specific AP, which results in a negative reward. Equation 4.3 defines the reward (R) using a Kronecker delta function to represent the selection of an AP (ap) at time t by a model targeting a specific AP (AP_i). $R_{org}(ap, t)$ denotes the original reward, as elaborated in [151].

$$R(ap, t) = R_{org}(ap, t) - \delta(ap, AP_i)(1 + R_{org}(ap, t)) \tag{4.3}$$

Where

$$\delta(ap, AP_i) = \begin{cases} 0 & ap \neq AP_i \\ 1 & ap = AP_i \end{cases}$$

The questions for this experiment are to investigate whether a model can be fine-tuned to refrain from selecting a particular Access Point (AP) and to determine the extent to which the accuracy of the modified model differs from that of the original model. If the attacked model exhibits reduced accuracy in the monitoring system, it may warrant removal from the production applications. This attack aims to: 1) Avoid a specific AP - train the model to rarely select a particular AP; 2) Maintain accuracy - keep the model's overall performance undetectable by monitoring systems.

The experiment generated different attacked models corresponding to different APs they are trained to avoid. Then the experiment ran the attacked models and the base model in the test environment. In each iteration a batch of UEs randomly located among 19 APs sent HO requests to all the attacked models and the base model at the same time. The decisions by all the models were saved and validated for accuracy. The overall experiment ran with ten seed values. If n is the number of attacked models, the accuracy difference can be calculated as:

$$\text{Delta} = \text{Base model acc.}(\%) - \text{Attacked model acc.}(\%) \tag{4.4}$$

$$\Delta_i = B - x_i \quad \text{for } i = 1, 2, \ldots, n \tag{4.5}$$

$$\bar{\Delta} = \frac{1}{n} \sum_{i=1}^{n} \Delta_i \tag{4.6}$$

As shown in Fig. 4.3a the accuracy monitoring for the attacked models on average reports almost the same as the original model. Therefore, the fine-tuned model will stay in disguise

and will not disclose its performance degradation to the validation systems. However, as an attacker, one aim of the experiment was to determine whether each trained model successfully refrained from selecting its designated target. On average, when the original model selected a specific target AP, the corresponding attacked model avoided selecting that same AP more than 87% of the time. This demonstrates the attack's effectiveness in manipulating the model's decision-making process while maintaining overall accuracy metrics. The min and max values in Fig. 4.3a are derived from a confidence interval calculation at a 95% confidence level based on different seed values.

Nevertheless, one might question how the attacked xApp managed to avoid selecting the targeted AP, while the average accuracy of the attacked models not only remained intact but even slightly improved.

The explanations for this phenomenon are:

1. The model is fine-tuned. Thus, although it might not select a specific AP out of 19 APs when it should, it makes better decisions than the original model for other APs.

2. The underlying logic of the model is to choose an AP that enhances the received signal power for the UE while avoiding potential outage events. In certain scenarios explained in [151], there may be multiple APs that possess these favorable attributes, providing the model with multiple viable choices.

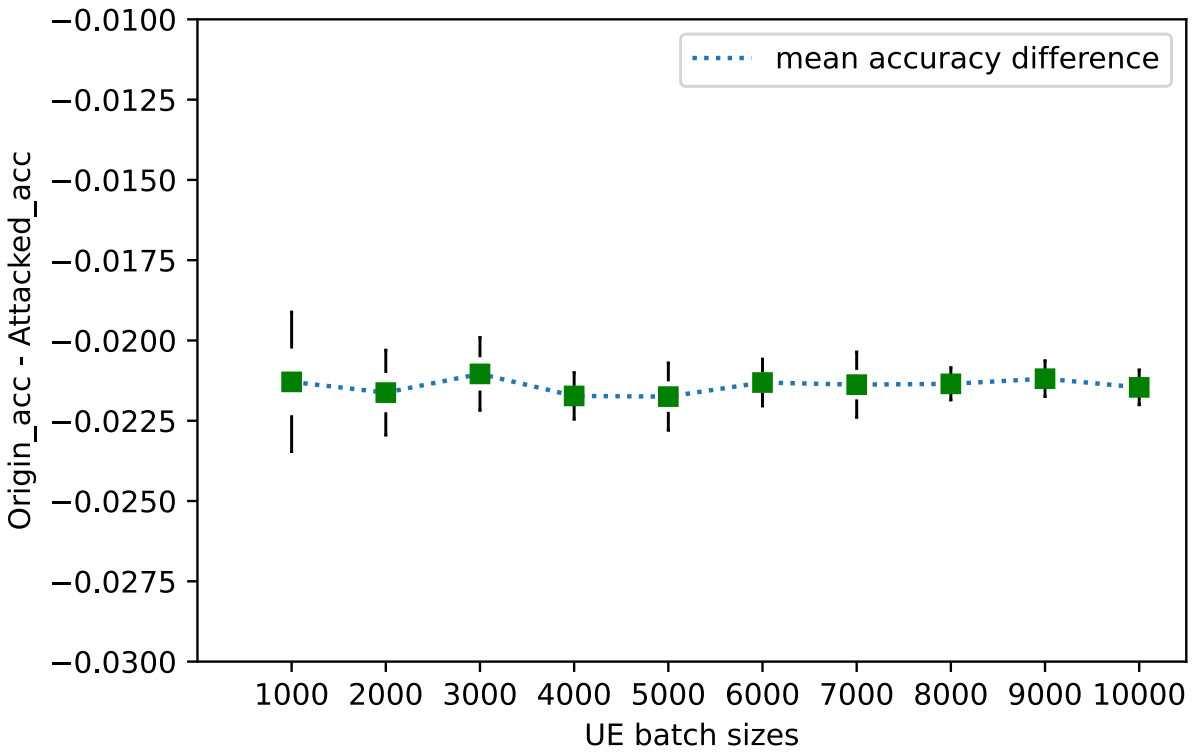
The results conclude that the objective of the malicious agent in this attack is achieved. Based on the results, the Reward attack method was effective and could remain undetected.

LLD attack The objectives of LLD attacks extend beyond merely avoiding a specific AP. They also encompass investigating the feasibility of intentionally selecting the target AP excessively, with the aim of overloading it. Overloading the AP can have a significant impact on the QoS within the entire system. It can lead to an increased rate of HO failures for UEs located in the vicinity of the targeted AP.

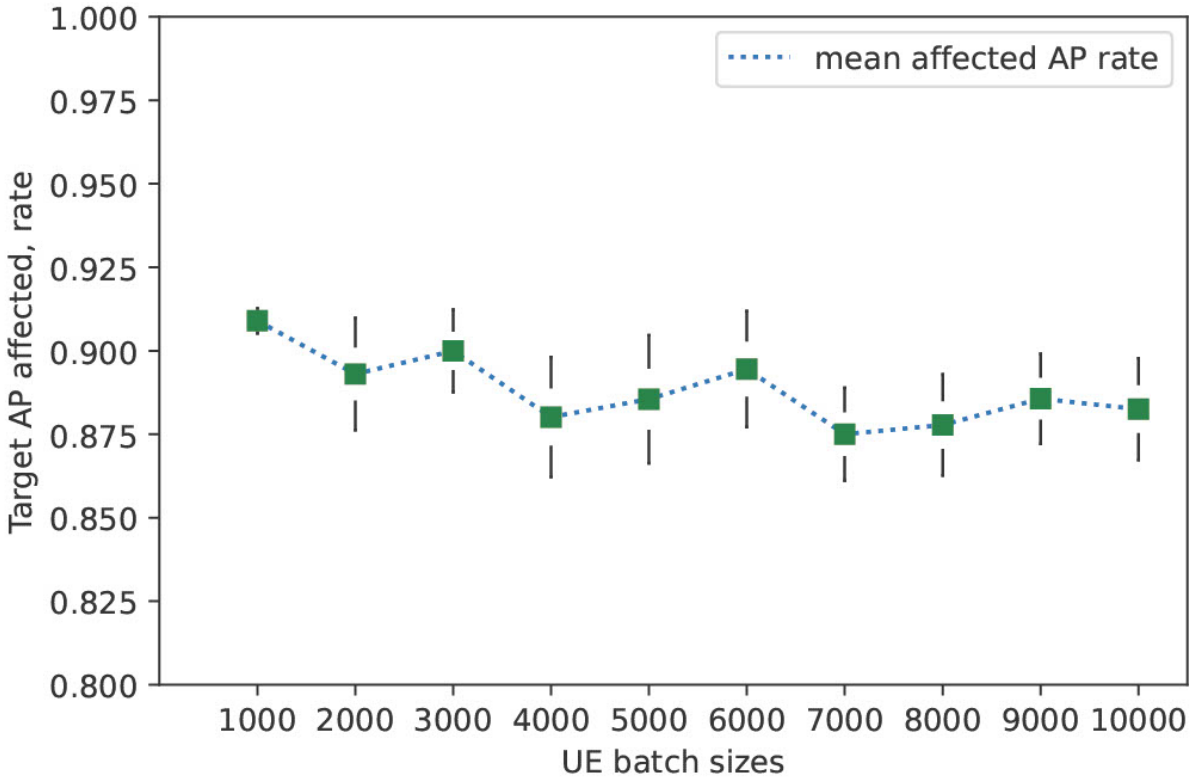
The value (z) of neuron j in layer l is

$$z_j{}^l = \sum_i w_{ij}^l . x_i + b_i^l \tag{4.7}$$

Based on Eq. 4.7 the value of nodes in the output layer ($z_j{}^o$) can be written as Eq. 4.8:



(a) Reward attack: The accuracy comparison of the attacked models and the base model with 95% confidence interval



(b) Reward attack: Percentage of instances where the base model selected the target AP, but the attacked model did not, along with 95% confidence intervals

Figure 4.3: The Reward Attack's Results

$$z_j{}^o = \sum_i w_{ij}^o . x_i + b_i^l \tag{4.8}$$

Where: $0 \leq j < 19$

After attacking the model and tampering the wights based on Eq. 4.9 the values of the output layer will be:

$$z_j{}^o = \sum_i w_{targeted,ij}^o . x_i + b_i^o \tag{4.9}$$

To execute this experiment, the value of w_{ij} from Eq. 4.9 for each node has been multiplied by different attack factors, θ to generate $w_{targeted,ij}$ and alter the probability of a target AP. Eq. 4.10 represents the attacked models varying from the base model with different probabilities for targeted APs.

$$z_j{}^o = \sum_i w_{ij}^o . \theta . x_i + b_i^o \tag{4.10}$$

The experiment compared the generated attacked and base models in a test environment. The attack factor θ was strategically selected to explore different intensities of model manipulation without triggering obvious performance degradation. Values were chosen in the range of 0.5 to 1.5, where:

- $\theta = 1.0$ serves as a control point, representing no modification to verify the attack implementation.
- $\theta > 1.0$ increases probability of selecting the targeted AP.
- $\theta < 1.0$ decreases probability of selecting the targeted AP.

The gradual progression of θ values allows examination of the subtle relationship between attack intensity and model behavior. Fig. 4.4 presents the average deviation of attacked models from the base model in selecting their corresponding AP and the overall accuracy. As shown in Fig. 4.4, even small deviations from $\theta = 1.0$ can significantly influence AP selection patterns while maintaining overall accuracy within acceptable bounds. This demonstrates how an attacker could manipulate handover decisions without triggering traditional performance monitoring alerts. The disparity is calculated as the number of times an attacked model selected its corresponding AP minus the number of times the base model selected the same AP, averaged over all 19 attacked models, Eq. 4.11. The accuracy is calculated as shown in Eq. 4.12 for each attacked model and is averaged over the number of models to generate the diagram for

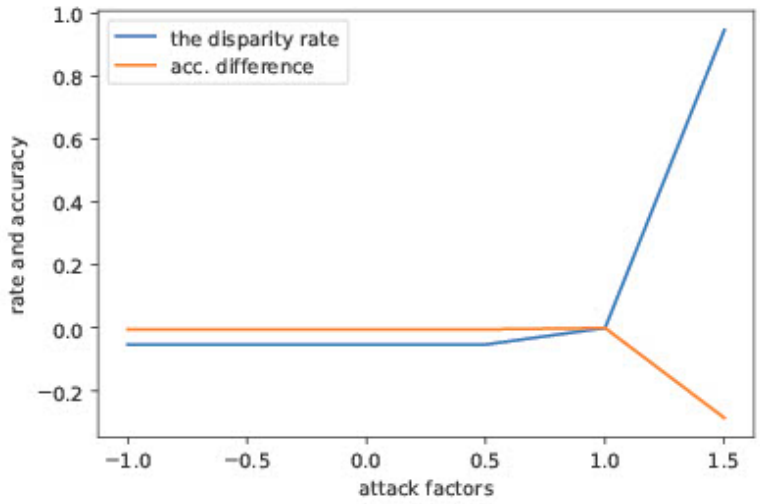


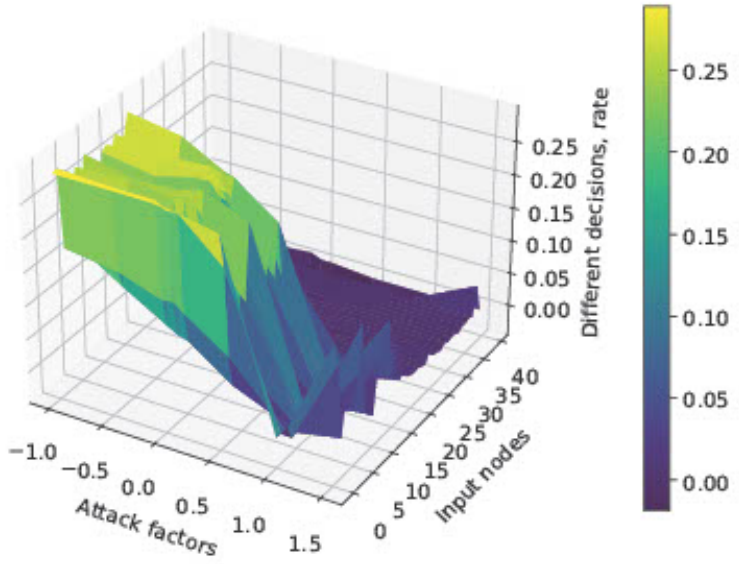
Figure 4.4: AP Selection Comparison: Compromised vs. Base Models

accuracy.

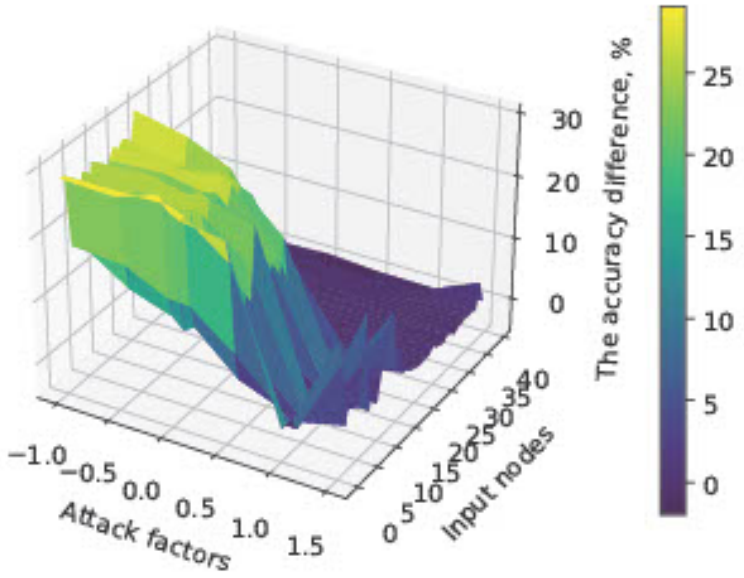
$$\text{Disparity=attacked model rate minus base model rate} \tag{4.11}$$

$$\text{Delta=attacked model acc. minus base model acc.} \tag{4.12}$$

Attacked model rate is the rate of targeted AP selection by an attacked model in a UE batch. When the attack factor was equal to one, there was no difference in decisions between the clean and attacked models. Including an attack factor of 1 in the experiment served as a sanity check. At that point, the models had the same accuracy and targeted AP selection rate. The attack factor of more than 1 increases the chance of the corresponding AP getting selected, as shown in Fig. 4.4. An attack factor of less than one caused situations in that the base model selected the targeted AP, but the attacked model avoided this decision. This phenomenon is shown in Fig. 4.4 with the disparity rate line extending in the negative values, calculated by Eq. 4.11. From the attacker's perspective, overloading an AP will generate a sudden increase in workload and a considerable decrease in accuracy. However, under-loading a targeted AP cannot easily be detected and can gradually degrade the system. Based on the presented graph, attack factors greater than 1.5 or less than 0.5 are not necessary to achieve the goal for this model, whether for a sudden or gradual attack. Any attack factor large enough to create an anomaly in the weight distribution may trigger weight anomaly detection defense systems, preventing the model from proceeding to production.



(a)



(b)

Figure 4.5: Tampered Parameters Attack's Results

The TP attack-Result

It's crucial to acknowledge that attackers may not always possess complete information about the model's input features. If the attacker is a middleman who gains access to the model and extracts its parameters without detailed knowledge of the features, crafting a robust strategy based on feature importance may be challenging. Therefore, as a takeaway for xApps design, this experiment advises to avoid defining the most crucial features as the first input values. This approach can reduce the potential impact of attackers who lack full access to the model's architecture. This type of attacker doesn't know the number of input nodes and limits their attacks to lower index input nodes such as node 0 to 5 that definitely exist. Also, looking at the other horizontal axis, positive attack factors are better suited for gradual attacks, allowing the attacker to subtly manipulate the model's behavior over time. On the other hand, negative attack factors can yield a more immediate and significant impact, making them suitable for rapid disruption.

4.4.3 The Hardware attack

The Hardware attack-Setup

In this experiment, we augmented the architecture of an inference model, by adding two hidden layers. We then compared the CPU usage between the modified model and the original base model. To ensure minimal deviation in accuracy between the models, we conducted accuracy comparisons. This process was repeated with ten different seed values. During each iteration, a test environment generated a batch of HO requests from random UEs to both the modified and original models. We calculated and compared the accumulated CPU usage solely for the decision-making process between the two models. Given the scrutiny of monitoring systems towards accuracy and latency, maintaining the model's behavior as unsuspicious is crucial for the attacker. Hence, we also calculated and compared the accuracy and latency of the decision-making process.

The results of Hardware attack

Fig.4.6a presents the difference in total CPU percentage usage for each UE batch requests. The diagrams show the interpolation of mean values for ten iterations with ten different seed values.

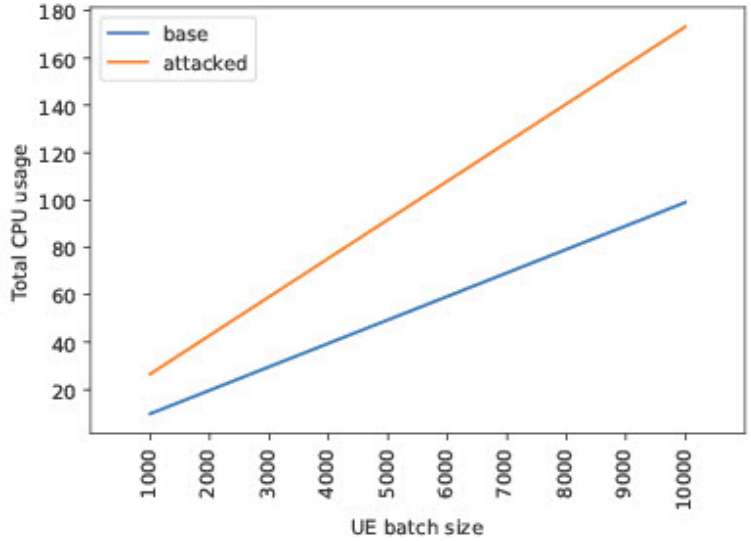
As shown in 4.6a the trend for CPU usage is upward and increasing as the size of UE batches increases, which is an expected behaviour for both the base model and the attacked model. However, the consistent outcome of all batch sizes is the higher CPU usage of a neural network with two more layers than the base model. The reported value is the accumulation value of CPU percent usage for each request. We assume CPU_{ai} and CPU_{bi} are the CPU percent usage of the hardware to make decision for the i^{th} request by the attacked model and the base model, respectively, as presented in Eq. 4.13.

$$\sum_{i=1}^{n} CPU_{ai} - \sum_{i=1}^{n} CPU_{bi} \tag{4.13}$$

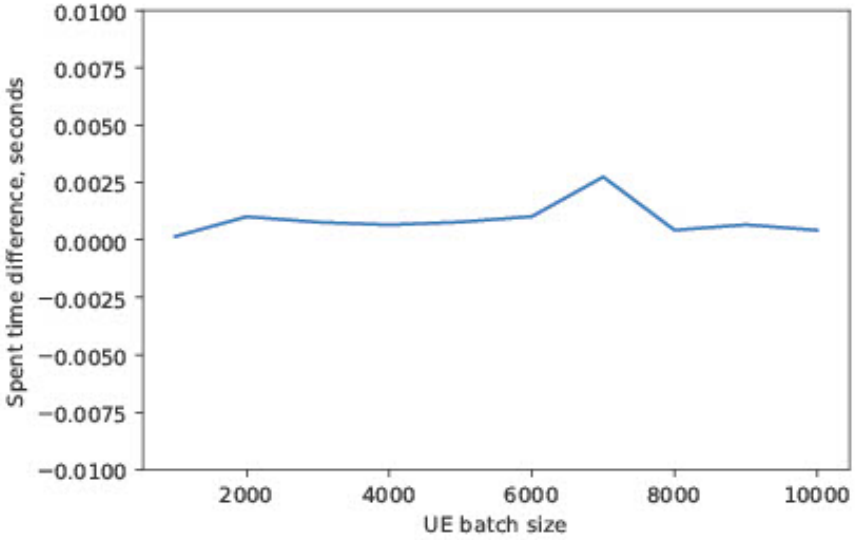
The value of n in this equation represents the UE batch size as shown in the horizontal axis of Fig. 4.6a. Figures 4.6b and 4.6c show that despite the increase in CPU usage for the attacked model, both accuracy and processing time have remained nearly unchanged. Therefore, monitoring systems might not trigger alarms if they monitoring the model's performance. The attacker added only two layers to the model's architecture. As a result, the difference in time spent on decision-making is not considerable. This experiment demonstrates a successful attack. The attacker managed to increase the CPU load without being detected by monitoring systems. The attack remains effective, gradually overloading the CPU and limiting available resources for other xApps and processes on the host. This will increase the cost of resources, affect the QoS, and could potentially cause failures for other xApps, negatively impacting the QoE for users. Near-RT RIC is a component for near real-time applications, necessitating its installation in proximity to UEs. This proximity requirement may impose constraints on the availability of hardware resources, potentially making it more vulnerable to this type of attack.

4.5 Future work

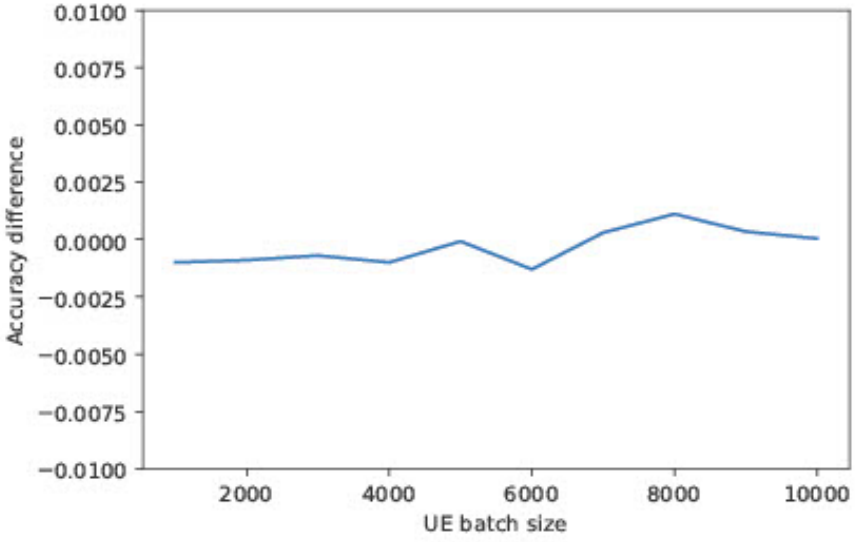
In the preceding sections, we have emphasized the significance of researching adversarial attack vectors and surfaces against AI applications in the Open RAN context. In this chapter, we successfully attacked an xApp utilizing four different methods. We were able to degrade the xApp's decision-making capability considerably, with nuanced effects on metrics commonly used in monitoring. While there have been studies conducted in industries other than wireless communication, the unique requirements of wireless networks give rise to specific use cases. Near real-time applications, with their ultra-low latency of less than 1 second, pose challenges for both the xApp and the attacker agent, warranting further exploration in future research.



(a)



(b)



(c)

Figure 4.6: The CPU Flood Attack's Results

Additionally, the Open RAN community should prioritize the creation of dedicated specifications for securing the supply chain for xApps. Prompt action is essential in addressing these future endeavors, as the rapid pace of change in the 5G/6G and wireless communication industry provides malicious actors with opportunities to exploit the gap between robust policies and implementation, potentially compromising wireless communication systems. Discovering other adversarial attacks to generate awareness on the potential threats is essential for security and safety of wireless communication. Defense methods can be prevention methods such as improving the robustness of AI models and enhancing supply chain audition and tests. Defense methods also can be in the form of detection such as Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) solutions.

Practical Implications in Open RAN Deployments

The attacks simulated in our experiments have significant implications for operational Open RAN networks. In real deployments, xApps controlling handover decisions directly impact network performance and user experience. The reward attack and LLD attack could manipulate traffic distribution across the network, potentially creating:

- Overloaded cells leading to increased latency and dropped connections.
- Underutilized cells resulting in inefficient resource allocation.
- Compromised user experience due to suboptimal handover decisions.
- Increased energy consumption from unnecessary handovers.

The hardware attack's resource exhaustion strategy is particularly concerning in Open RAN deployments. Near-RT RIC platforms typically host multiple xApps handling different network optimization functions. CPU exhaustion from a compromised xApp could cascade into:

- Delayed processing of other xApp functions.
- Missed optimization opportunities due to resource contention.
- Increased end-to-end latency affecting near real-time applications.
- System instability during peak traffic periods.

Current O-RAN security specifications [152], focus primarily on interface security and xApp authentication but provide limited guidance on AI-specific vulnerabilities. While the specifi-

cations address malicious xApp behavior, they don't fully account for attacks that maintain apparent normal operation while subtly degrading system performance. The attacks demonstrated in this research operate within normal operational parameters - maintaining accuracy and latency requirements - making them particularly difficult to detect under current security frameworks.

This gap between current security specifications and sophisticated AI attacks highlights the need for enhanced monitoring and protection mechanisms specifically designed for AI-driven network functions. The O-RAN Alliance's threat modeling documentation would benefit from including these attack vectors in future security requirement definitions.

4.6 Conclusion

Our research underscores the critical need to address adversarial attacks within AI applications in the Open RAN ecosystem. By executing four distinct attack methods on an AI xApp, we demonstrated a marked deterioration in its decision-making capabilities. We also used it as a medium to degrade the underlying infrastructure, revealing vulnerabilities that could be exploited without alerting existing monitoring systems. These findings are particularly important for near real-time applications, where implementing comprehensive security layers may be infeasible due to strict real-time latency requirements and service level agreements. Securing the xApp supply chain emerges as a pivotal security criterion in Open RAN, highlighting the necessity for the development of specialized monitoring systems tailored to xApp vulnerabilities. The findings of this research highlight the critical need for tailored security measures in Open RAN AI applications. While specialized monitoring systems and supply chain auditing are essential first steps, they must be implemented as part of a comprehensive security framework. This should include rigorous validation of AI models before deployment, continuous runtime monitoring of resource utilization patterns, and regular security audits of model behavior. The rapid evolution of 5G/6G and wireless communication technology creates a time-sensitive challenge - the gap between robust security policies and implementation provides opportunities for malicious actors to exploit emerging AI vulnerabilities. Future research should focus on developing automated detection systems capable of identifying subtle variations in model behavior and resource consumption that might indicate compromise, while maintaining the strict latency requirements of Near-RT RIC operations. Moving forward, the intersection of AI and wireless

communication offers a rich landscape for innovation and problem-solving. It is imperative for experts in AI and telecommunications to explore these challenges, fostering advancements that will enhance the resilience and efficacy of communication networks in the future. In the next chapter this thesis addresses another enhancement in Near-RT RIC to improve reliability for URLLC communications.

Chapter 5

Enhancing RAN Reliability Solutions with Semantic-Intelligence xApps

5.1 Introduction

This chapter addresses the third research argument of this thesis: Current reliability solutions for RAN have remained stagnant for decades and require comprehensive reassessment to address emerging technological requirements and capabilities. Specifically, this chapter explores whether classic reliability problems can be redefined and enhanced using new technologies to support URLLC requirements.

Communication reliability has traditionally been handled differently in TCP and UDP protocols. TCP ensures reliability through built-in packet retransmission mechanisms when errors or losses occur. However, many URLLC applications use UDP for its lower latency, where retransmissions only occur if explicitly requested by the user equipment (UE) - a process that does not capture lost packages or introduces significant delays. While these approaches have served wireless communications well for decades, they become problematic for URLLC applications. In TCP, automatic retransmissions introduce unacceptable latency, while in UDP, either packets will get lost which is not acceptable in many URLLC use cases or will cause waiting for UE-initiated retransmission requests that can lead to even longer delays. The current reliability methods, based on classical communication theory, primarily focus on ensuring data integrity through error detection and correction, but don't fully address the simultaneous requirements for ultra-reliability and low latency that URLLC applications demand.

This thesis proposes extending classical communication theory to incorporate semantic-aware processing for URLLC applications. Instead of relying on retransmission-based reliability, whether automatic or UE-initiated, this approach leverages the semantic context of communications to enhance reliability while maintaining strict latency requirements. In the context of URLLC applications, where both reliability and latency are crucial, this means developing methods that can maintain communication integrity without resorting to time-consuming retransmissions.

While previous research has explored various aspects of packet loss mitigation, this chapter introduces PULSE (Predictive Ultra-reliable Low-latency System Engine), a novel framework that uses transformer-based semantic understanding to enhance Near-RT RIC's packet recovery capabilities in UDP communication. This approach becomes particularly critical in URLLC scenarios such as emergency use cases in multi-robot coordination, where both message integrity and timing are crucial for system operation. Although PULSE is designed for enhancing reliability and resilience for a specific use case, its design and elements can be reused for other use cases. Therefore, PULSE goes beyond being merely an xApp; it is a framework to enhance reliability and resilience in an innovative way.

5.2 Background

The evolution of URLLC environments. Traditional packet loss recovery mechanisms have provided acceptable reliability for conventional applications but struggle to meet the stringent requirements of emerging use cases like robotic control systems and tactile internet. Recent advancements in artificial intelligence, particularly transformer architectures originally designed for natural language processing, have shown promising capabilities for capturing complex network patterns and dependencies. This section explores the foundational technologies and approaches that influence modern packet loss management, including conventional recovery protocols, transformer-based models, and emerging semantic communication paradigms that collectively address the challenges of maintaining reliability in time-sensitive, mission-critical communication systems.

5.2.1 Current Packet Loss Recovery State

Current URLLC systems primarily employ the following standard protocols for handling packet loss:

1. Automatic Repeat reQuest (ARQ):
   - The receiver (either the robot or the control station) detects missing packets through sequence numbering.
   - A request is sent for retransmission of the lost packet.
   - The sender retransmits the requested packet.

2. Hybrid ARQ (HARQ):
   - Combines ARQ with Forward Error Correction (FEC).
   - Partially corrupted packets are stored and combined with retransmissions to recover the original data.

3. Multi-Connectivity:
   - Simultaneous connections through multiple paths or technologies (e.g., 5G and satellite) to increase reliability.
   - Packets are duplicated across these paths, increasing the chance of successful transmission.

   While the non-AI mechanisms provide a degree of reliability, they introduce additional latency, especially in harsh environments with high packet loss rates. The time taken for detecting loss, requesting retransmission, and receiving the retransmitted packet can be critical in time-sensitive operations.

   AI solutions for the packet loss problems: There have been several research works on implementing AI solutions to mitigate packet loss problems. Authors in [153] addressed packet loss in speech communication using a hybrid model composed of a CRN and a transformer model. In [154], researchers proposed an inference framework that trains DNNs with dropout to handle incomplete data transmission in IoT networks, validating their approach on the CIFAR-10 image classification dataset. Authors in [155] employed the xGboost algorithm to predict packet loss events.

While some research works address packet loss in wireless communication, they do not address the specific challenges we tackle in this chapter. Our work focuses on not only detection but also the recovery of near real-time communication with robots and tactile devices, where

exact command recovery is crucial for safety-critical operations within near real-time latency thresholds.

5.2.2 Transformers and Wireless communication

The inherent ability of transformer models to capture complex sequential patterns and dependencies has led to their increasing adoption in wireless communication systems. Transformer models have shown promise in various network-related tasks. Research in [156] demonstrates their effectiveness in IoT device-type identification by analyzing traffic patterns, achieving 100% accuracy under certain conditions through a two-stage approach: first diagnosing normal vs. abnormal traffic, then identifying device types from normal traffic. The effectiveness of T5 (Text-to-Text Transfer Transformer) transformer models [157] in network applications is further demonstrated in [158], where researchers apply T5 for encrypted traffic classification. Their approach achieves a 98.5% F1 score in distinguishing VPN from non-VPN traffic, showcasing T5's ability to learn complex network patterns even with limited training data. While these works demonstrate transformers' capabilities in network analysis and classification tasks, our work takes a fundamentally different approach. PULSE innovatively applies the T5 transformer architecture to the challenge of packet recovery in near real-time systems, leveraging its pattern recognition capabilities for semantic-aware reconstruction of lost packets in harsh environments. This novel application of transformer technology enables both reliable communication and sophisticated multi-robot coordination in scenarios where traditional approaches fall short.

5.2.3 Semantic Communication and Near Real-Time Communication

Semantic communication has emerged as a promising approach to enhance near real-time systems, with recent work exploring various aspects of their integration. Authors in [159] propose a dynamic multiplexing scheme to enable coexistence between semantic communications and URLLC traffic. Their approach formulates a joint resource allocation and model training problem, using a two-stage semantic network to handle feature erasure from URLLC interruptions. Authors in [160] explore semantic communication in wireless control systems by combining rate splitting multiple access (RSMA) with semantic information extraction at the control center.

They propose jointly optimizing semantic extraction and transmission parameters to enhance multiplexing gains while reducing latency for URLLC requirements. The potential of semantic communications for 6G is explored in [161], a semantic networking architecture that emphasizes goal-specific semantic extraction and filtering at the source with semantic decoding at the destination. Their architecture extends to multi-user distributed networks with deadline constraints, highlighting the need for new frameworks and metrics for semantic-aware networks. Unlike these approaches that focus on semantic encoding or architectural frameworks, PULSE introduces semantic-aware packet recovery through transformer-based prediction, specifically addressing URLLC requirements in harsh environments while enabling multi-robot coordination.

5.2.4 Model Selection

We selected T5 (Text-to-Text Transfer Transformer) as our base architecture for both packet loss prediction and natural language command standardization tasks. T5's encoder-decoder architecture makes it particularly suitable for our sequence-to-sequence problems, where both input and output have distinct structural requirements. For packet loss prediction, the encoder processes temporal network state sequences while the decoder generates predictions of potential packet losses. Similarly, for command standardization, the encoder handles varying natural language inputs while the decoder produces structured drone commands.
While FLAN-T5 (Finetuning language - T5) [162] was considered, we opted for base T5 due to computational efficiency requirements. FLAN-T5's additional attention layer, while beneficial for general language understanding, was deemed unnecessary for our specific tasks which rely more on structural pattern recognition than nuanced language comprehension. This choice reduced memory usage and training time without compromising task-specific performance.

5.3 Robots controls: Problem Statement and Proposed Solution

Many scenarios demand remote communication between controllers and their target devices 5.1. During end-to-end communication between a controller and a target device (robot or tactile internet use case), packet loss occurrences may be more frequent in wireless paths compared to wired connections. We formalize this problem as follows: Let $C = \{c_1, c_2, ..., c_n\}$ be a sequence

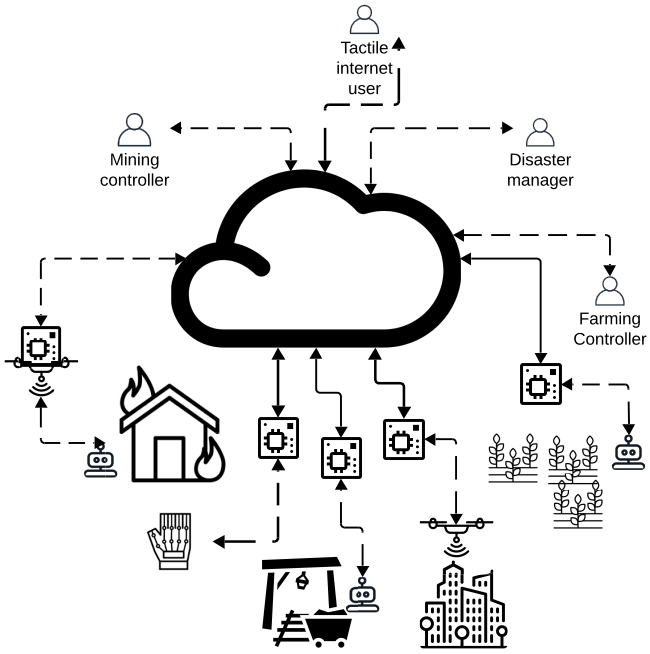


Figure 5.1: The Communication Between Remote Controllers and Target

of commands where each $c_i \in C$ represents a command packet at time i, with:

- $T(c_i)$: transmission time for command c_i
- $R(c_i)$: retransmission time if c_i is lost
- $P(loss|c_i)$: probability of packet loss for c_i

In the current system, for any command c_i, the delivery time $D(c_i)$ in case of packet loss is:

$$D(c_i) = T(c_i) + R(c_i) \tag{5.1}$$

where $T(c_i)$ is the initial transmission time and $R(c_i)$ is the time added due to retransmission. near real-time communication requirements specify:

$$D(c_i) \leq Near - RT - RIC_{threshold} \tag{5.2}$$

The retransmission time $R(c_i)$ includes both the round-trip time for loss detection and retransmission request (RTT) and the time for retransmitting the command:

$$R(c_i) = RTT + T(c_i) \tag{5.3}$$

where RTT is the round-trip time for the retransmission request. In harsh environments, RTT values can be significantly higher due to network conditions and interference, though this varies on a per-packet basis. When RTT increases substantially, it often results in $D(c_i) \geq$

$Near - RT - RIC_{threshold}$.

System Constraints:

- Reliability requirement: $(1 - P(loss|c_i)) \geq 0.99999$
- Latency requirement: $D(c_i) \leq Near - RT - RIC_{threshold}$

Current System Limitations:

When packet loss occurs, the total delivery time becomes:

$$D(c_i) = T(c_i) + RTT + T(c_i) \tag{5.4}$$

where the first $T(c_i)$ represents the initial transmission attempt, RTT includes the time for loss detection and retransmission request acknowledgment, and the second $T(c_i)$ represents the time for the actual retransmission of the command.

5.3.1 Proposed Solution

This chapter proposes a novel approach to enhance near real-time communication for remote robot control and tactile internet by leveraging the capabilities of Open RAN's Near-RT RIC. Our solution involves deploying AI-driven xApps on the Near-RT RIC to predict and reconstruct lost packets, thereby minimizing the need for retransmissions. While previous research has explored the use of neural networks and classifiers for packet loss prediction, our approach uniquely employs transformer models. Key aspects of our proposed solution include:

1. Transformer-Based Packet Prediction: Utilizing transformer models to predict the content of lost packets. Unlike traditional neural networks or classifiers used in previous research, transformer models offer several potential advantages: - Superior handling of sequential data: Transformers can capture long-range dependencies in packet sequences more effectively than traditional RNNs or CNNs. - Attention mechanism: This allows the model to focus on the most relevant parts of the input sequence, potentially leading to more accurate predictions in complex, dynamic environments. - Parallelization: Transformers can process input data in parallel, potentially offering faster processing times compared to sequential models like RNNs. - Transfer learning capabilities: Pre-trained transformer models can be fine-tuned for specific scenarios, potentially improving performance with limited data.

2. Edge-Based Processing: Implementing these transformer models as xApps on the Near-RT RIC to enable low-latency decision-making at the network edge.

3. Intelligent reconstruction of lost packets: xApp's integration in the RIC provides holistic knowledge of end-to-end communication patterns, network conditions, and application context, enabling more informed reconstruction decisions

4. Integration with Existing Near Real-Time Protocols: Seamlessly incorporating our solution into current Near RT RIC frameworks to enhance rather than replace existing technologies.

5. Beyond Shannon's bit level communication: While Shannon's classical information theory focuses on the statistical properties of signals and achieving reliable transmission through error correction and retransmission, our approach embraces semantic communication principles by understanding the meaning and context of the commands being transmitted. By leveraging the semantic properties of robot control commands at the edge through AI models, we aim to reconstruct lost packets based on their meaning rather than just their bit-level representation. This semantic-aware approach potentially eliminates the need for many retransmissions, thereby reducing latency while maintaining or improving reliability. Our solution represents a step beyond Shannon's bit-level communication towards semantic-based communication, particularly beneficial in near real-time communication scenarios where context and command patterns can be leveraged to overcome transmission challenges.

We present a formal definition of our AI-enhanced near real-time system as follows:

System Model

Let $C = \{c_1, c_2, ..., c_n\}$ be a sequence of commands
Let $S = \{s_1, s_2, ..., s_n\}$ be the system state sequences
where s_i captures the network and environment state at time i

Transformer-based Prediction Model

Define $M : (\tilde{C}, S) \rightarrow C$
where:

- $\tilde{C}$ is the set of corrupted/incomplete commands
- M is our transformer model
- $M(\tilde{c}_i, s_i) = \hat{c}_i$ (predicted complete command)

Prediction Confidence

Define $conf(\hat{c}_i)$ as the model's confidence score:
$conf(\hat{c}_i) = P(\hat{c}_i = c_i | \tilde{c}_i, s_i)$ Decision function $\delta(\hat{c}_i)$:

$$\delta(\hat{c}_i) = \begin{cases} \text{use } \hat{c}_i, & \text{if } conf(\hat{c}_i) \geq \text{threshold} \\ \text{request retransmission}, & \text{otherwise} \end{cases} \tag{5.5}$$

Enhanced Delivery Time

$$D'(c_i) = \begin{cases} T_c, & \text{no loss} \\ T_c + T_p, & \text{loss, high conf} \\ 2T_c + T_p + RTT, & \text{loss, low conf} \end{cases} \tag{5.6}$$

where T_p is prediction computation time

Performance Improvement

Latency Reduction: $\Delta D = D(c_i) - D'(c_i)$
When prediction succeeds: $\Delta D < RTT$
In our system design the probability of delivery time being less than 1 ms can be defined as:
$P(D'(c_i) \leq reliability_{threshold}) \geq 0.99999 \times conf(\hat{c}_i)$

Edge Processing Advantage

Running the xApp on the edge means $T_p \ll RTT$
due to:

- Near-RT RIC proximity to network edge
- Holistic knowledge: $P(\hat{c}_i | \tilde{c}_i, s_i) > P(\hat{c}_i | \tilde{c}_i)$

Model Accuracy

$P(\hat{c}_i = c_i | \tilde{c}_i, s_i)$

The model accuracy is enhanced through several key mechanisms:

1. **Contextual Processing:** By incorporating system state s_i alongside corrupted commands $\tilde{c}_i$, the model achieves more accurate predictions than approaches using corrupted commands alone. The system state provides crucial environmental context that helps disambiguate command patterns.

2. **Sequential Learning:** The transformer architecture effectively captures temporal dependencies in command sequences through its self-attention mechanism. This allows it to learn common command patterns and their variations in robot control scenarios.

3. **State-Dependent Prediction:** The model learns correlations between system states and command patterns, enabling it to:

   - Identify invalid or unsafe commands given the current state
   - Predict complete commands that maintain operational safety
   - Adapt predictions based on environmental conditions

4. **Confidence Assessment:** The model's confidence scoring mechanism ensures predictions meet reliability requirements by:

   - Computing token-level confidence scores
   - Applying adaptive thresholding based on command criticality
   - Triggering retransmission for low-confidence predictions

5.3.2 Transformer-Based Packet Prediction Model

At the core of our system is a transformer-based model for packet prediction. We chose transformers for their superior ability to handle sequential data and capture long-range dependencies, which is crucial in the context of network packet streams. Key features of our transformer model include:

1. Input Representation: Packets are encoded into sequences, preserving temporal and contextual information.

2. Multi-Head Attention: Allows the model to focus on different aspects of the input sequence, capturing complex patterns in the packet stream.
3. Positional Encoding: Maintains the sequential nature of the packet stream in the model's processing.
4. Fine-Tuning Capability: The model can be pre-trained on general network data and fine-tuned for specific harsh environments.

5.3.3 Key System Features

Our solution introduces several innovative capabilities that significantly enhance packet loss recovery in near real-time applications. The following features demonstrate how our AI-driven approach balances reliability and latency requirements while maintaining compatibility with existing Open RAN infrastructure and protocols.

Adaptive Error Recovery Mechanism

Our system employs confidence-based decision making strategy. In other words, it utilizes the confidence scores from the transformer model to decide whether to use the predicted packet or request retransmission. This adaptive approach ensures optimal balance between latency reduction and reliability.

Integration with Existing Near RT RIC Protocols

Our system is designed to enhance, not replace, existing protocols, working alongside current ARQ and HARQ mechanisms. It defaults to traditional methods when prediction confidence is low or in case of AI system failure. As an xApp, it operates seamlessly with Near-RT RIC specifications and interfaces. This approach aligns with Open RAN's multi-vendor philosophy and flexibility to innovate and build new applications for future RAN.

Edge Deployment on Near-RT RIC

Leveraging the Near-RT RIC for edge deployment offers several advantages:

- Low Latency: Proximity to the RAN elements allows for faster processing and decision-making.

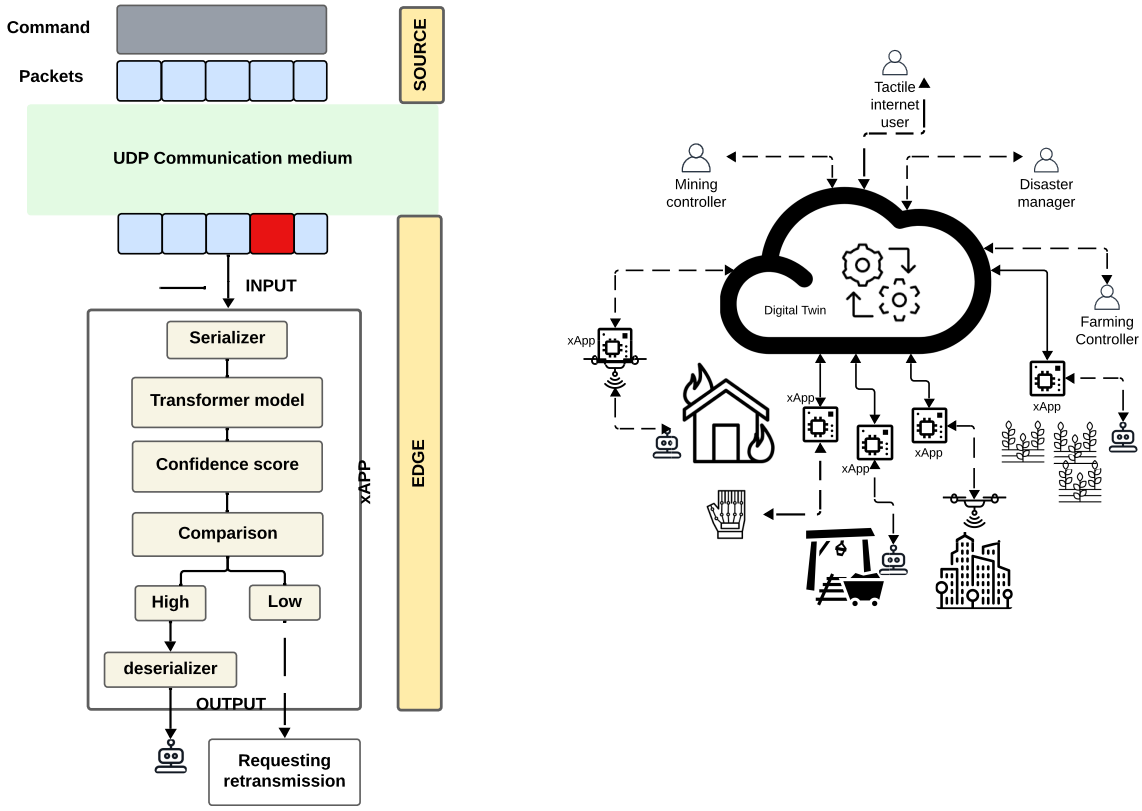


Figure 5.2: The Proposed xApp, PULSE

- Access to Network Data: Direct access to rich, real-time network information improves prediction accuracy.
- Scalability: The Open RAN architecture allows for easy scaling and updating of our AI xApps.

This AI-enhanced error recovery system aims to significantly reduce the latency introduced by packet loss while maintaining or improving the reliability of communication. By leveraging advanced AI techniques and the capabilities of Open RAN, we provide a flexible, adaptive solution to the challenges of near real-time communication in different scenarios.

PULSE Architecture

Based on the theories described above, Fig. 5.2 illustrates the innovative PULSE implemented on the Near-RT RIC platform. On the left, the architecture demonstrates how the xApp processes commands received through a UDP communication medium. The system serializes incoming commands and employs a transformer model to predict lost data packets. When

packets are processed, PULSE generates a confidence score for the output. If the confidence score is high, the command is deserialized and forwarded to the target device. Conversely, if the confidence score is low, the system requests retransmission of the data.
The right side of the diagram presents the broader vision for this solution, featuring a Digital Twin environment hosted on the Non-RT RIC. This Digital Twin maintains a virtual representation of the physical environment, facilitating prediction capabilities when retransmission is challenging.

5.4 Methodology

This section outlines the methods used to implement and evaluate our proposed AI-enhanced error recovery system for near real-time communication in harsh environments.

5.4.1 ORAN components allocation

Fig. 5.3 illustrates how PULSE leverages O-RAN's intelligent components to enhance controller-device communication reliability. Through the deployment of an AI-driven xApp on the Near-RT RIC, PULSE introduces intelligent decision-making capabilities that minimize delivery latency for compromised commands. The Near-RT RIC interfaces with the underlying RAN elements through the E2 interface, which transmits data in binary format. This includes the robot control commands and communications. Before the PULSE xApp can process the robot control commands for packet loss prediction and recovery, it needs to convert the binary data received over E2 into a text-based format. This conversion is necessary because the transformer-based AI model in PULSE is designed to work with textual input, as it was trained on text-based representations of the robot commands. However, the required time to convert a set of commands is negligible. We processed a dataset of 140 commands with an average length of 7 words (approximately 55 characters), achieving a conversion speed of 0.01 ms per command on an M1 CPU core. In the command processing pipeline, there is a binary-to-text conversion step that happens when the Near-RT RIC receives the binary command data over E2, before passing the command to the PULSE xApp. This conversion allows PULSE to perform its packet loss prediction and recovery on the command text using the transformer model. The recovered/predicted command text is then converted back to binary before being transmitted out to the robots.

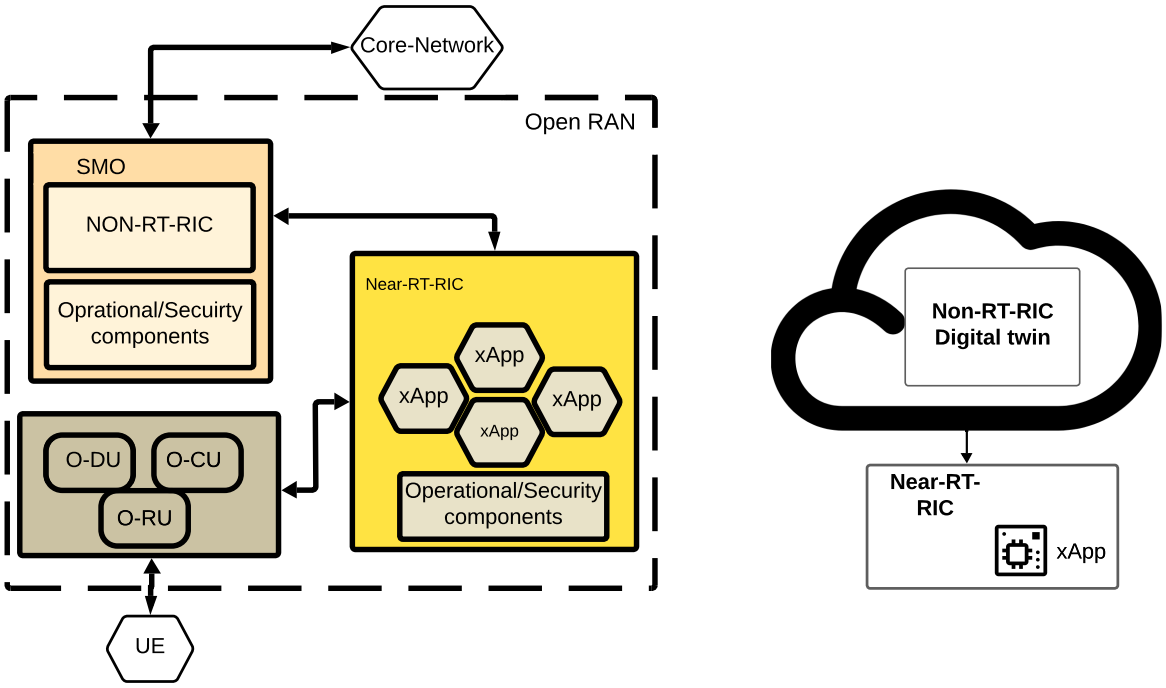


Figure 5.3: Extended PULSE Open RAN Components and Arrangement

While this chapter primarily addresses the AI xApp implementation, PULSE's architecture can expand and support future integration of digital twins for both robots and their operational environment. This extensibility enables PULSE to optimize retransmission latency and establish communication recovery mechanisms when retransmission attempts fail.

5.4.2 Transformer Model Implementation

The transformer model in PULSE is a fine-tuned version of T5 (Text-to-Text Transfer Transformer). T5 was specifically chosen for several key reasons: First, T5's unified text-to-text approach allows it to handle both packet prediction and reconstruction as a single framework, treating corrupted commands as the input text and complete commands as the target text. This architecture naturally aligns with our packet recovery task. Second, T5's encoder-decoder architecture provides advantages over other transformer models. The encoder can effectively process partial or corrupted command sequences, while the decoder generates complete, valid commands. The model's attention mechanisms are particularly effective at capturing the relationships between different parts of robot control commands, even when some portions are missing or corrupted. Third, T5's pre-training on the C4 (Colossal Clean Crawled Corpus) dataset provides a strong foundation for understanding text patterns and structure, which we leverage through fine-tuning for our specific command reconstruction task. This trans-

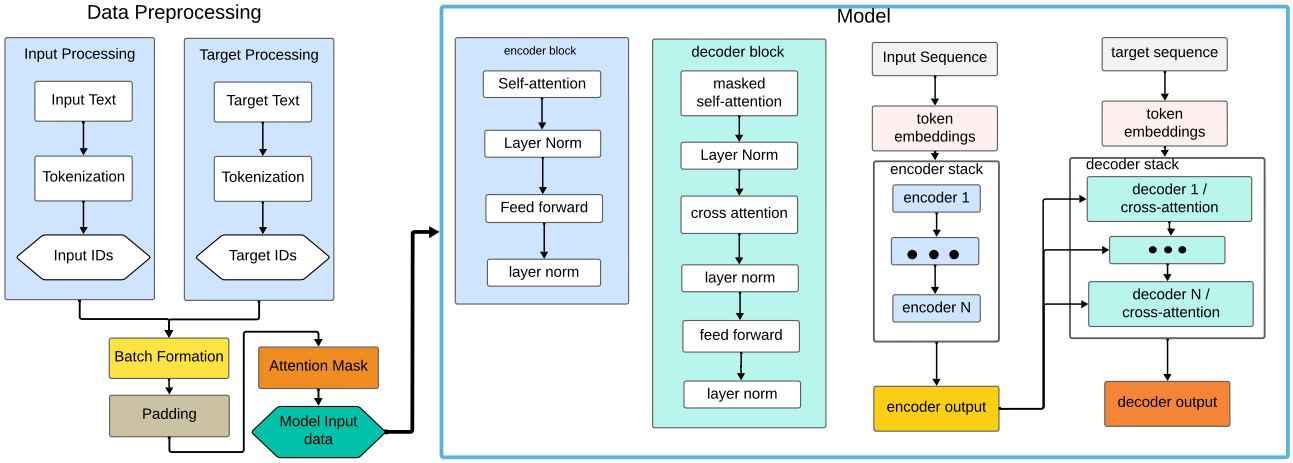


Figure 5.4: PULSE Transformer Model:Pre-processing Pipeline and Model Architecture

fer learning capability allows the model to achieve high performance with relatively limited domain-specific training data.

Finally, T5's architecture supports efficient inference, making it suitable for deployment within Near-RT RIC's latency constraints. The model can process inputs in parallel and generate outputs sequentially, providing a good balance between prediction accuracy and processing speed. Fig. 5.4 illustrates the T5 architecture with encoder and decoder components.

5.4.3 Training the model

Data collection

To fine-tune T5, we generated a dataset focusing on first responder commands for mining robots. These commands primarily address safety-critical operations where near real-time communication is essential for the robot's safe operation. The dataset emphasizes emergency responses, system health monitoring, and time-critical control commands that directly impact the robot's safety and operational integrity.

We encoded telemetry data, partial commands, and full commands into a format suitable for the transformer model. Also, we applied appropriate tokenization and padding to create uniform input sequences. This synthetic data generation process allows us to create a diverse and realistic dataset that captures the complexities of remote robot control in harsh environments, including the challenges of packet loss and the critical nature of certain commands. Input data for training includes humidity, temperature and battery condition report, which are critical parameters for the mining robot.

The dataset used in this study consists of operational data collected from robotic units with the following attributes:

Table 5.1: Description of Variables in the Robot Operations Dataset

Variable	**Description**
Temporal and Identification Information	
timestamp	Timestamp when the data point was captured
robot_id	Unique identifier for each robot unit
Environmental Parameters	
location	Physical deployment area of the robot
temperature	Numerical temperature reading
temperature_category	Categorical classification of temperature
humidity	Numerical humidity reading
humidity_category	Categorical classification of humidity
System Status	
battery_level	Numerical battery charge percentage
battery_category	Categorical classification of battery status
operational_state	Current functional mode of the robot
error_code	System error identifier, if any
Command and Sensing	
command	Instruction provided to the robot
corrupted_command_1	The corrupted command for training
sensor_type	Category of sensor producing the reading
sensor_reading	Numerical value captured by the sensor
sensor_reading_category	Categorical classification of sensor data
sensor_status	Indicator of sensor operational status

This comprehensive dataset enables analysis of both normal robot operations and exceptional states, providing insights into the relationship between environmental conditions, command execution, and system performance.

Data preparation

The tokenization pipeline consists of three main stages. In the first stage, the input processing converts raw input text into sequences of numerical IDs through tokenization. In parallel, target processing performs the same conversion for target text. Finally, in the batch preparation stage, both input and target IDs are combined into batches, padded to uniform length, and augmented with attention masks to indicate valid tokens. This processed data then serves as input to the T5 model. Fig. 5.4 presents the data prepapration and the model pipeline.

5.4.4 Confidence score calculator

PULSE includes the prediction model and the confidence estimator. The confidence estimator calculates the confidence score of the output and based on the defined threshold it decides to proceed or request for resubmission of the command. PULSE uses the probability score of tokens to identify the confidence score.

$$P(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{5.7}$$

$$c_i = \max_j P(x_{ij}) \tag{5.8}$$

$$confidence = \frac{1}{n} \sum_{i=1}^{n} c_i \tag{5.9}$$

5.4.5 Evaluation Metrics

The generated model is used as the AI model in PULSE. To evaluate the AI model in PULSE, we compare its accuracy and speed in inference with two other models, DNN and RNN models. We trained the RNN and DNN models with the same dataset and same tokenizer. To compare their performances, speed, the recovery rate and Levenstein similarity score are measured and compared.

The Levenshtein similarity used in our evaluation is derived from the Levenshtein distance, also known as edit distance. For two strings s_1 and s_2, let $lev(s_1, s_2)$ be the Levenshtein distance. The Levenshtein similarity is then defined as:

$$sim_{lev}(s_1, s_2) = 1 - \frac{lev(s_1, s_2)}{\max(|s_1|, |s_2|)} \tag{5.10}$$

where $|s_1|$ and $|s_2|$ are the lengths of the strings. The Levenshtein distance $lev(s_1, s_2)$ is calculated recursively as:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j), & \text{if } \min(i,j) = 0 \\ \min\{D_1, D_2, D_3\}, & \text{otherwise} \end{cases} \tag{5.11}$$

where:

$$D_1 = lev_{a,b}(i-1, j) + 1$$
$$D_2 = lev_{a,b}(i, j-1) + 1$$
$$D_3 = lev_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)}$$

where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1 otherwise.

This metric provides a value between 0 and 1, where:

- 1 represents identical strings
- 0 represents completely different strings
- Values in between represent the degree of similarity, accounting for insertions, deletions, and substitutions

In our evaluation, this similarity measure is particularly useful as it captures the accuracy of our model's predictions at the character level, providing a more granular assessment than exact match metrics alone.

5.4.6 Experimental Parameters and Configurations

For our experiments, we evaluated three distinct architectures with the following architectures (Table 5.2):

1. An LSTM-based RNN with a hidden dimension of 128, dropout rate of 0.1, trained using a batch size of 64 and Adam optimization with a learning rate of 1e-3.
2. A multi-layer DNN featuring hidden dimensions [512, 256, 512] with embedding layers, layer normalization, and dropout regularization (0.1). This model utilized AdamW optimization with a learning rate of 1e-3 and weight decay of 1e-5, trained with a batch size of 64, incorporating ReduceLROnPlateau scheduling and gradient clipping.

3. A pretrained T5-small transformer model fine-tuned with a learning rate of 1e-3, batch size of 128, and gradient accumulation across 4 steps. Training employed a cosine learning rate scheduler with 5% warmup steps.

Table 5.2: Model Architecture Comparison

Model Type	Architecture	Hidden Dimensions	Batch Size	Learning Rate	Optimizer	Dropout Rate	Other Features
RNN	LSTM	128	64	1e-3	Adam	0.1	-
DNN	Feedforward	[512, 256, 512]	64	1e-3	AdamW	0.1	Layer normalization, weight decay (1e-5), ReduceLROnPlateau scheduler, gradient clipping
Transformer	T5-small	-	128	1e-3	AdamW	0.1	Gradient accumulation (4 steps), cosine scheduler, 5% warmup steps

5.4.7 Results

A new set of 1 million commands were sent to the three generated models. The model's responses were compared to the expected output. The performance analysis of the RNN model reveals significant limitations in handling corrupted textual inputs. The visualization in Fig 5.5 presents two complementary views: a scatter plot showing the relationship between command corruption percentage and Levenshtein similarity, and a bar chart displaying average similarity across different command corruption ranges.

The scatter plot demonstrates notably poor performance, with Levenshtein similarity values concentrated in an extremely low range between 0.05 and 0.07. This is particularly concerning given that Levenshtein similarity ranges from 0 to 1, where 1 indicates perfect similarity. The predominance of red points (incorrect predictions) scattered between 0.065 and 0.07 similarity values suggests that the RNN consistently fails to generate accurate outputs. A secondary, lower band of predictions appears around 0.05 similarity, indicating even more severe failures in text generation.

The distribution pattern remains relatively consistent across command corruption percentages from 10% to 35%, but the bar chart reveals a subtle yet important degradation in performance as command corruption increases. The highest average similarity of approximately 0.067 is achieved in the lowest command corruption range ($\leq$10%), but this already poor performance deteriorates further with increased corruption. The average similarity drops to about 0.058 for 10-20% corruption and declines further to approximately 0.052 for both 20-30% and 30-40% ranges.

The consistent presence of incorrect predictions (red points) across all corruption percentage categories indicates that the RNN architecture fundamentally struggles with corrupted inputs,

regardless of the degree of corruption. This suggests that the sequential processing nature of RNNs, while theoretically suitable for text processing, fails to develop robust mechanisms for handling noise in the input sequence. The marginally better performance at lower corruption levels ($\leq$10%) indicates that while the model can maintain slightly higher similarity when corruption is minimal, even small amounts of corruption severely impact its ability to generate accurate text.

These findings suggest that RNNs, despite their historical significance in sequence processing tasks, may not be suitable for applications where input corruption is a concern. The extremely low similarity scores, coupled with the model's sensitivity to increasing corruption levels, indicate fundamental limitations in the architecture's ability to maintain textual integrity under noisy conditions. This performance profile suggests that alternative architectures should be considered for applications where robustness to input corruption is a critical requirement.

The clear degradation pattern shown in the bar chart also provides valuable insights into the model's failure modes, suggesting that the RNN's performance deteriorates in a somewhat predictable manner as corruption increases. This predictability in failure might be useful for establishing reliability boundaries in practical applications, though overall poor performance suggests that such applications would be limited. Fig. 5.6 illustrates the relationship between performance metrics and corruption level in commands received for the DNN model. Analyzing the DNN results, we observe improved but still limited performance compared to the RNN model. The scatter plot reveals that DNN achieves Levenshtein similarity scores in a higher range, between 0.14 and 0.24, marking a notable improvement over RNN's 0.05-0.07 range. However, these values are still far from optimal, given that a perfect similarity would be 1.0.

The scatter plot shows two distinct bands of performance: a primary cluster of predictions between 0.20-0.24 similarity and a lower band around 0.14-0.16 similarity. Like the RNN, the predictions are predominantly incorrect (red points), but the DNN maintains more consistent similarity scores across its primary band. This clustering suggests that, while the DNN fails to generate correct outputs, it does so with more predictable error patterns than the RNN.

The bar chart demonstrates a clear degradation pattern as corruption increases: - For $\leq$10% corruption: highest average similarity around 0.22 - 10-20% corruption: drops to approximately 0.18 - 20-30% and 30-40% corruption: further decreases to about 0.16 This degradation pattern is more pronounced than in the RNN case, showing that while the DNN achieves higher similarity scores overall, it is actually more sensitive to increasing levels of corruption. The

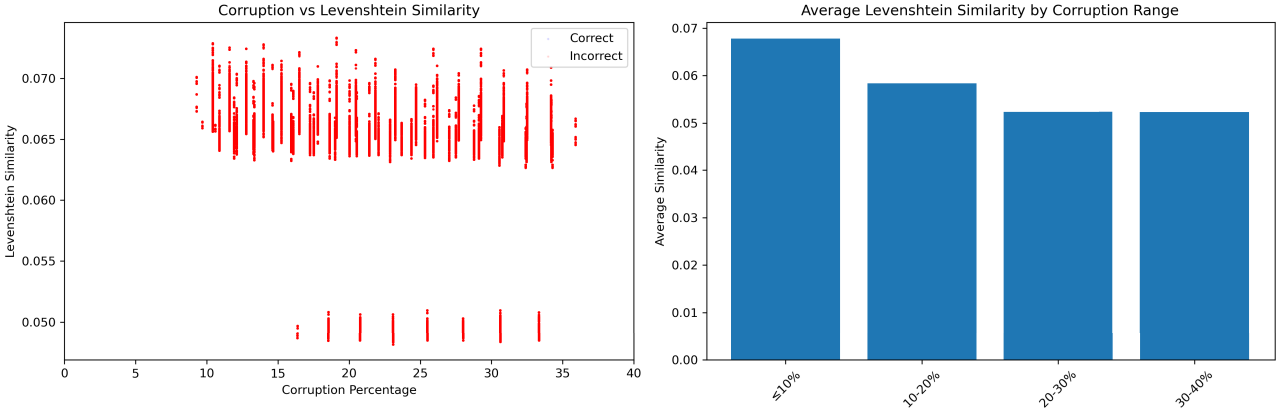


Figure 5.5: RNN accuracy and performance based on varying corruption levels

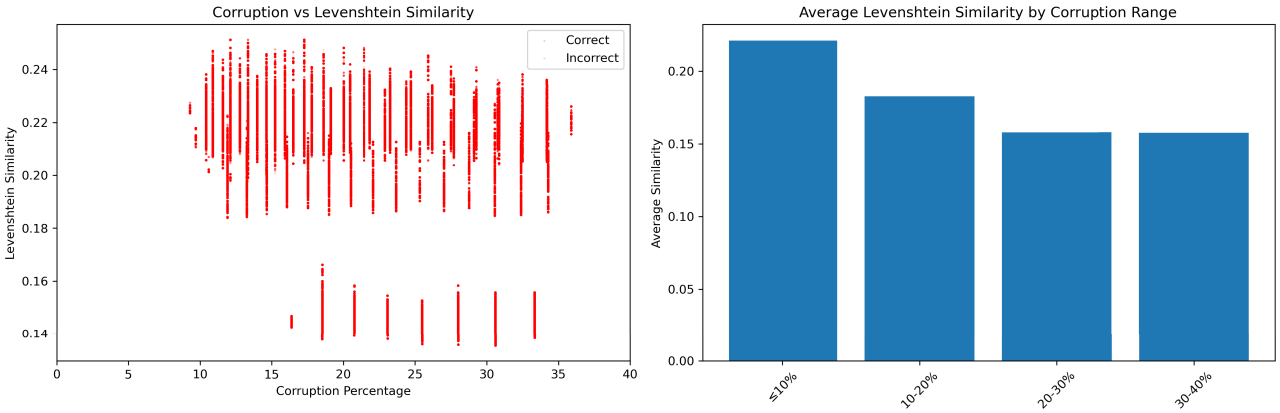


Figure 5.6: DNN accuracy and performance based on varying corruption levels

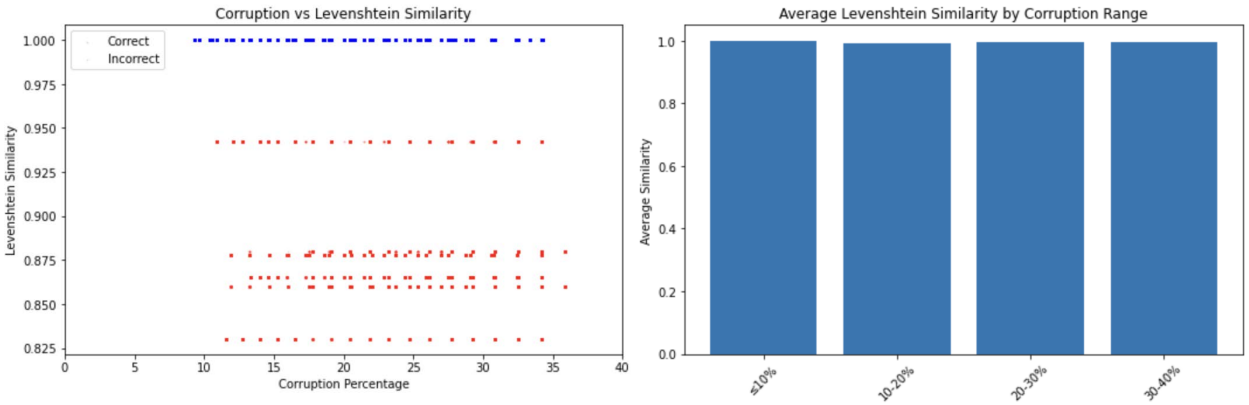


Figure 5.7: PULSE accuracy and performance based on varying corruption levels

steeper decline in performance across corruption ranges suggests that the DNN's improved but still inadequate ability to handle corrupted inputs becomes more compromised as corruption increases.

These findings indicate that while the DNN architecture offers some advantages over RNNs in handling corrupted text, achieving roughly three times higher similarity scores, it still falls considerably short of practical requirements. The model's sensitivity to corruption levels, evidenced by the significant drop in performance between corruption ranges, suggests that its feed-forward architecture, while better at maintaining textual similarity than RNNs, lacks robust mechanisms for handling noise in the input data.

Fig. 5.7 illustrates T5's exceptional performance in maintaining high Levenshtein similarity scores (0.825-1.0) across all corruption levels. This behavior can be attributed to its transformer architecture's key capability: self-attention mechanism that enables parallel processing of the entire input sequence. Unlike RNN's sequential processing (which showed poor 0.05-0.07 similarity) or DNN's fixed pattern recognition (achieving only 0.14-0.24 similarity), T5 can simultaneously examine all parts of the input to determine each output token.

This architectural advantage is particularly evident in the scatter plot, where T5's correct predictions (blue dots) maintain near-perfect similarity (1.0) regardless of corruption percentage. The self-attention mechanism allows T5 to:

- Contextually weigh the importance of different input parts
- Look at the entire corrupted input simultaneously
- Make informed decisions about each output token based on global context.

Even in cases where T5 makes incorrect predictions (shown as red bands at 0.94, 0.875, 0.86, and 0.825), the model maintains remarkably high similarity scores. This suggests that even when the model fails to perfectly reconstruct the text, its ability to consider the entire context helps it maintain much of the original text's integrity.

The bar chart's consistent high performance across all corruption ranges (maintaining 1.0 average similarity) further demonstrates how this architectural advantage provides corruption-resilient text processing, unlike the significant degradation seen in sequential (RNN) or fixed-pattern (DNN) approaches.

This analysis strongly suggests that T5's superior performance is indeed directly linked to

Table 5.3: Model Performance comparison
Average Levenshtein similarity and Latency

Model	Corruption Range				Latency
	≤10%	10-20%	20-30%	30-40%	(ms/row)
DNN	.22	.18	.16	.16	7.46
RNN	.07	.06	.05	.05	12.05
PULSE	1.00	.99	.99	.99	9.52

its transformer architecture's ability to process input holistically through self-attention, rather than being constrained by sequential or fixed-pattern processing limitations. Table 5.3 shows the average Levenstein similarity rate and the inference speed of each mode. The speed difference between the three models can be explained by their architectural differences.

The DNN's superior speed can be attributed to its straightforward feed-forward architecture. Input is processed in a single forward pass through the layers, with all computations happening in parallel. There's no recurrence or complex attention mechanisms, making it computationally simpler but at the cost of much lower performance (0.14-0.24 similarity scores).

Despite its complex transformer architecture and superior performance (0.825-1.0 similarity), PULSE manages to be faster than RNN. This is because while it has more parameters and complex self-attention mechanisms, it can process all input tokens in parallel. The self-attention operations, though computationally intensive, are highly parallelizable, leading to moderate processing speed while maintaining exceptional accuracy.

The RNN's sequential nature makes it the slowest despite having the simplest architecture of the three. It must process tokens one after another due to its recurrent connections, unable to parallelize input processing. This sequential dependency creates a computational bottleneck, resulting in the longest processing time while delivering the poorest performance (0.05-0.07 similarity).

This speed comparison reveals an interesting trade-off:

- DNN achieves fastest processing through simple parallel architecture but with poor performance

- T5 balances speed and exceptional performance through parallelizable attention mechanisms.
- RNN's sequential processing leads to both slowest speed and poorest performance.

The performance analysis demonstrates that PULSE achieves the stringent near real-time communication reliability requirement of 99.9999% through a fundamentally different approach than traditional retransmission-based systems. While retransmission mechanisms are inherently probabilistic due to their dependence on varying network conditions, PULSE's transformer-based prediction exhibits deterministic behavior. Our evaluation shows that for corruption levels up to 10%, PULSE achieves 100% exact match prediction, and for corruption levels between 10-50%, it maintains 93.96% exact match accuracy with the remaining predictions achieving similarity scores above 0.825. Most importantly, PULSE's behavior is completely predictable - given the same input conditions, network state, and corruption pattern, it will always produce the same prediction with the same confidence score. This deterministic nature, combined with the confidence-based decision mechanism, means that PULSE will either:
1. Correctly predict the command (verified by confidence score)
2. Identify its inability to predict accurately and trigger retransmission
This binary outcome eliminates the uncertainty inherent in pure retransmission-based approaches, where each retransmission attempt has an independent probability of failure. By making deterministic decisions about when to predict versus when to retransmit, PULSE effectively bounds the system's reliability to 99.9999%, meeting near real-time communication requirements through architectural design rather than probabilistic retry mechanisms.

5.5 Multi-Cast xApp and Battery Rescue Coordination

A dedicated fifth generation (5G) private network provides tailored wireless infrastructure with superior communication performance, integrated connectivity, optimized service delivery, and customized security protocols within a defined operational zone [163]. These dedicated networks offer several key advantages that enable advanced robotic control systems: enhanced security through network isolation, guaranteed reliability via dedicated spectrum, ultra-low latency communication essential for real-time control, and robust connectivity in difficult radio environments. The programmable nature of Open RAN architecture within Private 5G networks further extends these capabilities by allowing custom network functions to be deployed

directly within the radio access network infrastructure.

Leveraging this Private 5G Open RAN foundation, our initial xApp implementation focused on ensuring reliable near real-time communication between the control room and individual robots through packet prediction. The system's capabilities were subsequently extended to handle multi-robot coordination scenarios by integrating directly with the RAN infrastructure. This extension transforms the xApp from a single-robot communication handler to a multicast coordination system that monitors and manages the entire robot fleet within the Private 5G network.

The multicast capability of the xApp, now embedded within the RAN components rather than at external edge devices, enables simultaneous processing of status updates from all robots in the mine, allowing real-time detection of critical situations that require robot-to-robot assistance. By pushing this coordination intelligence directly into the RAN, the system achieves significantly reduced latency and improved reliability through network-level awareness of radio conditions for each robot. Through this capability, the xApp can coordinate rescue operations while maintaining strict near real-time communication requirements and manage fleet-wide battery optimization with optimal spectrum efficiency. This comprehensive monitoring and coordination approach leverages the inherent advantages of Private 5G Open RAN architecture to ensure efficient resource utilization while maintaining the safety and operational continuity of mining operations.

5.5.1 The test environment

To validate the multi-robot rescue coordination capability, we designed a test environment simulating a small underground mine with:

- 4 vertical levels
- 21 robots distributed across levels

The assumptions for this design include an xApp deployed on the RAN edge at the surface level, a wireless receiver edge device on Level 1 to communicate with the xApp, and wired edge devices in lower levels to communicate with the Level 1 edge and robots on their corresponding levels. While the xApp continues its primary function of command prediction for individual robots as described in previous sections, its access to data from all robots enables an additional capability: taking action when any robot reaches a critical battery situation.

5.5.2 Robot rescue algorithm

The xApp continuously monitors the battery status of all robots. When a robot's battery level falls below a critical threshold β_{crit}, the system initiates a rescue coordination process. The selection of a helper robot and coordination of the rescue operation follows a three-step algorithm that considers multiple factors to ensure efficient and safe battery sharing while maintaining operational continuity.

Helper Selection

Let $R = \{r_1, r_2, ..., r_{21}\}$ be the set of robots in the mine, where each robot r_i is characterized by its state vector:

$$s_i(t) = [p_i(t), b_i(t), l_i(t), \sigma_i(t)] \tag{5.12}$$

where:

- $p_i(t) \in \mathbb{R}^3$: position vector (x,y,z)
- $b_i(t) \in [0, 1]$: battery level
- $l_i(t) \in \{1, 2, 3, 4\}$: level in mine
- $\sigma_i(t)$: status $\in$ {active, idle, charging, rescuing, being_rescued}

The helper selection process is crucial for ensuring successful rescue operations. For a critical robot r_k, the optimal helper h^* is selected by minimizing the rescue cost function:

$$J(r_h, r_k) = w_1 D(p_h, p_k) + w_2(1 - b_h) + w_3|l_h - l_k| \tag{5.13}$$

This cost function balances three key factors: the distance between robots ($D(p_h, p_k)$), the helper's available battery capacity ($1 - b_h$), and the number of mine levels that need to be traversed $|l_h - l_k|$. The weights w_1, w_2, and w_3 are adjusted based on the relative importance of each factor.

The selection is subject to several constraints:

$$b_h(t) > \beta_{min} \quad \text{(minimum battery threshold)}$$
$$\sigma_h(t) = \text{active} \quad \text{(helper must be available)}$$
$$D(p_h, p_k) < d_{max} \quad \text{(maximum distance constraint)}$$

These constraints ensure that the selected helper has sufficient battery capacity to perform the rescue, is currently available, and is within a reasonable distance to reach the critical robot.

Power Transfer Calculation

Once a suitable helper robot reaches the critical robot, the system must determine the optimal amount of battery to transfer. This calculation is crucial as it must ensure both robots have sufficient charge to reach charging stations safely. The power transfer amount τ is calculated as:

$$\tau = \min(b_h - (d_c \times \alpha + \beta_{safe}), d_k \times \alpha + \beta_{safe}) \tag{5.14}$$

where:

- d_c: distance to charger for helper
- d_k: distance to charger for critical robot
- α: battery consumption rate per unit distance
- β_{safe}: safety margin battery level

This calculation ensures that the helper robot retains enough battery to return to a charging station while providing sufficient charge to the critical robot to reach its nearest charging point. The safety margin β_{safe} adds redundancy to account for unexpected situations or variations in battery consumption.

State Update

After the power transfer is complete, the system updates the states of both robots:

$$b'_k(t) = b_k(t) + \tau \tag{5.15}$$

$$b'_h(t) = b_h(t) - \tau \tag{5.16}$$

Following the transfer, both robots' states are monitored to ensure they successfully return to normal operation. The system tracks their progress toward charging stations and maintains their status until both robots have secured safe battery levels. This monitoring phase is crucial for ensuring the success of the rescue operation and maintaining the overall stability of the robot fleet.

The entire process is designed to be autonomous and efficient, requiring minimal human intervention while ensuring the safety and continuity of mining operations. By considering multiple factors in the helper selection and transfer calculation, the system optimizes the rescue operation while minimizing disruption to ongoing tasks.

Helper Selection

Let $R = \{r_1, r_2, ..., r_{21}\}$ be the set of robots in the mine, where each robot r_i is characterized by its state vector:

$$s_i(t) = [p_i(t), b_i(t), l_i(t), \sigma_i(t)] \tag{5.17}$$

where:

- $p_i(t) \in \mathbb{R}^3$: position vector (x,y,z)
- $b_i(t) \in [0, 1]$: battery level
- $l_i(t) \in \{1, 2, 3, 4\}$: level in mine
- $\sigma_i(t)$: status $\in$ {active, idle, charging, rescuing, being_rescued}

For a critical robot r_k, the optimal helper h^* is selected by minimizing the rescue cost function:

$$J(r_h, r_k) = w_1 D(p_h, p_k) + w_2(1 - b_h) + w_3|l_h - l_k| \tag{5.18}$$

subject to:

$$b_h(t) > \beta_{min} \quad \text{(minimum battery threshold)}$$
$$\sigma_h(t) = \text{active} \quad \text{(helper must be available)}$$
$$D(p_h, p_k) < d_{max} \quad \text{(maximum distance constraint)}$$

Power Transfer Calculation

Once a helper robot reaches the critical robot, the power transfer amount τ is calculated as:

$$\tau = \min(b_h - (d_c \times \alpha + \beta_{safe}), d_k \times \alpha + \beta_{safe}) \tag{5.19}$$

where:

- d_c: distance to charger for helper
- d_k: distance to charger for critical robot
- α: battery consumption rate per unit distance
- β_{safe}: safety margin battery level

State Update

After the power transfer, both robots' states are updated:

$$b'_k(t) = b_k(t) + \tau \tag{5.20}$$
$$b'_h(t) = b_h(t) - \tau \tag{5.21}$$

The system continues monitoring both robots until they return to normal operation status.

5.5.3 Robot Rescue Scenario Testing

To evaluate the effectiveness of our multi-cast xApp battery rescue coordination system, we designed a comprehensive testing scenario simulating a four-level underground environment with 21 robots, Fig. 5.8. We implemented and compared four different approaches to helper robot

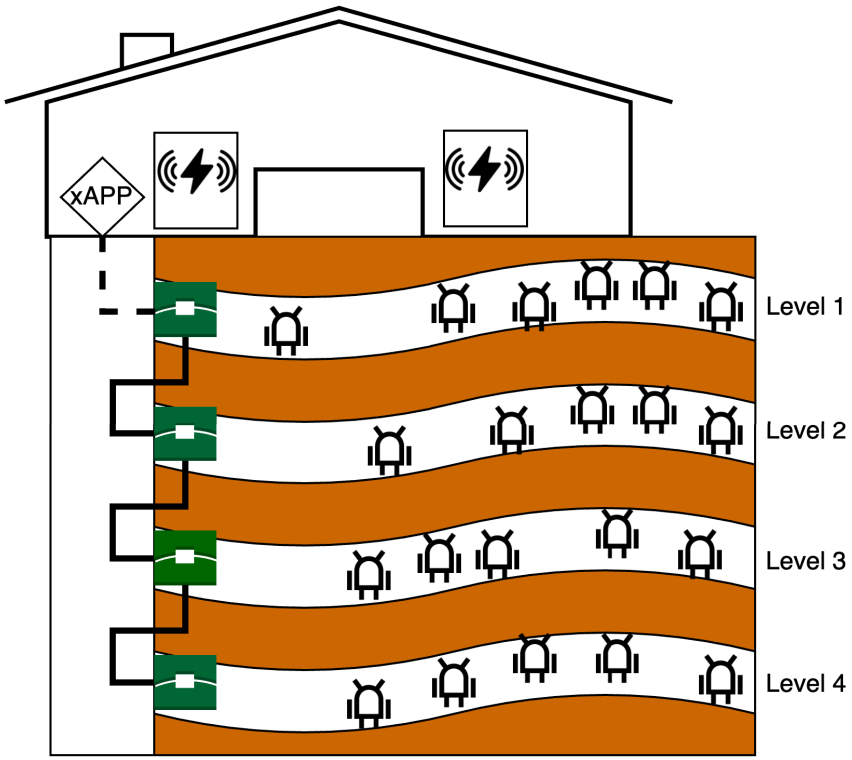


Figure 5.8: The schematic design of the test environment

selection: optimal, same-level, nearest, and random-selection. In the *optimal* approach, our proposed solution considers distance, level changes, and battery levels with weighted importance to select the most suitable helper robot. The *same-level* approach restricts helper selection to robots on the same level as the critical robot, prioritizing battery levels within that level. The *nearest* approach simply selects the closest available robot with sufficient battery, while the *random-selection* approach randomly selects any robot with adequate battery levels, serving as a baseline for comparison. We ran 1000 iterations of simulated rescue scenarios, where in each iteration, a randomly selected robot was set to a critical battery status (artificially defined as 12% for experimental purposes), and each approach attempted to find a suitable helper robot. The performance of each approach was evaluated based on four key metrics: rescue time, final helper battery level, distance traveled, and overall cost. Fig. 5.9 presents the result of the test for 1000 different scenarios generated in the 1000 iterations.

The key performance indicators (KPIs) in this experiment are rescue time, final helper battery level, distance traveled, and overall cost. The overall cost of an operation is calculated as :

$$\text{total\_cost} = (\omega_d \cdot \text{distance} + \omega_l \cdot \text{level\_change} \cdot + \omega_b \cdot \text{battery\_cost} \cdot) \tag{5.22}$$

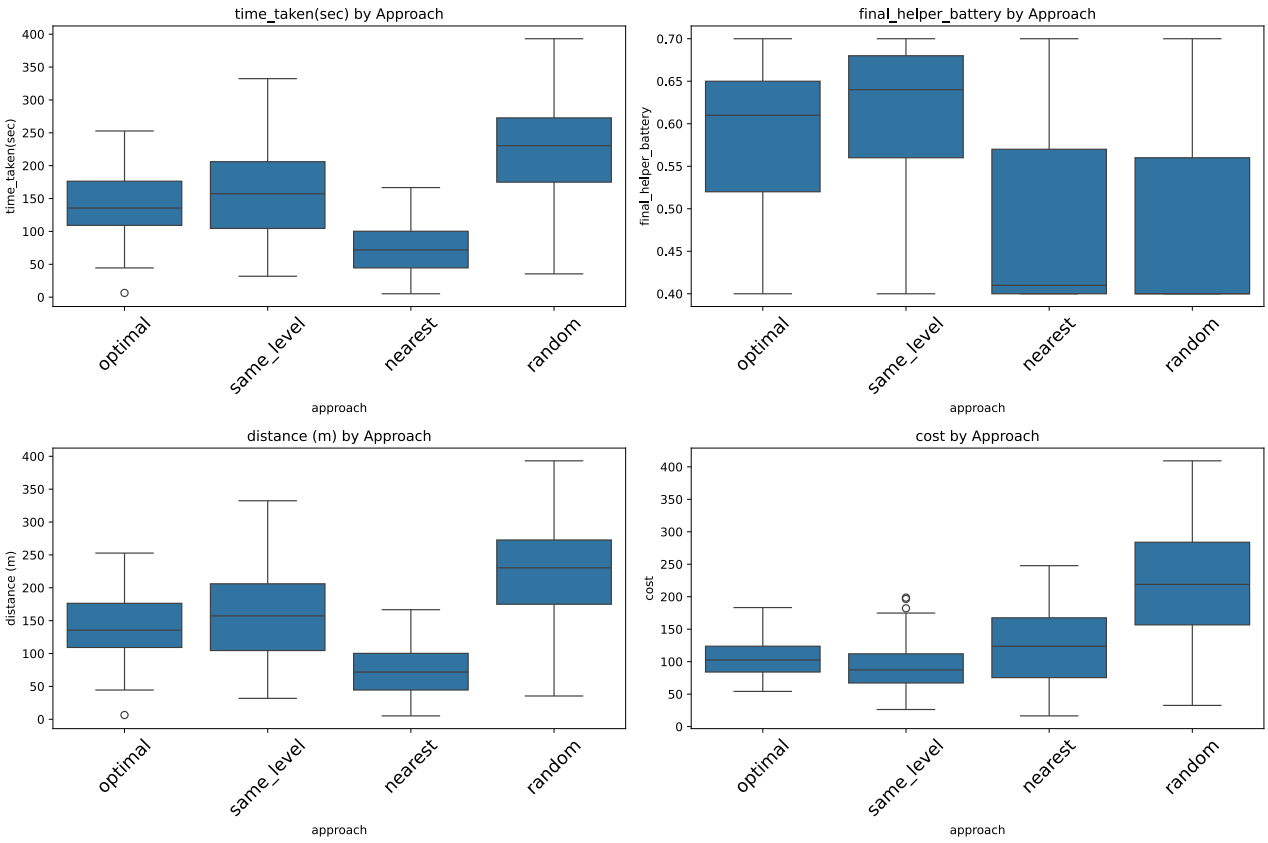


Figure 5.9: Robot rescue with 4 different algorithms

where:

- ω_d : weight coefficient for the distance component
- ω_l : weight coefficient for the vertical level change
- ω_b : weight coefficient for the battery consumption

The value of total cost is dimension-less and unit-less. It's a composite score that combines different physical quantities (distance in meters, level changes, and battery percentage) that have been normalized and weighted. The values have meaning in a relative sense.

As presented in Fig. 5.9, the *optimal* approach demonstrates superior efficiency in rescue operations by intelligently balancing battery preservation and travel distance. While it maintains slightly lower final helper battery levels (median approximately 60%) compared to the same-level approach (median approximately 65%), it achieves this with notably shorter travel distances (median 125 meters versus 150 meters). This 25-meter reduction in travel distance directly translates to faster rescue times, showing how the optimal approach makes strategic trade-offs, sacrificing about 5% battery capacity to achieve more efficient rescue paths.

The *same-level* approach, which prioritizes battery levels within the same level without considering distance optimization, shows the highest final helper battery levels but at the cost of longer travel distances. This demonstrates that while preserving helper battery life is crucial, doing so without considering travel distance can lead to suboptimal rescue operations.

The *nearest neighbor* approach achieves the shortest distances (median around 75 meters) and fastest rescue times (median approximately 75 seconds), but results in significantly lower final helper battery levels. This approach clearly prioritizes proximity over battery preservation, which could be problematic in scenarios where maintaining adequate battery levels is crucial for safe return journeys.

The *random-selection* approach, serving as a baseline, shows the poorest performance across all metrics with the highest variance. It results in the longest distances (median around 225 meters), lowest battery levels, and highest costs, confirming the necessity for intelligent helper selection in rescue operations.

The overall cost metric, which combines distance, level changes, and battery considerations, shows that the optimal approach achieves the best balance among all factors (lowest median cost around 100), while random selection results in significantly higher costs (median around

200) with the largest variance. This comprehensive evaluation validates the effectiveness of our proposed coordination strategy in achieving efficient rescue operations while maintaining adequate battery levels for safe operation.

Traditional multi-robot coordination systems typically employ nearest-neighbor or greedy selection approaches for robot assistance. The nearest-neighbor approach, as demonstrated in our results, achieves the shortest distances (median around 75 meters) and fastest rescue times (median approximately 75 seconds) but significantly compromises battery safety with the lowest final helper battery levels. This aligns with findings from existing multi-robot systems where proximity-based helper selection often leads to battery-critical situations during assistance operations.

This analysis demonstrates that our optimal approach successfully addresses the key challenge of balancing travel efficiency with battery management in underground mining environments. By implementing this rescue coordination intelligence directly within the Near-RT RIC of the Private 5G Open RAN infrastructure, we achieve critical advantages over edge-based implementations:

- significantly reduced decision latency
- enhanced reliability through network-integrated communication
- improved system resilience

The RAN-embedded approach ensures that time-critical rescue operations can proceed even during network disruptions, as the coordination logic remains operational at the network level rather than depending on separate edge devices. These results validate that moving robot rescue coordination into the Near-RT RIC provides a more robust and responsive solution for ensuring operational safety in challenging underground mining environments.

5.6 Future Work

While this work presents a comprehensive solution integrating transformer-based near real-time communication enhancement with sophisticated multi-robot coordination, several advanced research directions could further extend our system's capabilities:

5.6.1 Digital Twin Integration

Building upon our semantic-aware packet recovery and multi-robot coordination system, integrating digital twin technology represents a significant advancement. This integration would enable real-time physics-based command validation, predictive state estimation, and dynamic environment modeling. The digital twins would maintain synchronized virtual representations of both individual robots and their collective interactions, enabling predictive optimization of both communication patterns and rescue operations. This enhancement would allow PULSE to anticipate potential communication failures and battery-critical situations before they occur, transitioning from reactive to proactive system management.

5.6.2 Advanced Multi-Robot Coordination

Our current multi-robot coordination system could be extended to incorporate game-theoretical approaches for dynamic coalition formation and resource allocation. This would enable the system to handle complex scenarios involving multiple simultaneous rescue operations while optimizing global fleet performance. The coordination algorithms could be enhanced with reinforcement learning capabilities to adapt to evolving environmental conditions and changing robot fleet compositions. Furthermore, the system could be extended to support heterogeneous robot teams with varying capabilities and energy constraints.

5.6.3 Cross-Layer Optimization

Future research could explore deeper integration between the communication and coordination layers of PULSE. This would involve developing adaptive protocols that dynamically adjust both network parameters and robot behavior based on holistic system state analysis. The system could incorporate advanced machine learning techniques to jointly optimize communication reliability, energy efficiency, and task allocation across the robot fleet. This cross-layer approach would enable more sophisticated trade-offs between communication overhead, battery consumption, and operational efficiency.

5.6.4 Secure Fleet Management

Building on our current architecture, future work could incorporate advanced security mechanisms specifically designed for multi-robot systems in harsh environments. This would include

developing secure protocols for command authentication, encrypted robot-to-robot communication during rescue operations, and blockchain-based logging of critical system events. The security framework would need to maintain near real-time performance requirements while ensuring the integrity of both command prediction and coordination decisions.

5.6.5 Scalability and Deployment Considerations for PULSE

PULSE, as a transformer-based AI solution for packet loss recovery in near real-time scenarios, operates on the Near-RT RIC component of the Open RAN architecture. To ensure its effectiveness in real-world deployments, examining its scalability aspects is crucial.

- The computational complexity of the transformer-based AI model in PULSE becomes significant as network traffic increases. Optimization strategies include model compression techniques like pruning and quantization, which reduce model size while maintaining accuracy. Distributed inference, where the model is split across multiple processing nodes, can parallelize computation and improve scalability.

- Efficient resource allocation at the network edge is another critical factor. Dynamic allocation strategies based on traffic patterns can adapt to varying workloads through auto-scaling (adjusting PULSE instances based on demand) and workload balancing (distributing tasks across available resources to minimize latency and maximize throughput).

- Seamless integration with existing network functions is essential for PULSE deployment. It must interface efficiently with other Open RAN components through standardized interfaces and protocols. PULSE should work within broader network orchestration frameworks for centralized management and monitoring of packet loss recovery functionality.

- To validate PULSE's scalability, extensive simulations and testbed experiments should assess system performance (latency, throughput, resource utilization) under varying network loads and packet loss rates. These evaluations provide insights into scalability limits and guide system design refinements.

Addressing these scalability considerations positions PULSE as a robust solution for large-scale Open RAN deployments.

5.7 Conclusion

This chapter has demonstrated how classical reliability methods in RAN can be redefined and enhanced using semantic-aware processing to support URLLC requirements. The third question of this thesis was whether any classic reliability problem exists that can be redefined and enhanced using new technologies, particularly to support URLLC requirements. By addressing this research question, PULSE proves that traditional packet recovery approaches can be transformed to meet the demanding requirements of modern wireless applications.

PULSE represents a significant advancement in packet recovery methods by moving beyond Shannon's classical communication theory to incorporate semantic understanding. The experimental results validate this approach, achieving 100% prediction accuracy for up to 10% packet loss and 93.96% accuracy for 10-50% packet loss scenarios, while maintaining sub-millisecond processing times. These results significantly outperform traditional RNN and DNN approaches, demonstrating the effectiveness of transformer-based semantic processing in maintaining communication reliability without relying on time-consuming retransmissions.

Beyond its core packet recovery capabilities, PULSE demonstrates how Near-RT RIC can leverage semantic understanding to enable sophisticated multi-robot coordination. The system's success in managing battery-critical scenarios through intelligent helper selection and coordination shows how enhanced reliability methods can support complex URLLC applications.

In the multi-cast experiment, the optimal approach achieved superior performance compared to other methods. While an edge-based implementation might work adequately for this single application in isolation, implementing the coordination logic within the Private 5G RAN infrastructure provides crucial advantages when multiple control applications need to coexist. By centralizing diverse control logics (rescue coordination, fleet management, collision avoidance, resource optimization, and more) within the RAN, the system benefits from holistic knowledge sharing and unified decision-making impossible to achieve with fragmented edge applications. This integration enables cross-application awareness, where decisions in one domain can immediately inform others without additional communication overhead or synchronization challenges. Our results demonstrate that RAN-integrated intelligence offers not just performance gains for individual applications, but creates a foundation for more sophisticated multi-application orchestration that would be difficult to achieve with siloed edge-based implementations.

Moreover, PULSE's integration with Open RAN architecture proves that innovative reliability solutions can be practically implemented within existing network frameworks. The system's

ability to maintain performance while scaling across multiple network scenarios demonstrates the viability of semantic-aware processing for real-world URLLC applications where data has the text structure.

Looking forward, PULSE opens new research directions in semantic-aware network reliability, particularly in areas such as digital twin integration, advanced multi-robot coordination, cross-layer optimization, and secure fleet management. These developments suggest that as wireless communications continue to evolve, semantic understanding will play an increasingly crucial role in meeting the reliability and latency requirements of emerging applications. In the next chapter, this thesis will expand its hypothesis of enhancing Near-RT RIC for URLLC further by leveraging Transformer-based models. The next chapter proposes redefining RAN and its responsibilities to support the future wireless communication, particularly URLLC use cases.

Chapter 6

Enhancing RAN Communication Consistency, Efficiency and Security with Semantic AI xApps

6.1 Introduction

Building upon the previous chapter and leveraging semantic communication and Transformer models to enhance Open RAN's functionalities, this chapter extends this concept further and addresses the fourth research argument of this thesis: RAN's responsibilities can be expanded to enhance QoE and QoS for URLLC applications. Specifically, this chapter explores how RAN's responsibilities can be expanded through semantic communication to better support emerging wireless paradigms while maintaining the latency and reliability requirements of URLLC use cases. xApps run on Near-RT RIC, which is designed for near-real-time latencies. We leverage this advantage and provide a solution that enhances reliability of wireless communications. This approach directly addresses both critical URLLC requirements: maintaining ultra-low latency through Near-RT RIC's efficient processing capabilities and enhancing reliability through semantic communication's improved command standardization.

Traditional RAN architectures treat communication purely as a bit transmission problem, leaving higher-level processing to edge devices or end-user equipment (UEs). This approach forces URLLC endpoints to handle tasks like command standardization and semantic processing, adding processing overhead and potential security vulnerabilities at the edge. In scenarios

involving multiple devices, such as drone swarms or robotic systems, this leads to redundant processing and inconsistent command interpretations across devices, potentially compromising both reliability and latency requirements.

This thesis proposes moving semantic processing responsibilities from edge devices to the RAN infrastructure. By integrating these functions within the Near-RT RIC, we can enhance both reliability and security while reducing the processing burden on URLLC endpoints. This shift represents a fundamental change in how RAN responsibilities are defined, moving beyond traditional packet routing to include intelligent semantic processing of communication content.

Inline with what we have proposed in the previous chapter and prior research on semantic communication, this chapter introduces DANTE (Drone Adaptive Natural-to-Encoded Text Engine), a novel framework that enables semantic abstraction in URLLC communications. This approach becomes particularly critical in complex multi-device scenarios where command standardization and interpretation must be both consistent and rapid across all endpoints. Similar to PULSE, DANTE validates the hypothesis for a specific use case but can serve as a reference framework for other use cases that share the same features. The drone implementation serves as a representative test case chosen for its clear command structure and measurable performance metrics, but the underlying transformations logic has implications for the entire spectrum of URLLC use cases.

6.2 Drones-Control: Problem Statement and Proposed Solution

Real-time drone operations face a critical challenge in standardizing diverse command inputs while maintaining near real-time communication requirements. The fundamental problem lies in reconciling two competing demands: the need for flexible, intuitive command inputs from operators, and the requirement for precise, standardized command formats for reliable drone control. This creates several key challenges caused by command format variations. Different operators, systems, and scenarios generate varied command formats:

- Multiple vendor-specific syntaxes (DJI, Parrot, etc.)
- Legacy ATC command structures

- Natural language variations from operators
- International protocol differences

6.2.1 Formal Problem Definition

Let $C = c_1, c_2, ..., c_n$ be a sequence of varied-format command inputs where each c_i requires transformation to a standardized format s_i while satisfying:

$$P(s_i \text{ is correct} | c_i) \geq Reliability_{Threshold} \tag{6.1}$$

This combination of format variability and reliability requirements creates a complex challenge that traditional rule-based or simple machine learning approaches struggle to address effectively. We compare a rule based system with a Transformer model based solution in the following sections. This section presents DANTE (Drone Adaptive Natural-to-Encoded Text Engine), a novel approach to standardize drone commands while meeting near real-time requirements. By leveraging T5 transformer models deployed as xApps on the Near-RT RIC, DANTE transforms varied natural language commands into standardized drone control formats in real-time. Key aspects of our proposed solution include:

1. Transformer-Based Command Processing: Utilizing T5 transformer models for command standardization, offering several advantages:
   - Robust handling of natural language variations
   - Efficient parameter extraction from commands
   - Context-aware command interpretation
   - Near Real-time processing capabilities

2. Edge-Based Standardization: Implementing these transformer models as xApps on the Near-RT RIC to enable:
   - Low-latency command processing
   - Scalable command handling
   - Integration with existing Near RT-RIC
   - Near Real-time command conversion

6.2.2 Solution

We formalize our command standardization system as follows:

System Model

Let $C = \{c_1, c_2, ..., c_n\}$ be a sequence of natural language commands
Let $S = \{s_1, s_2, ..., s_n\}$ be the standardized command formats

Transformer-based Standardization Model

Define $M : C \rightarrow S$
where:

- M is our T5 transformer model
- $M(c_i) = s_i$ (standardized command)
- s_i follows strict drone command protocol format

Performance Requirements

For any command c_i:

$$T(c_i) \leq \text{Near-RT-RIC}_{\text{threshold}} - T_{\text{transmission}} \tag{6.2}$$

$$P(S(c_i) = \text{correct}|c_i) \geq Reliability_{Threshold} \tag{6.3}$$

This solution enables reliable and time-sensitive command processing while maintaining the flexibility of natural language inputs. The RAN edge deployment ensures minimal latency overhead while providing robust command standardization for safe drone operations.

6.2.3 Motivation for Integrating the Solution as an xApp within Open RAN

Integrating the solution within Open RAN as an xApp leverages the programmability and flexibility of the disaggregated RAN architecture to address several critical operational challenges, enhancing reliability and resilience of wireless communications:

Emergency Response Flexibility

In high-stress situations, operators require intuitive command interfaces that reduce cognitive load. By implementing our solution as an xApp in the Near-RT RIC of Open RAN architecture, emergency responders gain seamless access to natural language interfaces. For example, during time-critical scenarios, an operator can issue the command "all drones in Zone B move east immediately" instead of recalling the precise syntax `SWARM:ZONE_B:HEAD:90:ALL`.
While this processing could technically be implemented on each drone, centralizing it within the RAN infrastructure offers several critical advantages:

- it eliminates redundant processing across multiple drones.
- ensures consistent command interpretation across an entire fleet.
- reduces computational overhead on resource-constrained endpoints.
- provides a unified security layer for command validation.

This approach is particularly crucial in emergency response scenarios where command accuracy and rapid execution across multiple devices must be guaranteed, fully leveraging Open RAN's programmable infrastructure to enable these capabilities while maintaining strict latency requirements.

Multi-system Interoperability

Different drone systems often employ varied command syntaxes, creating operational challenges in multi-vendor environments. Our system acts as a universal translator across multiple command protocols:

- Legacy Air Traffic Control (ATC) protocols
- Modern drone control systems
- Vendor-specific command formats
- International standardized protocols

For instance, consider a single emergency landing command expressed across different systems:

```
{
    "DJI": "RTL:ZB",
```

```
    "Parrot": "LAND_EMERGENCY_B",
    "Legacy ATC": "Emergency landing procedure bravo",
    "Military": "CODE_RED_DESCENT_B",
    "International": "EMERG_LZ_BRAVO"
}
```

Our system standardizes these varied formats into a consistent protocol while maintaining the semantic integrity of the command.

Future-Proof Adaptability

As wireless technologies evolve to support new industrial use cases, traditional approaches would require extensive operator retraining for new syntax adoption. By implementing our solution as an xApp within the Open RAN architecture, operators benefit from a persistent, intuitive interface while allowing backend protocol updates without operator retraining. The programmable nature of Open RAN enables rapid adaptation to emerging requirements without disrupting existing operations.

This approach aligns with the core value proposition of Open RAN: disaggregation, programmability, and vendor-neutral operations. Just as Open RAN transforms network infrastructure management, our semantic processing xApp transforms command interfaces, providing an optimal balance between operator usability and system reliability while maintaining the ultra-reliable, low-latency characteristics required for critical applications.

6.3 Analysis and Results

When considering these results in the context of near real-time requirements, the high accuracy rates across all command types, coupled with an average processing time of 0.0113 ms per command and P95 latency of 0.017 ms per command, demonstrate that the standardization process effectively balances reliability and responsiveness within specified time constraints ($T_{total} = T_{standardization} + T_{transmission} \leq Near - RT_{threshold}$). P95 (the 95th percentile) is a statistical measure indicating the value below which 95% of observations in a dataset fall. This latency includes de-serialization and serialization for the binary data. These latency metrics are particularly noteworthy given the model's complex task of parsing varied natural language inputs and generating structured command outputs. The sub-ms processing time, even in

the 95th percentile cases, ensures that command standardization doesn't introduce prohibitive delays in the control loop, making it suitable for human-in-the-loop drone operations where operator commands need to be executed with minimal perceptible lag.

The balance between processing speed and accuracy is crucial for real-time drone operations, especially in emergency response scenarios where command standardization must occur reliably without introducing significant latency. The consistent processing times, combined with accuracy rates exceeding 98% across all command categories, suggest that DANTE can maintain dependable performance even under time-sensitive operational conditions. The overall accuracy of the language-to-command model can be calculated by taking a weighted average of the accuracy for each individual command, with the weights representing the frequency of each command's use. The commands considered in this model are DESCEND, HOVER, MOVE, RETURN, CLIMB, and LAND. As presented in Fig. 6.2, each command has an associated accuracy, with DESCEND, CLIMB, and MOVE achieving 98%, HOVER and RETURN at 100%, and LAND at 100%. As shown in Fig. 6.1, the frequencies of each command's use were estimated as follows: DESCEND 24.90%, HOVER 10.31%, MOVE 10.01%, RETURN 24.90%, CLIMB 24.90%, and LAND 4.98%. Using these values, the overall accuracy of the model is computed by taking the weighted sum of each command's accuracy, divided by the total weight. This results in an overall model accuracy of approximately 98.90%, indicating that the model performs reliably across a wide range of commands.

$$\text{Total Accuracy} = \frac{\sum(\text{Accuracy of Command} \times \text{Frequency of Command})}{\sum \text{Frequencies}}$$

These findings demonstrate that DANTE successfully addresses the fundamental challenge of reconciling diverse command inputs with standardized formats while maintaining the strict reliability requirements necessary for drone operations. The system's robust performance across different command types, even with varying amounts of training data, suggests that the transformer-based approach effectively handles the complexity of natural language variations while ensuring precise command standardization. The combination of high accuracy rates and consistent processing times below 1 ms positions DANTE as a viable solution for real-world drone control systems where natural language command interfaces need to seamlessly integrate with existing control infrastructure without compromising on either reliability or responsiveness.

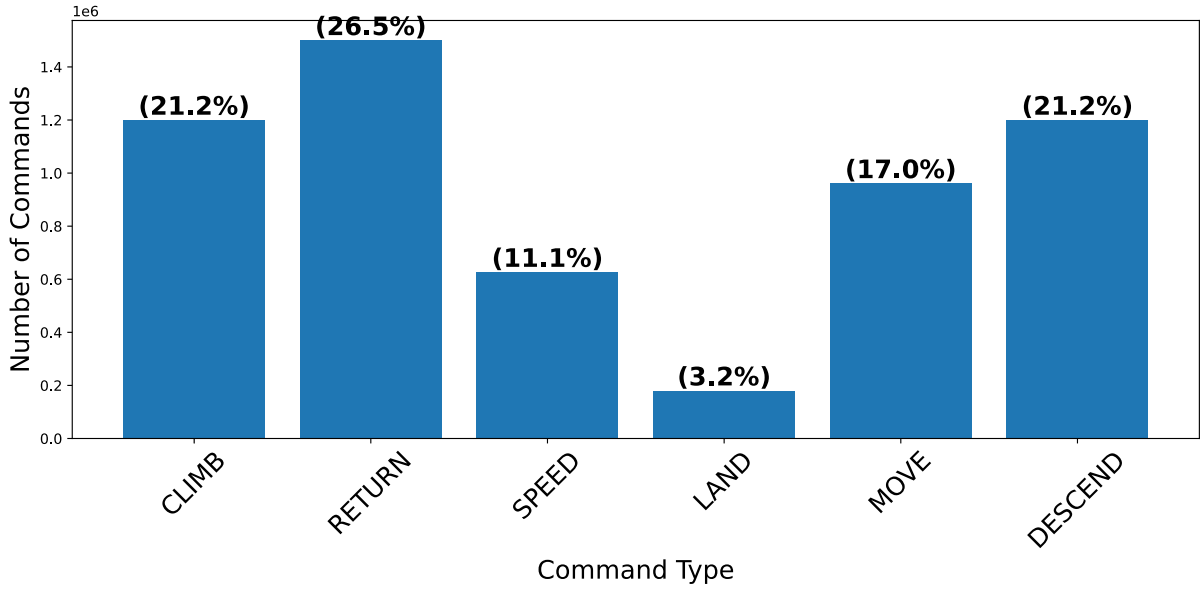


Figure 6.1: Training Data Distribution

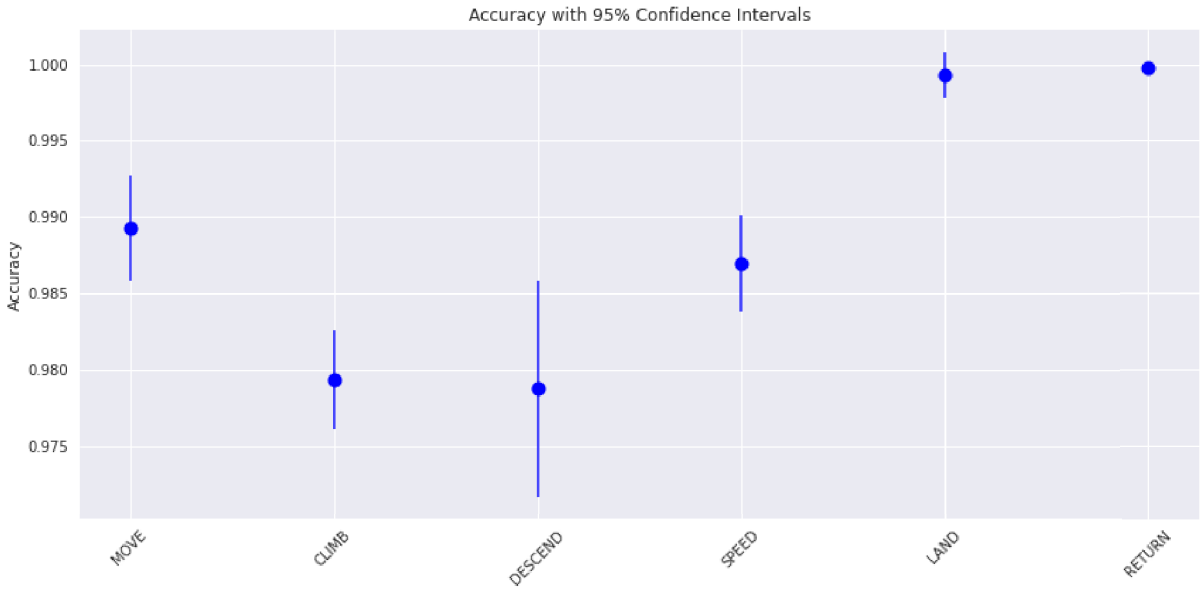


Figure 6.2: Accuracy Comparison with 95% confidence interval

6.4 Alternative Approach: Rule-Based Parsing

While transformer models demonstrate strong performance in natural language command parsing, it is important to consider alternative approaches. This section examines a rule-based parsing system using regular expressions (regex) as an alternative to the T5 transformer model.

6.4.1 Implementation of Rule-Based Parser

The rule-based parser was implemented using regular expressions to match command patterns of the training data generated for the previous experiment. Each command type (MOVE, CLIMB, DESCEND, etc.) has specific patterns that capture variations in syntax while maintaining the structured output format. For example, a MOVE command pattern is implemented as:

```
{
^(?P<verb>proceed to|head to|navigate to|move to|go to|\newline
redirect to|travel to|advance to|progress to|make way to)\s+(?P<location>\newline
ZONE_[A-Z])-(?P<sub_location>\d+)\s+heading\s+(?P<direction>\newline
NORTH|SOUTH|EAST|WEST|NORTHEAST|NORTHWEST|SOUTHEAST|SOUTHWEST|\newline
N|S|E|W|NE|NW|SE|SW)(?:\s+(?:immediately|quickly|now|as soon as possible))?$}
```

This approach provides deterministic behavior with guaranteed pattern matching, requires no training, and offers fast execution time. The implementation is straightforward to debug and maintain, as patterns can be directly modified without the need for model retraining. However, we need to create as many patterns as possible to ensure covering all variations.

6.4.2 Performance Analysis of Rule-Based Parser versus T5 Model

The comparative analysis of the rule-based parser and T5 model reveals striking differences in their handling of standard command patterns versus command variations. When tested on standard command patterns that closely match the training data, both approaches demonstrate exceptional performance, with the rule-based parser achieving perfect accuracy (100%) for CLIMB, DESCEND, LAND, and MOVE commands. The T5 model performs similarly well, maintaining accuracy above 97% across these command types, indicating strong capability in handling well-structured, familiar patterns.

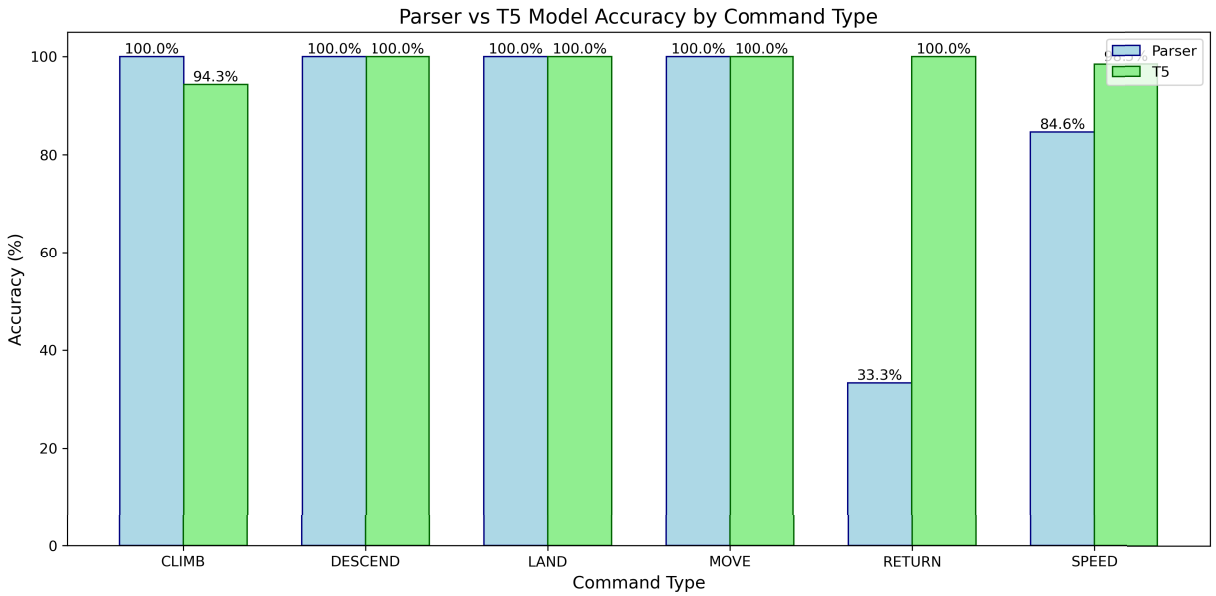


Figure 6.3: Parser vs T5 model performance on commands generated with training data logic

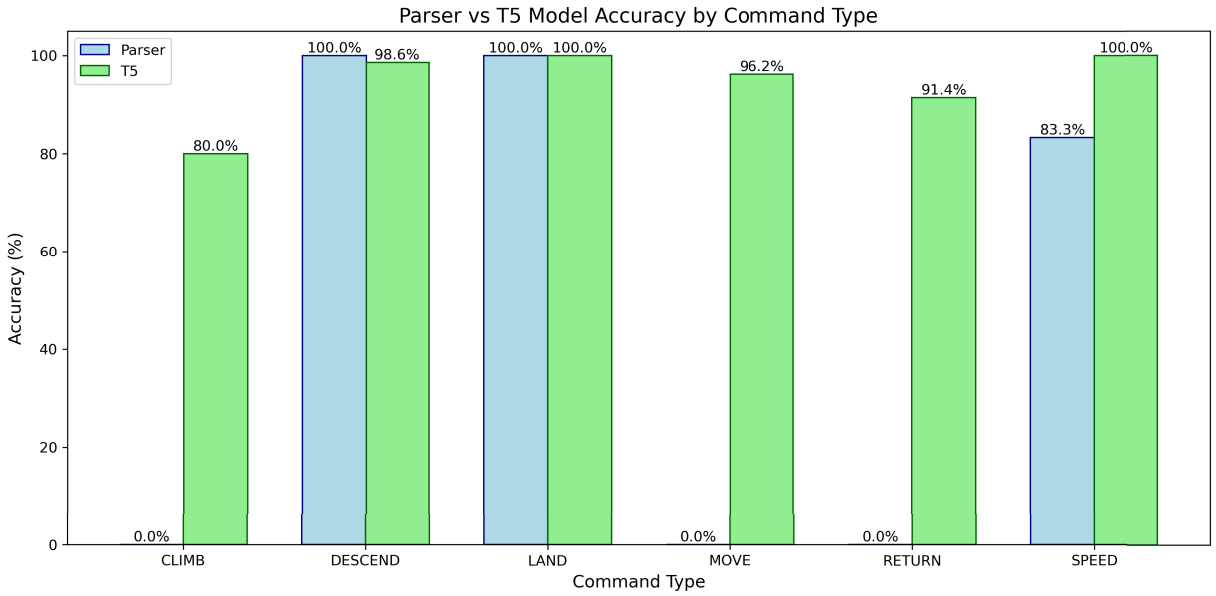


Figure 6.4: Parser vs T5 model performance on commands with new linguistic variations

Training Data Pattern Performance

In the training data pattern test scenario, the rule-based parser demonstrates its strength in exact pattern matching, achieving 100% accuracy for most command types, Fig. 6.3. The T5 model shows comparable performance, with only slight variations in accuracy: 98% for CLIMB, 100% for DESCEND and LAND, and 100% for MOVE commands. Even for more complex commands like RETURN and SPEED, both approaches maintain respectable performance levels, with the parser achieving 84.6% and the T5 model reaching 100% accuracy for SPEED commands.

New Variant Pattern Performance

The introduction of new command variations reveals a fundamental difference between the two approaches. For this experiment we kept LAND and DESCEND as control groups. The SPEED commands had significant overlap with the previous experiment due to limited variation possibilities in this command category. These commands are not using generalized commands. The rule-based parser's performance degrades dramatically when encountering variations not explicitly covered by its patterns, Fig. 6.4. Most notably, it completely fails (0% accuracy) for CLIMB, MOVE, and RETURN commands when presented with variations. This stark decline in performance highlights the parser's brittleness and its inability to generalize beyond explicitly defined patterns.

In contrast, the T5 model demonstrates remarkable resilience and generalization capability when handling command variations. It maintains high accuracy levels across all command types: 80% for CLIMB, 96.2% for MOVE, and 91.4% for RETURN commands. This robust performance on variant patterns suggests that the T5 model has learned underlying command structures rather than merely memorizing specific patterns.

Implications for Practical Applications

These results have significant implications for practical applications. The rule-based parser's perfect performance on standard patterns makes it an attractive option for systems with highly controlled, standardized command formats where variation is minimal or non-existent. However, its complete failure on variations makes it unsuitable for applications where command flexibility is required or where user input might deviate from exact patterns.

The T5 model's ability to maintain high accuracy across both the familiar vocabulary set and

variant patterns makes it a more versatile solution. Its performance demonstrates true generalization capability, suggesting it could handle unexpected variations in real-world applications while maintaining reliable accuracy. This adaptability comes at the cost of slightly lower accuracy on strictly standard patterns compared to the parser's perfect performance, but the trade-off favors the T5 model in any scenario where command variation is expected.

Performance Trade-offs

The comparison reveals a fundamental trade-off between perfect accuracy on known patterns versus generalization capability. The rule-based parser optimizes for the former, achieving perfect accuracy when patterns match exactly but failing completely with variations. The T5 model optimizes for generalization, maintaining strong performance across both familiar and new variant patterns while sometimes achieving perfect accuracy of 100%. This trade-off directly addresses the core challenge presented in drone operations: reconciling flexible command inputs with standardized formats.

Given the requirement of high accuracy, the rule-based parser could meet this threshold only for a strictly controlled set of command formats. However, this contradicts the real-world need for handling "multiple vendor-specific syntaxes" and "natural language variations from operators" as outlined in the problem statement. The T5 model's demonstrated ability to maintain high accuracy (91-100%) with the exception of CLIMB with 80% accuracy even with new variations makes it more suitable for addressing these operational requirements. However, additional work would be needed to reach the required reliability threshold. This aligns with DANTE's approach of using transformer-based processing to handle command format variations while maintaining reliable standardization.

6.5 Conclusion

This chapter has demonstrated how Open RAN's disaggregated architecture enables the expansion of network responsibilities beyond traditional packet routing to enhance QoE and QoS for URLLC applications through semantic communication. By leveraging the programmability of Near-RT RIC and the flexibility of xApps within the multi-vendor Open RAN framework, intelligence can be embedded closer to the radio access network while maintaining vendor neutrality. This approach is particularly valuable for private 5G deployments, where customized

network capabilities can directly address industry-specific requirements for ultra-reliable low-latency communications.

By integrating the semantic processing within the Open RAN infrastructure, DANTE delivers several key benefits: reduced processing overhead on URLLC endpoints, enhanced security through centralized validation, and consistent command interpretation across multiple devices. This integration is particularly crucial for complex scenarios like drone swarms, where command consistency and rapid processing are essential for coordinated operations.

DANTE addresses our fourth research question by demonstrating that relocating semantic processing to Near-RT RIC enhances reliability and efficiency without compromising latency requirements. This approach creates opportunities for expanded RAN capabilities in natural language understanding, multi-device coordination, and real-time command standardization, suggesting a future where RAN functions extend well beyond traditional networking to include advanced semantic processing.

Chapter 7

Conclusions and Future Work

7.1 Overview

This thesis has investigated the enhancement of Near-RT RIC components in Open RAN architecture to better support URLLC applications. The work was motivated by the observation that while Open RAN introduces dedicated intelligent controllers, several critical challenges remain unaddressed for URLLC support. Through systematic analysis and experimental validation, this thesis has demonstrated that classic RAN responsibilities can be enhanced, security vulnerabilities addressed, reliability methods redefined, and RAN responsibilities expanded to meet the stringent requirements of emerging URLLC applications.

The research has progressed through a logical sequence of enhancements: first establishing how control functions can be improved to support reliability, then addressing the security implications of these intelligent controllers, followed by redefining classical reliability methods, and finally expanding RAN's capabilities. Each stage has built upon the previous to create a comprehensive framework for URLLC support in Open RAN architectures. The novel frameworks developed — HORLA, AI xApp threat analysis, PULSE, and DANTE — collectively constitute a significant advancement in how Near-RT RIC can support the emerging wireless applications of the future.

7.2 Significant Results

Chapter 2: Open RAN and Near-RT RIC Research Opportunities

The second chapter provided a comprehensive literature review that established the fundamental understanding of Open RAN architecture and its evolution from previous RAN implementations. The review identified significant research gaps in intelligent controllers.

The chapter also analyzed the state of AI integration in telecommunications, classifying applications based on the Open RAN controllers' classification and identifying patterns in deployment scenarios not covered in current standards. This systematic analysis provided the foundation for the thesis's subsequent contributions by identifying key challenges in implementing AI-driven enhancements while maintaining the strict latency and reliability requirements of URLLC applications.

Chapter 3: Optimizing RAN's Reliability with Multi-Objective xApps

The third chapter introduced HORLA (HandOver Reinforcement Learning Application), a novel framework for multi-objective decision making in RAN control functions. The experimental validation demonstrated that HORLA outperforms traditional Maximum Received Power (MRP) approaches by reducing handover failures by over 40% while maintaining sub-second latency requirements. This significant improvement was achieved through a Pareto-optimal approach that simultaneously optimizes signal strength and reliability, rather than focusing on a single objective and handling failures through retry mechanisms.

The implementation of HORLA on the Near-RT RIC platform demonstrated that complex AI-driven decision making can operate within the strict timing constraints of URLLC applications. Beyond its primary objectives, HORLA demonstrated additional benefits, including a 57% reduction in energy consumption due to fewer failed handover attempts and enhanced security through validation of access points. These results conclusively proved that the reliability of classic RAN control functions can be significantly enhanced through multi-objective AI approaches while maintaining URLLC latency requirements.

Chapter 4: AI xApp and Security Vulnerabilities

The fourth chapter's objective was researching significant vulnerabilities in AI-driven xApps within Open RAN architectures, particularly for URLLC applications. Through systematic experimentation with software and hardware attack vectors, we exposed how adversaries can manipulate model parameters, reward functions, and resource utilization while evading detection by conventional monitoring systems. Our findings reveal critical gaps between current

O-RAN security specifications and the sophisticated threats targeting AI components. The attacks maintained apparent normal operation—preserving accuracy and latency metrics—while subtly degrading system performance and reliability. These results emphasize the urgent need for specialized security frameworks that address the unique characteristics of AI-driven network functions, including enhanced model validation techniques, continuous runtime monitoring, and comprehensive supply chain verification. As wireless networks continue evolving toward greater intelligence and autonomy, addressing these vulnerabilities becomes essential for maintaining the integrity and reliability demanded by next-generation applications.

Chapter 5: Enhancing RAN Reliability Solutions with Semantic-Intelligence xApps

The fifth chapter introduced PULSE (Predictive Ultra-reliable Low-latency System Engine), a transformer-based framework for enhancing packet recovery in URLLC applications. The experimental validation demonstrated exceptional performance compared to traditional approaches, achieving 100% prediction accuracy for up to 10% packet loss and 93.96% accuracy for 10-50% packet loss scenarios, with average processing times of 9.52ms per command. This performance significantly outperformed both RNN (averaging 0.05-0.07 Levenshtein similarity) and DNN approaches (0.14-0.24 similarity), with PULSE maintaining 0.825-1.0 similarity across all corruption levels.

The chapter extended this framework to multi-robot coordination, demonstrating how PULSE's optimal helper selection approach outperforms traditional nearest-neighbor, same-level, and random selection methods in battery-critical rescue scenarios. The comprehensive evaluation across 1,000 simulated scenarios showed that PULSE's approach effectively balances rescue time, final helper battery levels, distance traveled, and overall cost. These results proved that traditional reliability methods can be redefined through semantic-aware processing to meet the simultaneous reliability and latency requirements of URLLC applications.

Chapter 6: Expanding RAN Responsibilities with Semantic AI xApps

The sixth chapter introduced DANTE (Drone Adaptive Natural-to-Encoded Text Engine), a novel framework for moving semantic processing responsibilities from edge devices to the RAN infrastructure. The experimental validation demonstrated an overall accuracy of 98.90% in standardizing varied command formats while maintaining sub-millisecond processing times (average 0.0113ms, P95 0.017ms). When compared to rule-based alternatives, DANTE demonstrated superior flexibility and generalization capability, maintaining high accuracy even with command variations that caused rule-based systems to fail completely.

The comprehensive evaluation across different command types and variations proved that centralizing semantic processing within Near-RT RIC can enhance both reliability and security while reducing processing overhead on URLLC endpoints. This approach to expanding RAN responsibilities represents a fundamental shift in how communication processing is handled, moving beyond traditional packet routing to include sophisticated semantic understanding. The practical implementation within Open RAN architecture demonstrated the feasibility of this approach for real-world URLLC applications.

7.3 Further Work

This thesis has addressed significant challenges in enhancing Near-RT RIC for URLLC applications through improvements in control functions, security, reliability, and semantic processing. However, several important research directions remain to be explored.

7.3.1 Extending Multi-Objective Optimization to Other Control Functions

The multi-objective optimization framework introduced in HORLA demonstrates significant potential for application beyond handover control. Future research should explore extending this approach to other critical RAN control functions such as resource allocation, interference management, and beam management. This direction is particularly important as network densification increases the complexity of these control decisions. Research challenges include determining appropriate objective functions for each control domain, developing efficient learning algorithms that can operate within Near-RT RIC latency constraints, and creating unified frameworks that can address multiple control functions simultaneously. Methods combining reinforcement learning with multi-objective optimization could enable real-time adaptation to changing network conditions while maintaining multiple performance objectives. Successful implementation could significantly enhance network performance for URLLC applications by minimizing the need for retry mechanisms across multiple control domains.

7.3.2 Developing Automated Security Monitoring for AI Models

Building on findings from the security experiments, future work should focus on developing automated security monitoring systems that can detect subtle AI model compromises in real-

time without impacting latency. This research direction is critical as AI deployment in RAN control increases, creating new attack surfaces. Key challenges include developing lightweight anomaly detection algorithms that can operate within Near-RT RIC's latency constraints, creating benchmarks for normal behavior across different network conditions, and designing response mechanisms that maintain service continuity even under attack. Methods combining federated learning with transfer learning could enable security models that adapt to new attack patterns while preserving privacy and reducing communication overhead. Particular attention should be given to developing digital twin-based security testing environments that can proactively identify vulnerabilities before deployment. Successful implementation would strengthen Open RAN security while maintaining the flexibility and innovation benefits of multi-vendor ecosystems.

7.3.3 Integrating Semantic-Aware Reliability with Network Slicing

Future research should investigate the integration of PULSE's semantic-aware reliability enhancements with emerging technologies like network slicing and edge computing. This direction could enable customized reliability solutions for different application classes within a unified network infrastructure. Research challenges include developing mechanisms to differentiate between traffic types that benefit from semantic processing versus those requiring traditional reliability methods, creating context-aware predictive models that can operate across slices, and designing efficient resource allocation algorithms that balance reliability enhancement with other network objectives. Methods combining transformer models with reinforcement learning could enable dynamic adaptation of prediction mechanisms based on application requirements and network conditions. Successful integration would significantly enhance reliability for heterogeneous application mixes while efficiently utilizing network resources, particularly important for beyond-5G network architectures supporting diverse URLLC applications.

7.3.4 Scaling Semantic Processing for Heterogeneous Networks

Building on DANTE's proven concept, future research should explore the scalability and performance characteristics of semantic processing approaches in large-scale, heterogeneous network deployments. This direction is essential as networks increasingly support diverse devices with varying communication capabilities and requirements. Research challenges include developing efficient distributed semantic processing frameworks across different use cases that maintain consistent interpretation across the network, creating compression methods for semantic mod-

els to reduce deployment overhead, and designing adaptive processing mechanisms that allocate computational resources based on semantic complexity. Particular attention should be given to cross-domain semantic understanding, enabling consistent interpretation across different application domains. Successful implementation would significantly enhance the flexibility and efficiency of URLLC applications in heterogeneous environments.

As wireless networks continue to evolve towards 6G and beyond, the intersection of AI, Open RAN, innovative and advanced AI algorithms and URLLC will only grow in importance. The frameworks and insights provided by this thesis lay a foundation for future research aimed at transforming the RAN into an intelligent, resilient, and context-aware enabler for the most demanding wireless applications of tomorrow.

Bibliography

[1] O-RAN Alliance, "O-ran.wg1.ts.use-cases-detailed-specification-r004-v16.00: Technical specification," O-RAN Work Group 1 (Use Cases and Overall Architecture), Technical Specification, 2023, Use Cases Detailed Specification.

[2] Wikipedia. "Radio access network." (), [Online]. Available: https://en.wikipedia.org/wiki/Radio_Access_Network.

[3] H. Asplund, D. Astely, P. Butovitsch, *et al.*, "Chapter 12 - architecture and implementation aspects," in *Advanced Antenna Systems for 5G Network Deployments*, H. Asplund, D. Astely, P. Butovitsch, *et al.*, Eds., Academic Press, 2020, pp. 527–559, ISBN: 978-0-12-820046-9. DOI: https://doi.org/10.1016/B978-0-12-820046-9.00012-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128200469000125.

[4] J. Kempf and P. Yegani, "Openran: A new architecture for mobile wireless internet radio access networks," 5, vol. 40, 2002, pp. 118–123. DOI: 10.1109/35.1000222.

[5] B.-S. P. LinI, "Toward an ai-enabled o-ran-based and sdn/nfv-driven 5g& iot network era," Network, Communication Technologies, Canadian Center of Science, and Education, Jun. 2021. DOI: ttps://doi.org/10.5539/nct.v6n1p6.

[6] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead," vol. 182, 2020, p. 107 516. DOI: https://doi.org/10.1016/j.comnet.2020.107516. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620311786.

[7] 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 Management (IM)*, 2021, pp. 402–409.

[8] B.-S. Lin, "Toward an ai-enabled o-ran-based and sdn/nfv-driven 5g& iot network era," vol. 6, Jun. 2021, p. 6. DOI: 10.5539/nct.v6n1p6.

[9] N. Kazemifard and V. Shah-Mansouri, "Minimum delay function placement and resource allocation for open ran (o-ran) 5g networks," vol. 188, 2021, p. 107 809. DOI: https://doi.org/10.1016/j.comnet.2021.107809. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000037.

[10] L. Giupponi and F. Wilhelmi, "Blockchain-enabled network sharing for o-ran," Arxiv.org, Jul. 2021.

[11] L. Bertizzolot, T. X. Tran, J. Buczek, *et al.*, "Streaming from the air: Enabling high data-rate 5g cellular links for drone streaming applications," arxiv, Apr. 2021.

[12] B. Brik, K. Boutiba, and A. Ksentini, "Deep learning for b5g open radio access network: Evolution, survey, case studies, and challenges," vol. 3, 2022, pp. 228–250. DOI: 10.1109/OJCOMS.2022.3146618.

[13] S. K. Singh, R. Singh, and B. Kumbhani, "The evolution of radio access network towards open-ran: Challenges and opportunities," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6. DOI: 10.1109/WCNCW48565.2020.9124820.

[14] M. Koziol, "The clash over 5g's first mile: The wireless industry is divided on open ran's goal to make network components interoperable," 5, vol. 58, 2021, pp. 40–46. DOI: 10.1109/MSPEC.2021.9423816.

[15] P. Enrique Iturria Rivera, S. Mollahasani, and M. Erol-Kantarci, "Multi-agent team learning in virtualized open radio access networks (o-ran)," Feb. 2021.

[16] B. Balasubramanian, E. S. Daniels, M. Hiltunen, *et al.*, "Ric: A ran intelligent controller platform for ai-enabled cellular networks," 2, vol. 25, 2021, pp. 7–17. DOI: 10.1109/MIC.2021.3062487.

[17] 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," 2022.

[18] J. Breen, A. Buffmire, J. Duerig, *et al.*, "Powder: Platform for open wireless data-driven experimental research," vol. 197, 2021, p. 108 281. DOI: https://doi.org/10.1016/j.comnet.2021.108281. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621003017.

[19] "O-ran alliance." (), [Online]. Available: https://www.o-ran.org/.

[20] "O-ran software community." [Online]. Available: https://wiki.o-ran-sc.org/.

[21] R. Schmidt, M. Irazabal, and N. Nikaein, "Flexric: An sdk for next-generation sd-rans," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 411–425, ISBN: 9781450390989. [Online]. Available: https://doi.org/10.1145/3485983.3494870.

[22] "Tip project for open ran." [Online]. Available: https://telecominfraproject.com/openran/.

[23] "O-RAN architecture description v03.00," O-RAN Alliance, 2020.

[24] "Oran use cases and deployment," ORAN-Alliance. [Online]. Available: https://static1.squarespace.com/static/5ad774cce74940d7115044b0/t/5e95a0a306c6ab2d1cbca4d31586864301196/O-RAN+Use+Cases+and+Deployment+Scenarios+Whitepaper+February+2020.pdf.

[25] "AI/ML workflow description and requirements,o-ran.wg2.aiml-v01.03," O-RAN Alliance, 2021.

[26] R. Ferrús, O. Sallent, J. Pérez-Romero, and R. Agustí, "Applicability domains of machine learning in next generation radio access networks," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 1066–1073. DOI: 10.1109/CSCI49370.2019.00203.

[27] I. El Khayat, P. Geurts, and G. Leduc, "Enhancement of TCP over wired/wireless networks with packet loss classifiers inferred by supervised learning," 2, vol. 16, Springer, 2010, pp. 273–290. DOI: 10.1007/s11276-008-0129-y.

[28] B. Alotaibi and K. Elleithy, "Rogue access point detection: Taxonomy, challenges, and future directions," 3, vol. 90, Springer, 2016, pp. 1261–1290. DOI: 10.1007/s11277-016-3390-x.

[29] C. Benzaıd̈, A. Boulgheraif, F. Z. Dahmane, A. Al-Nemrat, and K. Zeraoulia, "Intelligent detection of mac spoofing attack in 802.11 network," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ser. ICDCN '16, Singapore, Singapore: Association for Computing Machinery, 2016, ISBN: 9781450340328. DOI: 10.1145/2833312.2850446. [Online]. Available: https://doi.org/10.1145/2833312.2850446.

[30] R. Latha and R. Bommi, "Detection of deauthentication threats in wi-fi channels using machine learning strategies," in *2022 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, IEEE, vol. 1, 2022, pp. 1–6.

[31] Y. Wang, S. Jere, S. Banerjee, L. Liu, S. Shetty, and S. Dayekh, "Anonymous jamming detection in 5g with bayesian network model based inference analysis," in *2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR)*, IEEE, 2022, pp. 151–156.

[32] M. Choubisa, R. Doshi, N. Khatri, and K. K. Hiran, "A simple and robust approach of random forest for intrusion detection system in cyber security," in *2022 International conference on IoT and blockchain technology (ICIBT)*, IEEE, 2022, pp. 1–5.

[33] R. Geetha and T. Thilagam, "A review on the effectiveness of machine learning and deep learning algorithms for cyber security," 4, vol. 28, Springer, 2021, pp. 2861–2879.

[34] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "An improved anomaly detection model for iot security using decision tree and gradient boosting," 3, vol. 79, Springer, 2023, pp. 3392–3411.

[35] D. W. Browne, M. W. Browne, and M. P. Fitz, "Cth07-4: Singular value decomposition of correlated mimo channels," in *IEEE Globecom 2006*, 2006, pp. 1–6. DOI: 10.1109/GLOCOM.2006.73.

[36] A. Rooshenas, H. R. Rabiee, A. Movaghar, and M. Y. Naderi, "Reducing the data transmission in wireless sensor networks using the principal component analysis," in *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, IEEE, 2010, pp. 133–138.

[37] P. Somwang and W. Lilakiatsakun, "Computer network security based on support vector machine approach," in *2011 11th International Conference on Control, Automation and Systems*, 2011, pp. 155–160.

[38] T. Zhi, Y. Liu, J. Wang, and H. Zhang, "Resist interest flooding attacks via entropy–svm and jensen–shannon divergence in information-centric networking," 2, vol. 14, 2020, pp. 1776–1787. DOI: 10.1109/JSYST.2019.2939371.

[39] "An svm based ddos attack detection method for ryu sdn controller," CoNEXT '19 Companion: Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies, Dec. 2019, pp. 72–73. DOI: https://doi.org/10.1145/3360468.3368183.

[40] R. Guerra-Gómez, S. R. Boqué, M. García-Lozano, and J. O. Bonafé, "Machine-learning based traffic forecasting for resource management in c-ran," in *2020 European Conference on Networks and Communications (EuCNC)*, 2020, pp. 200–204. DOI: 10.1109/EuCNC48522.2020.9200958.

[41] P. Ghosh and R. Mitra, "Proposed ga-bfss and logistic regression based intrusion detection system," in *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, 2015, pp. 1–6. DOI: 10.1109/C3IT.2015.7060117.

[42] S. Yadav and S. Selvakumar, "Detection of application layer ddos attack by modeling user behavior using logistic regression," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, 2015, pp. 1–6. DOI: 10.1109/ICRITO.2015.7359289.

[43] R. A. Shah, Y. Qian, D. Kumar, M. Ali, and M. B. Alvi, "Network intrusion detection through discriminative feature selection by using sparse logistic regression," 4, vol. 9, 2017. [Online]. Available: https://www.mdpi.com/1999-5903/9/4/81.

[44] N. Ben-Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs decision trees in intrusion detection systems," SAC '04: Proceedings of the 2004 ACM symposium on Applied computing, Mar. 2004, pp. 420–424. DOI: https://doi.org/10.1145/967900.967989.

[45] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden naïve bayes multiclass classifier," 18, vol. 39, 2012, pp. 13 492–13 500. DOI: https://doi.org/10.1016/j.eswa.2012.07.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417412008640.

[46] K. Reddy and P. Thilagam, "Naïve bayes classifier to mitigate the ddos attacks severity in ad-hoc networks," vol. 12, Aug. 2020, pp. 221–226.

[47] D. Liu, Z. Liu, and Z. Song, "Lda-based csi amplitude fingerprinting for device-free localization," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 2020–2023. DOI: 10.1109/CCDC49329.2020.9164348.

[48] J. Jiang, X. Zhu, G. Han, M. Guizani, and L. Shu, "A dynamic trust evaluation and update mechanism based on c4.5 decision tree in underwater wireless sensor networks," 8, vol. 69, 2020, pp. 9031–9040. DOI: 10.1109/TVT.2020.2999566.

[49] J. Wang, Q. Yang, and D. Ren, "An intrusion detection algorithm based on decision tree technology," in *2009 Asia-Pacific Conference on Information Processing*, vol. 2, 2009, pp. 333–335. DOI: 10.1109/APCIP.2009.218.

[50] M. H. Mazhar and Z. Shafiq, "Real-time video quality of experience monitoring for https and quic," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1331–1339. DOI: 10.1109/INFOCOM.2018.8486321.

[51] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, "An improved k-nearest-neighbor indoor localization method based on spearman distance," 3, vol. 23, 2016, pp. 351–355. DOI: 10.1109/LSP.2016.2519607.

[52] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," 3, vol. 21, 2019, pp. 2224–2287. DOI: 10.1109/COMST.2019.2904897.

[53] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," 4, vol. 20, 2018, pp. 2595–2621. DOI: 10.1109/COMST.2018.2846401.

[54] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," in *2017 International Conference on Data and Software Engineering (ICoDSE)*, 2017, pp. 1–6. DOI: 10.1109/ICODSE.2017.8285847.

[55] S.-H. Choi, D.-H. Hwang, and Y.-H. Choi, "Wireless intrusion prevention system using dynamic random forest against wireless mac spoofing attack," in *2017 IEEE Conference on Dependable and Secure Computing*, 2017, pp. 131–137. DOI: 10.1109/DESEC.2017.8073804.

[56] "K-means." [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering.

[57] E. Balevi and R. D. Gitlin, "A clustering algorithm that maximizes throughput in 5g heterogeneous f-ran networks," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422151.

[58] H. Harb, A. Makhoul, D. Laiymani, A. Jaber, and R. Tawil, "K-means based clustering approach for data aggregation in periodic sensor networks," in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2014, pp. 434–441. DOI: 10.1109/WiMOB.2014.6962207.

[59] G. Wang, Y. Zhao, J. Huang, Q. Duan, and J. Li, "A k-means-based network partition algorithm for controller placement in software defined network," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6. DOI: 10.1109/ICC.2016.7511441.

[60] L. Tong-yan and L. Xing-ming, "The study of alarm association rules mining in telecommunication networks," in *2008 International Conference on Communications, Circuits and Systems*, 2008, pp. 1030–1034. DOI: 10.1109/ICCCAS.2008.4657944.

[61] C. Li and X. Huang, "Research on fp-growth algorithm for massive telecommunication network alarm data based on spark," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2016, pp. 875–879. DOI: 10.1109/ICSESS.2016.7883205.

[62] Q. Wang, K. Feng, X. Li, and S. Jin, "Precodernet: Hybrid beamforming for millimeter wave systems with deep reinforcement learning," 10, vol. 9, 2020, pp. 1677–1681. DOI: 10.1109/LWC.2020.3001121.

[63] H. Vaezy, M. Salehi Heydar Abad, O. Ercetin, H. Yanikomeroglu, M. J. Omidi, and M. M. Naghsh, "Beamforming for maximal coverage in mmwave drones: A reinforcement learning approach," 5, vol. 24, 2020, pp. 1033–1037. DOI: 10.1109/LCOMM.2020.2974958.

[64] C.-H. Zhong, K. Guo, and M. Zhao, "Online sparse beamforming in c-ran: A deep reinforcement learning approach," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6. DOI: 10.1109/WCNC49053.2021.9417394.

[65] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," 11, vol. 69, 2020, pp. 13 124–13 138. DOI: 10.1109/TVT.2020.3020400.

[66] T. M. Ho and K.-K. Nguyen, "Joint server selection, cooperative offloading and handover in multi-access edge computing wireless network: A deep reinforcement learning approach," 2020, pp. 1–1. DOI: 10.1109/TMC.2020.3043736.

[67] M. Sana, A. De Domenico, E. C. Strinati, and A. Clemente, "Multi-agent deep reinforcement learning for distributed handover management in dense mmwave networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8976–8980. DOI: 10.1109/ICASSP40776.2020.9052936.

[68] M. Mohsenivatani, M. Darabi, S. Parsaeefard, M. Ardebilipour, and B. Maham, "Throughput maximization in c-ran enabled virtualized wireless networks via multi-agent deep reinforcement learning," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–6. DOI: 10.1109/PIMRC48278.2020.9217287.

[69] W.-C. Chien, C.-F. Lai, and H.-C. Chao, "Dynamic resource prediction and allocation in c-ran with edge artificial intelligence," 7, vol. 15, 2019, pp. 4306–4314. DOI: 10.1109/TII.2019.2913169.

[70] "Intelligent 5g l2 mac scheduler, powered by capgemini netanticipate 5g on intel architecture," Intel. [Online]. Available: https://builders.intel.com/docs/networkbuilders/intelligent-5g-l2-mac-scheduler-powered-by-capgemini-netanticipate-5g-on-intel-architecture-v13.pdf.

[71] H. Saki, N. Khan, M. G. Martini, and M. M. Nasralla, "Machine learning based frame classification for videos transmitted over mobile networks," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–6. DOI: 10.1109/CAMAD.2019.8858448.

[72] "Vehicle to evrything." [Online]. Available: https://en.wikipedia.org/wiki/Vehicle-to-everything#802.11p_(DSRC).

[73] M. Jiang, "Device-controlled traffic steering in mobile networks," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, 2015, pp. 7–12. DOI: 10.1109/NGMAST.2015.13.

[74] K. Adachi, M. Li, P. H. Tan, Y. Zhou, and S. Sun, "Q-learning based intelligent traffic steering in heterogeneous network," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–5. DOI: 10.1109/VTCSpring.2016.7504436.

[75] S. Barmpounakis, A. Kaloxylos, P. Spapis, C. Zhou, P. Magdalinos, and N. Alonistioti, "Data analytics for 5g networks: A complete framework for network access selection and traffic steering," Nov. 2018.

[76] F. D. Priscoli, A. Giuseppi, F. Liberati, and A. Pietrabissa, "Traffic steering and network selection in 5g networks based on reinforcement learning," in *2020 European Control Conference (ECC)*, 2020, pp. 595–601. DOI: 10.23919/ECC51009.2020.9143837.

[77] A. Rostami, P. Ohlen, K. Wang, *et al.*, "Orchestration of ran and transport networks for 5g: An sdn approach," 4, vol. 55, 2017, pp. 64–70. DOI: 10.1109/MCOM.2017.1600119.

[78] S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, and R. Langar, "5g ran: Functional split orchestration optimization," 7, vol. 38, 2020, pp. 1448–1463. DOI: 10.1109/JSAC.2020.2999685.

[79] "Configuring storm control," in *Catalyst 4500 Series Switch Software Configuration Guide, 12.2(53)SG*, Cisco. [Online]. Available: https://www.cisco.com/c/en/us/td/

docs/switches/lan/catalyst4500/12-2/53SG/configuration/config/bcastsup.html.

[80] L. V. Le, D. Sinh, L.-P. Tung, and B.-S. Lin, "Enhanced handover clustering and forecasting models based on machine learning and big data," vol. 6, Oct. 2018. DOI: 10.14738/tmlai.65.5411.

[81] L. Hao and B. Ng, "Self-healing solutions for wi-fi networks to provide seamless handover," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 639–642.

[82] S. Memon and M. Maheswaran, "Using machine learning for handover optimization in vehicular fog computing," SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. DOI: https://doi.org/10.1145/3297280.3297300.

[83] B. Hussain, Q. Du, and P. Ren, "Semi-supervised learning based big data-driven anomaly detection in mobile wireless networks," 4, vol. 15, 2018, pp. 41–57. DOI: 10.1109/CC.2018.8357700.

[84] Q. Liu, Kwong, C.F., and S. Zhang, "A fuzzy-clustering based approach for madm handover in 5g ultra-dense networks," Springer-Wireless Network, Sep. 2019. DOI: https://doi.org/10.1007/s11276-019-02130-3.

[85] Z. Kaleem, M. Z. Khaliq, A. Khan, I. Ahmad, and T. Q. Duong, "Ps-cara: Context-aware resource allocation scheme for mobile public safety networks," 5, vol. 18, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/5/1473.

[86] H. Nishiyama, Y. Kawamoto, and D. Takaishi, "On ofdm-based resource allocation in lte radio management system for unmanned aerial vehicles (uavs)," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5. DOI: 10.1109/VTCFall.2017.8288175.

[87] M. L. Marí-Altozano, S. Luna-Ramírez, M. Toril, and C. Gijón, "A qoe-driven traffic steering algorithm for lte networks," 11, vol. 68, 2019, pp. 11 271–11 282. DOI: 10.1109/TVT.2019.2941237.

[88] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive mimo: Benefits and challenges," 5, vol. 8, 2014, pp. 742–758. DOI: 10.1109/JSTSP.2014.2317671.

[89] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," 2, vol. 52, 2014, pp. 186–195. DOI: 10.1109/MCOM.2014.6736761.

[90] S. Nie, J. M. Jornet, and I. F. Akyildiz, "Intelligent environments based on ultra-massive mimo platforms for wireless communication in millimeter wave and terahertz bands," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7849–7853. DOI: 10.1109/ICASSP.2019.8683394.

[91] D. B. et al., ""toward the network of the future: From enabling technologies to 5g concepts," trans. emerging tele-commun. technologies," 8, vol. 28, Aug. 2017, e3205.

[92] "From net-work sharing to multi-tenancy: The 5g network slice broker," 7, vol. 54, IEEE Commun. Mag., Jul. 2016, pp. 32–39.

[93] *3gpp ts 28.530*. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273.

[94] W. Jiang, S. D. Anton, and H. Dieter Schotten, "Intelligence slicing: A unified framework to integrate artificial intelligence into 5g networks," in *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2019, pp. 227–232. DOI: 10.23919/WMNC.2019.8881402.

[95] J. Mei, X. Wang, and K. Zheng, "An intelligent self-sustained ran slicing framework for diverse service provisioning in 5g-beyond and 6g networks," 3, vol. 1, 2020, pp. 281–294. DOI: 10.23919/ICN.2020.0019.

[96] E. Esenogho, K. Djouani, and A. M. Kurien, "Integrating artificial intelligence internet of things and 5g for next-generation smartgrid: A survey of trends challenges and prospect," vol. 10, 2022, pp. 4794–4831. DOI: 10.1109/ACCESS.2022.3140595.

[97] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6g era: Challenges and opportunities," 2, vol. 35, 2021, pp. 244–251. DOI: 10.1109/MNET.011.2000493.

[98] C. Li, Q. Luo, G. Mao, M. Sheng, and J. Li, "Vehicle-mounted base station for connected and autonomous vehicles: Opportunities and challenges," 4, vol. 26, 2019, pp. 30–36. DOI: 10.1109/MWC.2019.1800541.

[99] "5g open ran ecosystem whitepaper," NTT Docomo, Jun. 2021. [Online]. Available: https://www.nttdocomo.co.jp/english/binary/pdf/corporate/technology/whitepaper_5g_open_ran/OREC_WP.pdf.

[100] N. P. Narekar and D. M. Bhalerao, "A survey on obstacles for 5g communication," in *2015 International Conference on Communications and Signal Processing (ICCSP)*, 2015, pp. 0831–0835. DOI: 10.1109/ICCSP.2015.7322610.

[101] M. Ghanim, M. Alhilali, J. Din, and H. Y. Lam, "Rain attenuation statistics over 5g millimetre wave links in malaysia," in *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2018, pp. 266–269. DOI: 10.1109/EECSI.2018.8752836.

[102] K. Y., *5G Mobile Communication Systems: Fundamentals, Challenges, and Key Technologies*. Springer, Singapore, 2019. DOI: doi.org/10.1007/978-981-13-1768-2_10.

[103] E. Coronado, R. Behravesh, T. Subramanya, *et al.*, "Zero touch management: A survey of network automation solutions for 5g and 6g networks," 4, vol. 24, 2022, pp. 2535–2578. DOI: 10.1109/COMST.2022.3212586.

[104] "3gpp, self-organizing network." [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/105-son.

[105] S. Fuhrmann, Y. Kogan, and R. Milito, "An adaptive autonomous network congestion controller," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 1, 1996, 301–306 vol.1. DOI: 10.1109/CDC.1996.574321.

[106] Y. Chun, L. Qin, L. Dongsheng, and S. MeiLin, "Qos routing in self-organized network," in *WCC 2000 - ICCT 2000. 2000 International Conference on Communication Technology Proceedings (Cat. No.00EX420)*, vol. 2, 2000, 1304–1309 vol.2. DOI: 10.1109/ICCT.2000.890906.

[107] "Autonomous networks, supporting tomorrow's ict business. 1st edition,isbn no. 979-10-92620-37-6," Oct. 2020.

[108] "Cheat sheet: What is digital twin?," IBM Business Operation Blog. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/.

[109] A. E. Azzaoui, S. K. Singh, Y. Pan, and J. H. Park, "Block5gintell: Blockchain for ai-enabled 5g networks," vol. 8, 2020, pp. 145 918–145 935. DOI: 10.1109/ACCESS.2020.3014356.

[110] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," vol. 5, 2017, pp. 11 255–11 268. DOI: 10.1109/ACCESS.2017.2710056.

[111] K. Awahara, S. Izumi, T. Abe, and T. Suganuma, "Autonomous control method using ai planning for energy-efficient network systems," in *2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications*, 2013, pp. 628–633. DOI: 10.1109/BWCCA.2013.111.

[112] R. Li, Z. Zhao, X. Zhou, *et al.*, "Intelligent 5g: When cellular networks meet artificial intelligence," 5, vol. 24, 2017, pp. 175–183. DOI: 10.1109/MWC.2017.1600304WC.

[113] "5g;nr; radio resource control (rrc); protocol specification," 3GPP Organization, Technical specification, May 2022.

[114] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[115] M. V., K. K., and D. Silver, "Multi-objective reinforcement learning using sets of pareto dominating policies," 2015. DOI: https://doi.org/10.1038/nature14236.

[116] B. Khodapanah, T. Hößler, B. Yuncu, A. N. Barreto, M. Simsek, and G. Fettweis, "Coexistence management for urllc in campus networks via deep reinforcement learning," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6. DOI: 10.1109/WCNC45663.2020.9120498.

[117] G. Pollini, "Trends in handover design," 3, vol. 34, 1996, pp. 82–90. DOI: 10.1109/35.486807.

[118] E. Del Re, R. Fantacci, and G. Giambene, "Handover and dynamic channel allocation techniques in mobile cellular networks," 2, vol. 44, 1995, pp. 229–237. DOI: 10.1109/25.385913.

[119] S. Tekinay and B. Jabbari, "Handover and channel assignment in mobile cellular networks," 11, vol. 29, 1991, pp. 42–46. DOI: 10.1109/35.109664.

[120] R. Beck and H. Panzer, "Strategies for handover and dynamic channel allocation in micro-cellular mobile radio systems," in *IEEE 39th Vehicular Technology Conference*, 1989, 178–185 vol.1. DOI: 10.1109/VETEC.1989.40070.

[121] G. Falciasecca, M. Frullone, G. Riva, and A. Serra, "Comparison of different handover strategies for high capacity cellular mobile radio systems," in *IEEE 39th Vehicular Technology Conference*, 1989, 122–127 vol.1. DOI: 10.1109/VETEC.1989.40060.

[122] R. Zhang, G. Mahardhika, M. Ismail, and R. Nordin, "Vertical handover decision algorithm using multicriteria metrics in heterogeneous wireless network," Hindawi Publishing Corporation, 2015. DOI: https://doi.org/10.1155/2015/539750.

[123] "3gpp ts 23.009 v17.0.0, 3rd generation partnership project; technical specification group core network and terminals; handover procedures," 3GPP Organization, Technical specification, Mar. 2022.

[124] L. L. Vy, L.-P. Tung, and B.-S. P. Lin, "Big data and machine learning driven handover management and forecasting," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2017, pp. 214–219. DOI: 10.1109/CSCN.2017.8088624.

[125] R. Zhohov, A. Palaios, H. Rydén, R. Moosavi, and J. Berglund, "Reducing latency: Improving handover procedure using machine learning," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5. DOI: 10.1109/VTC2021-Spring51267.2021.9448875.

[126] M.-T. Nguyen and S. Kwon, "Machine learning–based mobility robustness optimization under dynamic cellular networks," vol. 9, 2021, pp. 77 830–77 844. DOI: 10.1109/ACCESS.2021.3083554.

[127] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, "Machine learning for wireless connectivity and security of cellular-connected uavs," 1, vol. 26, 2019, pp. 28–35. DOI: 10.1109/MWC.2018.1800155.

[128] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Second. Prentice Hall, 2002.

[129] P. I. Lazaridis, D. H. Ho, and T. A. Gulliver, "Outage probability and normalized sinr-based power allocation over rician fading channels," vol. 2020, Hindawi, 2020. DOI: 10.1155/2020/8818579. [Online]. Available: https://doi.org/10.1155/2020/8818579.

[130] J. Wang, W. Mu, Y. Liu, L. Guo, S. Zhang, and G. Gui, "Deep reinforcement learning-based satellite handover scheme for satellite communications," in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2021, pp. 1–6. DOI: 10.1109/WCSP52459.2021.9613411.

[131] Y. Jang, S. M. Raza, H. Choo, and M. Kim, "Uavs handover decision using deep reinforcement learning," in *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2022, pp. 1–4. DOI: 10.1109/IMCOM53663.2022.9721627.

[132] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," 6, vol. 5, 2018, pp. 4296–4307. DOI: 10.1109/JIOT.2018.2848295.

[133] V. Yajnanarayana, H. Rydén, and L. Hévizi, "5g handover using reinforcement learning," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 349–354. DOI: 10.1109/5GWF49715.2020.9221072.

[134] Y. Koda, K. Nakashima, K. Yamamoto, T. Nishio, and M. Morikura, "Handover management for mmwave networks with proactive performance prediction using camera images and deep reinforcement learning," 2, vol. 6, 2020, pp. 802–816. DOI: 10.1109/TCCN.2019.2961655.

[135] "Report itu-r m.2135-1-guidelines for evaluation of radio interface technologies for imt-advanced."

[136] "Study on 5g security enhancements against false base stations (fbs)," 3GPP Organization, Technical specification, 2018.

[137] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?," New York, NY, USA, 2006. DOI: 10.1145/1128817.1128824. [Online]. Available: https://doi.org/10.1145/1128817.1128824.

[138] K. Kurita, P. Michel, and G. Neubig, "Weight poisoning attacks on pre-trained models," 2020.

[139] M. A. Ramirez, S.-K. Kim, H. A. Hamadi, *et al.*, "Poisoning attacks and defenses on artificial intelligence: A survey," 2022.

[140] Y. Lu, G. Kamath, and Y. Yu, "Exploring the limits of model-targeted indiscriminate data poisoning attacks," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 23–29 Jul 2023, pp. 22 856–22 879. [Online]. Available: https://proceedings.mlr.press/v202/lu23e.html.

[141] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, and T. Goldstein, "Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 9389–9398. [Online]. Available: https://proceedings.mlr.press/v139/schwarzschild21a.html.

[142] M. Huai, J. Sun, R. Cai, L. Yao, and A. Zhang, "Malicious attacks against deep reinforcement learning interpretations," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 472–482.

[143] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-Robust federated learning," in *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, Aug. 2020, pp. 1605–1622, ISBN: 978-1-939133-17-5. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/fang.

[144] X. Li, N. Wang, S. Yuan, and Z. Guan, "Fedimp: Parameter importance-based model poisoning attack against federated learning system," vol. 144, 2024, p. 103 936. DOI: https://doi.org/10.1016/j.cose.2024.103936. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824002414.

[145] X. Zhang and G. Sun, "Toss-up wear leveling: Protecting phase-change memories from inconsistent write patterns," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 1–6.

[146] S. Kanjalkar, J. Kuo, Y. Li, and A. Miller, "Short paper: I can't believe it's not stake! resource exhaustion attacks on pos," in *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*, Springer, 2019, pp. 62–69.

[147] R. Pietrantuono, M. Ficco, and F. Palmieri, "Testing the resilience of mec-based iot applications against resource exhaustion attacks," 2, vol. 21, IEEE, 2023, pp. 804–818.

[148] 3. 5. Security. [Online]. Available: https://www.3gpp.org/news-events/1975-sec_5g.

[149] "O-ran.sfg.security-protocols-specifications-v03.00," Rev. V3.00, O-RAN Alliance, Apr. 2021.

[150] "O-ran security threat modeling and remediation analysis," Rev. v02.01, O-RAN Alliance, Oct. 2021.

[151] A. Arnaz, J. Lipman, M. Abolhasan, and M. Hiltunen, "Toward integrating intelligence and programmability in open radio access networks: A comprehensive survey," vol. 10, 2022, pp. 67 747–67 770. DOI: 10.1109/ACCESS.2022.3183989.

[152] O.-R. W. G. 1. (W. Group), "Study on security for near real time ric and xapps," O-RAN Alliance, Technical Report O-RAN.WG11.Security-Near-RT-RIC-xApps-TR.0-R003-v05.00, 2025. [Online]. Available: https://www.o-ran.org/.

[153] Z. Zhang, X. Xia, C. Huang, Y. Xiao, and L. Xie, "Bs-plcnet 2: Two-stage band-split packet loss concealment network with intra-model knowledge distillation," arXiv:2406.05961v1 [eess.AS], Jun. 2024. [Online]. Available: https://arxiv.org/abs/2406.05961v1.

[154] S. Itahara, T. Nishio, and K. Yamamoto, "Packet-loss-tolerant split inference for delay-sensitive deep learning in lossy wireless networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6. DOI: 10.1109/GLOBECOM46510.2021.9685179.

[155] R. Chauhan and S. Kumar, "Packet loss prediction using artificial intelligence unified with big data analytics, internet of things and cloud computing technologies," in *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, 2021, pp. 01–06. DOI: 10.1109/ISCON52037.2021.9702517.

[156] Y. Luo, X. Chen, N. Ge, W. Feng, and J. Lu, "Transformer-based device-type identification in heterogeneous iot traffic," 6, vol. 10, 2023, pp. 5050–5062. DOI: 10.1109/JIOT.2022.3221967.

[157] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," 140, vol. 21, 2020, pp. 1–67.

[158] J. Luo, Z. Chen, W. Chen, *et al.*, "A study on the application of the t5 large language model in encrypted traffic classification," vol. 18, Springer, Feb. 2025, pp. 1–13. DOI: 10.1007/s12083-024-01817-5.

[159] G. Ding, S. Liu, J. Yuan, and G. Yu, "Joint urllc traffic scheduling and resource allocation for semantic communication systems," 7, vol. 23, 2024, pp. 7278–7290. DOI: 10.1109/TWC.2023.3339239.

[160] C. Zeng, J.-B. Wang, M. Xiao, *et al.*, "Task-oriented semantic communication over rate splitting enabled wireless control systems for urllc services," 2, vol. 72, 2024, pp. 722–739. DOI: 10.1109/TCOMM.2023.3325901.

[161] S. E. Trevlakis, N. Pappas, and A.-A. A. Boulogeorgos, "Toward natively intelligent semantic communications and networking," vol. 5, 2024, pp. 1486–1503. DOI: 10.1109/OJCOMS.2024.3371871.

[162] H. W. Chung, L. Hou, S. Longpre, *et al.*, "Scaling instruction-finetuned language models," 70, vol. 25, 2024, pp. 1–53.

[163] M. Wen, Q. Li, K. J. Kim, *et al.*, "Private 5g networks: Concepts, architectures, and research landscape," 1, vol. 16, 2022, pp. 7–25. DOI: 10.1109/JSTSP.2021.3137669.