Grant Agreement No.: 101192650

Type of action: HORIZON JU Research and Innovation

Topic: HORIZON-JU-SNS-2024-STREAM-B-

Call: HORIZON-JU-SNS-2024

D4.1 INTERMEDIATE REPORT ON DISTRIBUTED INTELLIGENCE SOLUTIONS FOR RESOURCE PROVISIONING, EDGE- CLOUD CONTINUUM AND CONFLICT RESOLUTION FOR INTEGRATED COMPONENTS

Work package	WP4
Task	T4.1, T4.2. T4.3
Due date	30/06/2026
Submission date	30/06/2026
Deliverable lead	University of Bologna
Version	1.0
Dissemination Level	PU
Editor	Carla Amatetti and Bruno De Filippo (UniBo)
Authors	Carla Amatetti (UniBo), Alessandro Guidotti (UniBo), Alessandro Vanelli-Coralli (UniBo), Bruno De Filippo (UniBo), Zefu Gao (UniBo), Navid Hosseinzadehdehlan (UniBo), Farid Eshaghzadeh (UniBo) Luis Blanco (CTTC), Engin Zeydan (CTTC), Cristian J. Vaca-Rubio (CTTC), Husnain Sahid (CTTC) Stefanos Plastras (NCSRD), Harilaos Koumaras (NCSRD) Efi Dvir (CER) Sotirios Spantideas (FDI), Anastasios Giannopoulos (FDI), Marilina Bartsioka (FDI)

	Gianluca Fontanesi (NOKIABL) Alice Piemonti (MAR), Vito Cianchini (MAR) Berk Buzcu (HES-SO), Gianluca Rizzo (HES-SO) Marcel Garczyk (PUT), Paweł Sroka (PUT), Paweł Hatka (PUT), Paweł Płaczkiewicz (PUT) Cristian Brunetto (LINKS) Sihem Cherrared (ORANGE) Balaji Kirubakaran (DLR), Purva Joshi (DLR) Javier Velazquez Martinez (TID), Samuel Santos Hernan (TID), Luis Contreras Murillo (TID) Kanakaris Nikolaos (IPTO), Stamatiou Nikolaos (IPTO) Angelos Antonopoulos (NBC), Godfrey Kibalya (NBC)
Reviewers	Nuria Trujillo (HSA), Oriol Font (SRS), Tomaso de Cola (DLR), Gil Kedar (CER)

Abstract	D4.1 is the intermediate report on the design of the AI-based solutions developed for achieving effective resource provisioning, edge-cloud continuum, multi-connectivity towards energy efficiency enabling, and conflict resolution in integrated components. Therefore, the deliverable describes the algorithms, their possible implementation in the UNITY-6G architecture, the project KPIs that they target, and the preliminary results of the AI-based solutions/algorithms.
Keywords	O-RAN, AI, NTN, Semantic Communications

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	5/03/2026	ToC definition	Carla Amatetti (UNIBO)
V0.2	25/05/2026	Version with all contributions	All
V0.3	05/06/2026	First harmonization	Carla Amatetti (UNIBO)
V0.4	16/06/2026	Ethical supervisor review	Carmela Occhipinti (Cyberethics Lab)
V0.5	19/06/2026	Internal and external review	Reviewers
V0.6	26/06/2026	Version with comments implemented	All
V0.7	29/06/2026	Second harmonization	Carla Amatetti and Bruno De Filippo (UNIBO)
V1.0	30/06/2026	Submitted version	Engin Zeydan (CTTC)

DISCLAIMER

Co-funded by the European Union

6GSNS

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

The **unity-6G** project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No *101192650*. This work has received funding from the Swiss State Secretariat for Education, Research, and Innovation (SERI).

COPYRIGHT NOTICE

EXECUTIVE SUMMARY

This deliverable, D4.1, is the first output of WP4 and presents the distributed, AI-powered algorithms designed for network and service resource management in the UNITY-6G integrated 6G system. UNITY-6G targets an AI-native, sustainable and resilient architecture that unifies terrestrial and non-terrestrial networks, the Open RAN domain and the edge-cloud continuum under a hierarchical, multi-domain management and orchestration framework. The purpose of this deliverable is to define, position, and begin validating the algorithmic building blocks that will realise this vision.

The algorithms are organised into four categories: i) distributed AI for the edge-cloud continuum and network and service resource management in dynamic multi-tenant environments, covering orchestration, resource allocation and routing, and forecasting and prediction; ii) sustainable integrated networks and energy-efficient enablers, addressing planning, radio-resource management and grid-aware operation; iii) semantic-communication techniques that improve communication efficiency while preserving service quality, and iv) conflict-resolution mechanisms that allow multiple AI controllers to operate safely on shared resources.

For each algorithm, the deliverable provides a description of the proposed approach, its positioning with respect to the state of the art, and explicitly maps it onto the components of the UNITY-6G architecture, so that every solution has a well-defined place within the system. In parallel, the project key performance indicators (KPIs) to be addressed have been associated to each algorithm. Collectively, the contributions address the core objectives of the project: ultra-high service continuity and resilience, energy efficiency and sustainability, efficient use of communication and computational resources across integrated terrestrial and non-terrestrial segments, and safe coordination among autonomous intelligent controllers.

At this stage, the deliverable is primarily descriptive, establishing the design and architectural positioning of each algorithm. As detailed in the conclusions, D4.2 and D4.3, i.e., the next versions of this deliverable, will consolidate the algorithm portfolio, provide quantitative results against the target KPIs, and progress the integration of the selected algorithms into the project PoCs and experimental scenarios.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5GC	5th Generation Core
5QI	5G QoS Identifier
ACT	Actuator
AE	Analytics Engine
AI	Artificial Intelligence
AoI	Age of Information
AP	Access Point
BSR	Buffer Status Report
CAC	Call Admission Control
CD	Conflict Detector
CFE	Container Forecasting Engine
CFL	Clustered Federated Learning
CMF	Conflict Management Framework
CN	Core Network
CNF	Cloud-native Network Function
CNN-RNN	Convolutional Neural Network – Recurrent Neural Network
CP	Control Plane
CS	Cell Switching
CSR	Cyber Security and Resilience
CTI	Cooperative Transport Interface
CU	Central Unit
CU-SP	Central Unit Semantic Plane
D3QL	Deep Q-Learning
DC	Direct Conflict
DDQN	Double Deep Q-Network
DE	Decision Engine
DFOO	Dynamic F-gNB ON/OFF
DLO	Decarbonization Level Objective
DLT	Distributed Ledger Technology
DMO	Domain Management Orchestrator
DNN	Deep Neural Network
DPIR	Deep Predictive Interference-Aware Routing
DQN	Deep Q-Network
DR	Delay Rate
DRL	Deep Reinforcement Learning
DSL	Domain-Specific Language
DSNB	Distributed State Network of Brokers
DT	Digital Twin
DU	Distributed Unit
E2E	End-to-End
EA	Edge Agent
EC	Energy Consumption

EE	Energy Efficiency
EPSO	Enhanced Particle Swarm Optimisation
ES	Experimental Scenario
ETSI	European Telecommunications Standards Institute
FL	Federated Learning
GNN	Graph Neural Network
HLO	High-Level Orchestrator
HPA	Horizontal Pod Autoscaler
IBN	Intent-Based Networking
IDM	Infrastructure Domain Manager
IDMO	Inter-Domain Management Orchestrator
IE	Infrastructure Element
iEMS	Inter-domain Energy Management System
IoT	Internet of Things
ISD	Inter-Site Distance
JSCC	Joint Source-Channel Coding
KAN	Kolmogorov-Arnold Network
KPI	Key Performance Indicator
LEO	Low Earth Orbit
LLM	Large Language Model
LLO	Low-Level Orchestrator
LoS	Line-of-Sight
LSTM	Long Short-Term Memory
M2P	Motion-to-Photon
MAC	Medium Access Control
MARL	Multi-Agent Reinforcement Learning
MBS	Macro Base Station
MCDA	Multi-Criteria Decision Analysis
MCP	Model Context Protocol
MCS	Modulation and Coding Scheme
MDP	Markov Decision Process
MEC	Multi-access Edge Computing
MIP	Mixed-Integer Programming
ML	Machine Learning
MS	Monitoring Service
MSE	Mean Squared Error
NDT	Network Digital Twin
NF	Network Function
NFV-SDN	Network Function Virtualization – Software-Defined Networking
NLoS	Non-Line-of-Sight
NPN	Non-Public Network
NTN	Non-Terrestrial Network
O-RAN	Open Radio Access Network
OPEX	Operational Expenditure
P2P	Peer-to-Peer
PCRA	Power Control and Resource Allocation

PDR	Packet Delivery Ratio
PF	Proportional Fairness
PoC	Proof of Concept
QoE	Quality of Experience
QoS	Quality of Service
RA	Resource Allocation
RAG	Retrieval-Augmented Generation
RAT	Radio Access Technology
RBUR	Resource Block Utilisation Ratio
RDF	Resource Description Framework
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RRUR	Radio Resource Utilisation Ratio
RSRP	Reference Signal Received Power
RT	Real-Time
SA-AE	Semantic-Aware Analytics Engine
SA-DE	Semantic-Aware Decision Engine
SA-MS	Semantic-Aware Monitoring Service
SBMA	Service-Based Management Architecture
SBS	Small Base Station
SC	Smart Contract
SE	Spectral Efficiency
SFS	Semantic Fidelity Score
SINR	Signal-to-Interference-plus-Noise Ratio
SLA	Service Level Agreement
SLO	Service Level Objective
SMBS	Super Macro Base Station
SMO	Service Management and Orchestration
SoA	State-of-the-Art
SPA	Single-Page Application
TFT	Temporal Fusion Transformer
TM	Traffic Management
TN	Terrestrial Network
TU	Time Utility
UDN	Ultra-Dense Network
UE	User Equipment
UI	User Interface
UP	User Plane
VNF	Virtual Network Function
VPP	Virtual Power Plant
XR	Extended Reality
ZSM	Zero-touch Service Management

1 INTRODUCTION

This deliverable presents the design of Artificial Intelligence (AI)-driven solutions for 6G networks to be integrated within the UNITY-6G architecture.

A distinctive feature of the UNITY-6G architecture is its AI-native design paradigm, in which AI is embedded as a fundamental capability of the system rather than introduced as an external optimisation layer. Under this paradigm, intelligence is inherently distributed across the architecture by means of autonomous agents that continuously monitor, analyse, decide upon, and act on the network state.

The AI-native approach affords several key capabilities. First, it supports distributed and hierarchical intelligence, whereby local decisions are taken close to the infrastructure to enable rapid reactions, while global optimisation is performed at higher orchestration levels. Second, it enables autonomous closed-loop operation, thereby reducing the need for human intervention and supporting zero-touch service management. Third, it facilitates cross-domain coordination, as agents interact through the service-based management bus and exchange telemetry, insights, and intents in a standardised manner.

Moreover, the AI-native paradigm is tightly integrated with core architectural components, notably the data continuum and the digital twin framework, thereby enabling continuous learning, simulation-driven optimisation, and data-driven decision-making. This integration establishes a robust foundation for the advanced AI-based resource provisioning, edge–cloud orchestration, and conflict-resolution mechanisms that constitute the principal focus of this deliverable.

Specifically, this deliverable defines a first category of AI-based algorithms intended to optimise service-networking operations through the control of radio resource allocation, the functional decomposition and placement of network elements, the allocation of computational and storage resources, and the automation of 6G network operation and service/resource provisioning, within the broader context of edge–cloud and overall network management. A second category of algorithms addresses sustainability; in particular, data-driven strategies are designed to achieve low energy consumption and a reduced carbon footprint in integrated networks. The deliverable further investigates efficient conflict-resolution strategies, with an emphasis on prediction techniques for proactive conflict detection and avoidance. Finally, the paradigm of semantic communication is also considered.

It should be noted that this deliverable reports the outcomes of the first phase of three tasks within Work Package (WP) 4, namely T4.1, T4.2, and T4.3. Accordingly, the present document

defines the initial design of AI-driven solutions for 6G networks, a subset of which is integrated into the various Proofs of Concept (PoCs) with the aim of validating the advances proposed by the project. The foundations established herein will be refined iteratively, and the preliminary results reported here will be extended in subsequent phases of WP4 and presented in forthcoming deliverables.

1.1 RELATIONS WITH OTHER WPS

The work presented in this deliverable takes inputs from:

- WP2, where the definition of the UNITY-6G architecture (D2.3) is taking place, jointly with the specification of the project's goals – in terms of KPIs and KVIs (D2.1)
- WP3, which defines the different components of the UNITY-6G architecture, i.e., open radio access network (O-RAN), including its near-real time (RT) management and control by means of the RAN Intelligent Controller (RIC), the core network, the non-3GPP RAN (IEEE 802.11 RAN) and its O-RAN enabled access points, the transport (xHaul) network and, finally, non-terrestrial networks component (D3.1).

At the same time, the outcomes produced in WP4 serve as input to WP3.

Finally, some of the results presented here (of WP4) will be integrated into the PoC that are being developed in WP6.

1.2 STRUCTURE OF THE DELIVERABLE

The deliverable is organized as follows:

- Section 2 briefly recalls the UNITY-6G architecture.
- Section 3 reports the AI algorithms for the edge-cloud continuum scenarios and for network and service resource management in multi-tenant environments.
- Section 4 is dedicated to the design of AI-based algorithms for sustainable integrated networks.
- Section 5 focuses on the semantic communication frameworks, including the description of both the considered framework and the developed algorithms.
- Section 6 is devoted to the design of conflict resolution strategies.
- Section 7 summarizes the algorithms developed in WP4 and presented in this deliverable, highlighting the project KPIs targeted by the algorithms themselves.
- Section 8 describes how potential ethical risks due to AI are addressed.
- Section 9 concludes the deliverable.

2 OVERVIEW OF THE UNITY 6G ARCHITECTURE

The UNITY-6G architecture represents a comprehensive framework designed to address the increasing complexity, heterogeneity, and intelligence requirements of future 6G networks. It introduces a unified, service-based and AI-native architecture capable of orchestrating and managing resources across highly diverse domains, including terrestrial and NTN, RAN, core and edge-cloud infrastructures.

A key motivation behind this architecture is overcoming the fragmentation observed in current 5G systems, where management and orchestration functions operate in isolation and lack mechanisms for coordinated, cross-domain optimization. UNITY-6G addresses this limitation through a unified service-based management architecture (SBMA), in which all functionalities are exposed as interoperable services and consumed via standardized interfaces. This enables seamless end-to-end (E2E) lifecycle management, from service creation and deployment to monitoring, adaptation, and termination, across the entire edge–cloud continuum.

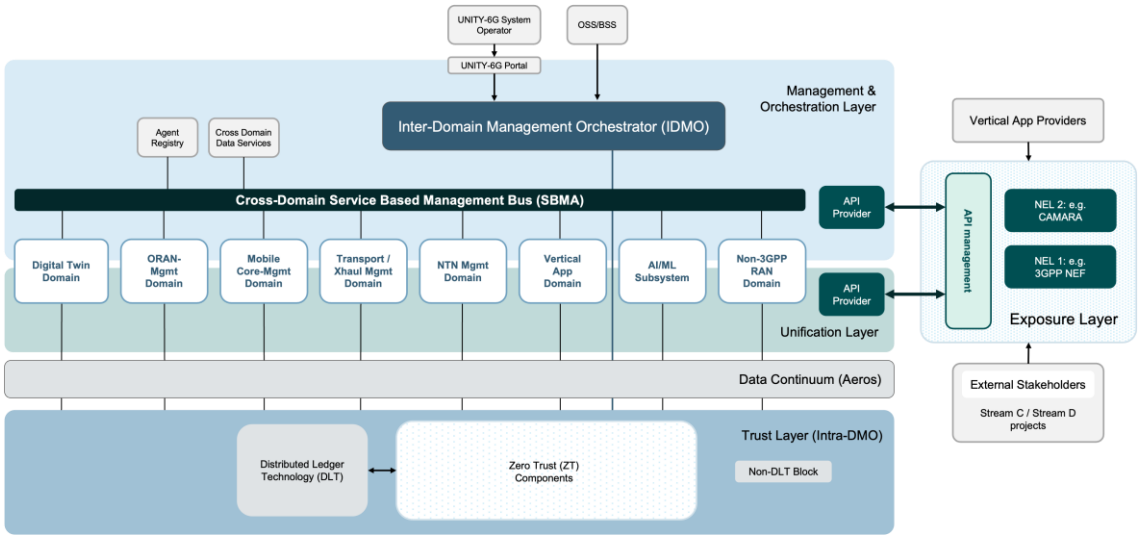


Figure 1. UNITY-6G Architecture

At a structural level, the architecture (Figure 1) is organized into multiple layers, including the management and orchestration layer, the unification layer, the data continuum, the network exposure framework, and the trust layer. These layers collectively provide the required abstractions, coordination mechanisms, and governance capabilities to support cross-domain service provisioning, ensuring scalability, flexibility, and interoperability in heterogeneous 6G environments.

A key aspect of the architecture is its hierarchical orchestration model, centered on the inter-domain management orchestrator (IDMO), supported by domain management orchestrators (DMOs) and infrastructure domain managers (IDMs). This structure enables intent-based service provisioning, where high-level requirements are translated into actionable configurations and executed across multiple domains while ensuring consistency, conflict resolution, and real-time adaptation.

One of the most distinctive features of the UNITY-6G architecture is its AI-native design paradigm, where artificial intelligence is embedded as a fundamental capability of the system rather than an external optimization layer. In this approach, intelligence is inherently distributed across the architecture through autonomous agents that continuously monitor, analyse, decide, and act upon the network state.

This paradigm is operationalized through the adoption of the MS–AE–DE–ACT tuple model (Monitoring System, Analytics Engine, Decision Engine, and Actuator), which provides a unified abstraction for implementing closed-loop control functions across all domains. Each agent in the system is defined by a combination of these capabilities, enabling flexible deployment of distributed intelligence across edge, cloud, and device layers.

The AI-native approach enables several key capabilities. First, it supports distributed and hierarchical intelligence, allowing local decisions to be taken close to the infrastructure for fast reactions, while global optimization is handled at higher orchestration levels. Second, it enables autonomous closed-loop operation, reducing the need for human intervention and supporting zero-touch service management. Third, it facilitates cross-domain coordination, as agents interact through the service-based management bus, sharing telemetry, insights, and intents in a standardized manner.

Furthermore, the AI-native paradigm is tightly integrated with key architectural components such as the data continuum and the digital twin framework, enabling continuous learning, simulation-driven optimization, and data-driven decision-making. This provides a solid foundation for implementing advanced AI-based resource provisioning, edge-cloud orchestration, and conflict resolution mechanisms, which are the main focus of this deliverable.

Complementing the AI-native paradigm, the UNITY-6G architecture incorporates a dedicated AI/ML subsystem, which plays a central role in supporting the development, deployment, and operation of intelligent algorithms across all domains. This subsystem is designed as an independent yet fully integrated domain, providing AI/ML capabilities “as a service” to the rest of the architecture.

The AI/ML subsystem is responsible for managing the complete lifecycle of machine learning models, including data collection, training, validation, deployment, inference, and evaluation. It enables the construction of AI pipelines aligned with the MS–AE–DE–ACT paradigm, supporting training, inference, and evaluation workflows that can operate either centrally or in a distributed manner across the edge–cloud continuum.

A key capability of this subsystem is its ability to operate across multiple domains, collecting data from heterogeneous domain sources. This allows the development of cross-domain AI models that can capture system-wide correlations and enable more accurate and efficient decision-making. Additionally, the subsystem supports advanced techniques such as distributed learning and federated learning, enabling privacy-preserving and scalable model training.

This domain subsystem facilitates the deployment of analytics and decision-making functions that can dynamically adapt to changing network conditions, optimize resource utilization, and coordinate actions across domains.

Moreover, the subsystem interacts closely with other architectural components, including the data continuum, which provides access to real-time and historical data, and the trust layer, which ensures the integrity and reliability of AI-driven decisions. This holistic integration ensures that AI/ML functionalities are not isolated components but are deeply embedded into the operational fabric of the UNITY-6G system.

Overall, the UNITY-6G architecture provides a robust and scalable framework that integrates distributed intelligence, AI-driven orchestration, and cross-domain coordination, making it particularly suitable for the development of the solutions addressed in this deliverable.

3 DISTRIBUTED AI-POWERED EDGE-CLOUD CONTINUUM AND NETWORK AND SERVICE RESOURCE MANAGEMENT IN DYNAMIC MULTI-TENANT ENVIRONMENT

3.1 INTRODUCTION

This section is devoted to the definition and design of distributed AI-based algorithms for network and service resource management across the edge-cloud continuum and the overall integrated network. The algorithms address service-networking by controlling radio-resource allocation, functional decomposition and placement of network elements, transmitted power, signal bandwidth and operating frequencies according to the prevailing network demand as well as the allocation of computational and storage resources, and the automation of 6G network and service provisioning.

Consistent with the UNITY-6G architecture, the algorithms presented here are designed to operate within the hierarchical management-and-orchestration framework, spanning the IDMO and the per-domain DMOs, and to instantiate the AI-native closed-loop control pattern composed of the MS, AE, DE and ACT.

The following three categories of algorithms have been identified:

- Network orchestration: AI-driven, intent- and SLO-based orchestration of services and resources across the edge-cloud continuum and integrated TN/NTN domains, including agentic and energy-aware orchestration frameworks.
- Resource allocation, traffic load balancing, and routing: distributed mechanisms for task offloading, traffic steering, dynamic routing and network recovery that optimise the use of communication and computational resources.
- Forecast and prediction: learning-based forecasting and anomaly-detection methods that anticipate demand, failures and resource consumption to enable proactive, rather than reactive, network management.

The remainder of the section is organised accordingly. Section 3.2 reviews the state of the art (SoA) for each of the three categories, and Section 3.3 describes the individual algorithms. For every algorithm, a description of the proposed approach is followed by its high-level mapping onto the UNITY-6G architecture and, where available, by preliminary results. In addition, the project KPIs (defined in D2.1) targeted by the aforementioned algorithms are listed.

3.2 STATE OF THE ART

3.2.1 Network orchestration

3.2.1.1 Agentic Orchestration framework

6G envisions achieving hyper-connectivity, ultra-low latency, and integration of a vast number of devices, necessitating extensive edge-cloud infrastructure deployment. To address the complexity of orchestrating services across such large-scale distributed infrastructure, the concept of zero-touch service orchestration has been proposed, emphasizing AI/ML-driven automation for dynamic, real-time service provisioning with minimal human involvement [1].

However, traditional AI's reliance on structured input poses challenges for this vision. This has prompted exploration of Generative AI (GenAI), particularly Large Language Models (LLMs) in recent works due to its ability to interpret and decompose high-level user intent and reason over massive data-sets [2]-[4]. LLMs have also been leveraged for tasks such as zero-shot time series forecasting [5], synthetic data generation [6], and automated fault detection and mitigation [7].

However, the service orchestration use cases envisaged in 6G will demand more adaptive, intelligence-driven orchestration solutions capable of handling complex, evolving requirements and diverse and sometimes conflicting user intents, making conventional AI-driven approaches inadequate. These use cases often involve multi-step processes, planning, and reasoning, which exceed the capabilities of stand-alone LLMs, as they may require access to external tools and real-time information beyond the LLMs' internal knowledge base.

To address these limitations, Agentic AI has emerged as a promising approach, with potential to offer enhanced decision-making for adaptive, intelligent orchestration in complex scenarios [8][9]. Agentic AI enables autonomous, context-aware decision-making, continuous learning, and real-time adaptability to uphold user intents in resource-constrained, dynamic environments, a requisite requirement for zero-touch orchestration especially in cloud-native scenarios characterized by dynamic states. Recent studies have explored the potential of Agentic AI in emerging use cases such as software development [9] generative information retrieval [10], and zero-touch network management [11]. Other works have explored architectures and frameworks for LLM-based agents integration. For instance, AIOpsLab, proposed in [7], introduces a prototype for autonomous cloud service self-healing using an agent-cloud interface to simulate faults and enable resolution via LLM-based agents. Similarly, [2] presents an end-to-end Intent-Based Networking (IBN) architecture that uses LLMs for natural language interaction, intent decomposition, and translation, validated using tools from the 5G EURECOM facility. MAESTRO [12] offers a

collaborative intent-based network management framework where LLM agents, combined with optimization and reasoning modules, manage resource negotiation and align stakeholder goals across the business, service, and network planes. Finally, [13] introduces a resource allocation scheme that uses LLM agents to interpret user intent and feed structured inputs into a Deep Reinforcement Learning (DRL) model, enhancing dynamic resource management in complex 6G scenarios.

Despite growing interest in Agentic AI, its potential for cloud-native edge-cloud service orchestration remains underexplored. This area presents significant challenges due to: (i) the heterogeneity of edge-cloud nodes in terms of computational and energy capacities, (ii) the massive data volumes generated by edge devices, (iii) the limited resources available at edge servers and devices, and (iv) the cloud-native shift toward microservices and containerized, distributed service deployments, all of which increase orchestration complexity.

3.2.1.2 Green Orchestration through multi-criteria decision analysis

Future 6G systems require an orchestration framework capable of managing cross-domain complexity while embedding energy awareness and sustainability into core systems and placement decisions. Currently, several industry initiatives address multi-domain automation. The ETSI Zero-touch Service Management (ZSM) framework outlines intent-based, closed-loop automation [14], while the O-RAN Alliance specifies RAN Intelligent Controllers (RICs) for AI-based radio optimization [15]. At the system level, the 3GPP SBMA facilitates multi-domain management and data-driven control [16]. However, a critical limitation of the current state of the art is that these frameworks lack explicit, native mechanisms to incorporate energy sustainability as a primary decision parameter for end-to-end mobile network configuration.

Recent literature on green orchestration has attempted to integrate energy consumption models into scheduling decisions. Approaches such as renewables-aware virtual network function (VNF) placement and energy-efficient service chaining show future. Yet, these models typically operate strictly at the post-deployment resource management level, lacking coordination across multiple hierarchical orchestration layers. While projects like Hexa-X [17] and ADROIT6G [18] have pushed for AI-native, hierarchical orchestration, the holistic view of near-real-time energy telemetry (e.g., data from Virtual Power Plants) directly into inter-domain orchestration decision loops remains largely unexplored.

3.2.1.3 Intent-based Transport Configuration

A transport segment that is improperly dimensioned or configured can compromise end-to-end behaviour even when the radio and core functions are otherwise correctly instantiated [19], [20]. This problem becomes more severe when the same infrastructure must simultaneously

support different traffic classes and slice profiles, each with distinct performance expectations. The traditional operational model, based on static templates, manually selected protection modes, or direct device-by-device parameterization, scales poorly when the network is dynamic and when service requests must be fulfilled in a context-sensitive manner [21][22]. Accordingly, a more abstract, policy-aware, and adaptive mode of transport management becomes necessary [23] [14].

3.2.1.4 Service Level Objectives (SLO)-based Proactive Autonomous Network Orchestration

Conventional network management approaches are predominantly reactive, relying on threshold-based alarms and human-driven intervention mechanisms that are triggered only after service degradation has been observed [24] [14]. While such approaches have been adequate for static and moderately complex environments, they are no longer sufficient in modern scenarios characterized by fluctuating traffic patterns, dynamic resource allocation, and strict performance isolation across tenants [20][23]. In particular, reactive mechanisms fail to prevent Service Level Agreement (SLA) breaches in premium services, as corrective actions are inherently delayed until after degradation has already impacted service quality.

The emerging “AI for Networks” paradigm provides a transformative opportunity to address these limitations. AI-driven network automation enables the continuous analysis of large-scale telemetry data, facilitating the identification of subtle patterns that precede performance degradation events [25]. Within this paradigm, the introduction of agentic AI, where autonomous intelligent agents act with a high degree of independence, represents a significant step beyond traditional automation frameworks.

3.2.1.5 The aeriOS framework

The development of the aeriOS Meta-Operating System is positioned at the intersection of four rapidly evolving technical domains. By evaluating the SoA in these categories, the unique value proposition of aeriOS, shifting from static management to autonomous, distributed intelligence, is clearly defined within the UNITY-6G architecture. Traditional orchestration, governed by ETSI NFV and cloud-native standards like Kubernetes, relies on imperative descriptors where resources are explicitly defined by the user. While recent research into Intent-Based Networking (IBN) has introduced goal-oriented management, existing solutions remain largely centralized and reactive. aeriOS advances the SoA by implementing a multi-level orchestration hierarchy (HLO and LLO) that translates high-level "Intention Blueprints" into implementation-ready manifests. Unlike current frameworks that require a persistent connection to a central controller, aeriOS supports "Island Mode" operations, allowing edge

domains to maintain autonomous service orchestration during backhaul failures, a critical requirement for 6G resilience [26].

Current SoA for Placement, Instance Assignment, Request Prioritization, and Allocation (PIRA) typically involves centralized solvers or basic DRL models that optimize for single-objective KPIs like throughput. These models often fail to account for the energy-sustainability constraints or the combinatorial complexity of a hyper-distributed edge-cloud continuum [27]. aeriOS moves beyond this baseline by integrating the ORIENT algorithm, which utilizes Graph Neural Networks (GNN) and Double Dueling Deep Q-Learning (D3QL). This allows the framework to treat the 6G network as a dynamic topological graph, enabling power-aware placement decisions that balance strict sub-millisecond latency with the project's aggressive energy-saving targets. The management of heterogeneous data in 5G-Advanced is often fragmented across siloed monitoring tools. While the ETSI NGSI-LD standard provides a blueprint for context information management, its integration into the real-time orchestration loop is not yet a standard industry practice [28]. aeriOS bridges this gap by positioning an NGSI-LD-compliant Data Fabric at the core of its architecture. By utilizing the Orion-LD Context Broker, aeriOS ensures that the orchestrator is "context-aware," meaning it can factor in real-time telemetry such as fluctuating Non-Terrestrial Network (NTN) link quality to trigger proactive workload migrations. This provides a unified data plane that far exceeds the capabilities of current fragmented edge-management tools.

The SoA for service exposure in telecommunications is currently defined by the 3GPP Common API Framework (CAPIF). While many platforms offer proprietary Northbound APIs, few natively integrate these with the underlying compute and network orchestration [29]. aeriOS aligns with and extends this SoA by exposing all Meta-OS capabilities, including orchestration, data access, and self-management, through a CAPIF-compliant API Gateway. This allows UNITY-6G vertical applications to request complex, cross-domain resources via standardized OpenAPI/AsyncAPI endpoints. By providing this unified, secure access point, aeriOS transitions the network from a simple transport pipe into a programmable service environment, fulfilling the 6G vision of an open and integrated ecosystem.

3.2.2 Resource allocation, traffic load balancing, and routing

3.2.2.1 TN-NTN Traffic Offloading

The integration of NTNs with terrestrial networks (TNs) has attracted significant research interest, particularly in the areas of traffic offloading, load balancing, and resource management across heterogeneous radio access technologies (RATs). Based on the available literature, the following observations arise:

- The earliest approaches rely on manually designed thresholds and rule-based handover triggers (e.g., radio resource utilization ratio (RRUR) thresholds, time or location events in [30][31]). The field has progressively shifted toward metaheuristic optimisation (EPSO, [31]) and then fully to deep reinforcement learning (DQN in [32], multi-agent DRL in [33]), where the network learns offloading policies directly from interaction with the environment rather than from pre-defined rules.
- Early works balance load within a single terrestrial RAT and treat the satellite as a simple overflow buffer. More recent contributions consider true multi-layer NTNs, combining low-Earth orbit (LEO), medium Earth orbit (MEO), geostationary Earth orbit (GEO), and high altitude platforms (HAPs) simultaneously, and optimise resource allocation across all layers jointly, as in [33] and [34].

Despite this progress, some aspects remain underexplored: time-varying fading and Doppler modelling for LEO links, service-class differentiation (ultra reliable low latency communication (URLLC) vs. enhanced mobile broadband (eMBB)) within satellite offloading, an analysis of the complexity of the networks, as well as a joint design of the integrated network and a clear mapping of the AI pipeline on network architecture. In addition, there is a need to jointly optimise throughput, fairness (Jain's index), latency, and load variance, recognising that optimising a single metric often degrades the others.

3.2.2.2 Energy-Efficient Decentralized Computation Offloading in the Continuum

Existing task offloading approaches can be broadly divided into optimization-based, heuristic, and learning-based methods. Classical optimization methods, including mixed-integer formulations and decomposition-based schemes, often require a global view of the system and become computationally expensive in dynamic large-scale environments [35]. Heuristic approaches can provide faster decisions but usually require manual tuning and may need to be re-executed whenever the environment changes. Existing DRL-based approaches improve adaptivity, but many of them remain centralized, focus on a single objective, or insufficiently capture the simultaneous horizontal and vertical offloading options of a true computing continuum [36][37]. The Decentralized Computation Offloading for Energy Efficiency in the Continuum (DECOFFEE) framework explicitly identifies these gaps: current approaches often neglect joint optimization of latency, energy consumption, and workload drop rate in a decentralized and proactive manner.

3.2.2.3 Clustered Federated Learning for Managed Wi-Fi

Managed Wi-Fi systems represent a salient response to this challenge; wherein centralized controllers are employed to optimize Access Point (AP) radio parameters and ensure robust

connectivity through continuous monitoring of operational statistics. While such operator-managed solutions have significantly enhanced Wi-Fi network administration, the sheer volume and intrinsic complexity of data generated by Wi-Fi devices mandate advanced analytical capabilities. In this context, AI-driven methodologies are proving indispensable, offering robust capabilities for processing large datasets and extracting actionable insights to streamline network management.

Among the various AI paradigms, Federated Learning (FL), and more specifically Clustered Federated Learning (CFL), presents a particularly compelling solution. CFL facilitates the development of multiple, specialized Machine Learning (ML) models, each tailored to accurately reflect the distinct statistical distributions inherent in the data collected by different APs. A key advantage of CFL is its ability to achieve this without requiring the direct transfer of raw data from APs to a central cloud controller, thereby addressing potential data granularity constraints imposed by centralized controller while also enhancing privacy.

Despite the burgeoning interest in distributed learning frameworks for Wi-Fi systems, the comprehensive integration of CFL within this domain remains an area of nascent exploration. Prior investigations have applied CFL to specific applications such as Wi-Fi fingerprinting-based indoor localization [38], Wi-Fi sensing systems utilizing hierarchical clustering [39], and Wi-Fi-based intrusion detection [40]. Furthermore, FL has been applied to traffic prediction in Wi-Fi networks [41], [42]. However, fruitful integration of CFL strategies within managed Wi-Fi solutions is yet largely unexplored. This integration is posited to enable more intelligent and adaptive solutions for centrally-orchestrated Wi-Fi networks.

3.2.2.4 ML-Based Dynamic Network Routing

Dynamic routing in wireless and integrated 6G networks is a coupled optimization problem because the quality of a selected path is not determined only by the links belonging to that path. In wireless environments, simultaneous transmissions on spatially or spectrally overlapping links may mutually interfere, even when the corresponding flows do not share the same physical links. Therefore, the routing decision of one flow may degrade the achievable rate of another flow. This makes the routing problem fundamentally different from conventional shortest-path, delay-aware, or load-aware routing, where paths are often evaluated independently or with only additive link costs.

Existing dynamic-routing methods commonly select paths using additive or locally computed criteria, such as shortest-path cost, queue/backpressure differentials, and path-level delay, load, or available-capacity estimates [43][44][45]. Although such mechanisms can adapt to traffic and link-state variation, they are not generally formulated as a joint optimization of the

complete multi-flow route profile, in which the route assigned to one flow changes the interference and effective capacity experienced by other flows. Learning-based routing schemes extend this adaptivity under uncertain and time-varying conditions [44][46]; however, representative approaches typically react to the observed network state or allocate routes sequentially to individual traffic demands, rather than predicting and jointly optimizing a time-indexed routing schedule that captures the evolving interference footprint of all concurrently active routes. Dedicated interference-aware and joint multi-flow approaches have been proposed [47][48]. These works should be recognized as important exceptions; the distinct contribution sought here is predictive joint route scheduling under evolving interference, rather than interference awareness alone.

3.2.2.5 Network Recovery & Disaster Resilience via Multi-Agent Reinforcement Learning

Modern 5G/6G networks depend on a hierarchical core: UPF (User Plane Functions), gNBs (base stations), and wireless transport relay nodes, including integrated access backhaul (IAB)-like relay functionality, are connected through high-capacity fiber or wireless backhaul to one or more central data centers [49][50]. In a disaster—— earthquake, flooding, military strike, large-scale power failure, or severe transport-network outage—— the physical links that form this core can be cut simultaneously, partitioning the surviving radio and transport nodes from the internet, from the core network, and from each other [51].

The surviving infrastructure, including gNBs, relay nodes, and edge servers that remain physically operational, may become functionally isolated. User equipment (UE) devices associated with those nodes can still communicate locally, but lose access to external services, emergency-service anchors, and network paths that require routing beyond their immediate cell or local island [49] [51]. This creates a post-severance operating condition in which infrastructure is available, but the normal control-plane and user-plane connectivity assumptions no longer hold.

We define this network severance state as Island Mode. During Island Mode:

- all core or UPF nodes are down or unreachable via normal backhaul;
- each surviving gNB can only serve its local UEs through intra-cell or locally available communication paths;
- UE-to-UE flows that require routing through more than one gNB may fail completely;
- emergency responders, life-safety services, and coordination messages may become unreachable;

- normal centralized orchestration may be unavailable, delayed, or only partially connected.

Without network recovery, this state persists until physical infrastructure repair or manual network reconfiguration is completed. This may take hours or days, while the absence of connectivity directly impairs disaster response: rescue coordination fails, medical telemetry is interrupted, public-safety communication collapses, and local service continuity is lost [51].

Conventional recovery mechanisms are insufficient in this setting. Pre-configured fallback routes assume that the surviving topology is known in advance, whereas real disaster events may remove arbitrary combinations of nodes and links [52]. A centralized software-defined networking (SDN) controller may itself be disconnected from part of the damaged network [53]. Manual re-provisioning is too slow for emergency scenarios, and fixed relay-chain solutions cannot adapt to the actual set of surviving nodes, available wireless links, radio conditions, UE locations, and service priorities [54], [51]. Broadcast flooding may maintain partial reachability, but it consumes spectrum and battery resources and does not optimize for connectivity quality, interference, or emergency-flow priority [55].

The fundamental challenge is therefore that the set of surviving nodes, available wireless links, traffic demands, and UE locations is unknown and different in every disaster event. A system that learns to recover from a distribution of disaster scenarios, rather than from one predefined topology, is required.

3.2.3 Forecast and prediction

3.2.3.1 Hy-DREAM for Anomaly Detection

Various categories of machine learning methods exist for detecting anomalies, including supervised, semi-supervised, and unsupervised approaches [56]-[58].

Supervised detection treats the problem as a classification task using labeled data to define boundaries between normal and anomalous states; however, such labels are frequently scarce, biased toward specific incidents, and incomplete. Since anomalies are dynamic, supervised models often overfit known patterns and struggle to identify new types of incidents, which limits their utility for proactive network management.

Semi-supervised forecasting mitigates these limitations by learning exclusively from histories of normal behavior to identify departures during runtime. This strategy avoids a dependency on anomaly labels and generalizes well to new conditions by modeling expected behavior and flagging instances when reality diverges from that model. While this avoids the pitfalls of labelled data, these methods often fail to incorporate information from known anomalies and typically do not optimize for detection delay, which is a critical factor for timely intervention.

Hybrid strategies attempt to bridge this gap by integrating data from known anomalies into unlabelled pipelines, such as by merging unsupervised elements with mechanisms that utilize specific incident history. Although this can improve detection performance, such architectures are often computationally demanding because they require the deployment of extra components, creating significant hurdles for real-time application.

3.2.3.2 Radio Failure Prediction and Localized Recovery Estimation for Transport Wireless Links

Current research is moving beyond reactive, threshold-based fault monitoring toward AI-driven predictive maintenance that fuses multivariate radio KPIs, alarms, device-health telemetry, topology context and environmental information to predict radio-link degradation before service impact. Sequential models such as LSTM/TCN and attention-based architectures are used to capture temporal degradation patterns and support multi-horizon forecasts, while graph-based learning increasingly incorporates spatial and topological dependencies between neighbouring links and sites. Recent 5G work demonstrates the value of combining weather-aware graph aggregation with Transformer-based temporal modelling for radio-link-failure prediction; however, most existing approaches remain focused on binary failure prediction rather than jointly estimating root cause, fault location, time-to-failure, time-to-recovery, intervention need and uncertainty under partial or disconnected observations. The proposed approach extends this direction through multi-task, topology-aware and locally executable inference, aligned with the emerging 3GPP vision of AI-enabled network management and digital-twin-supported automation [59][60][61].

3.2.3.3 Multivariate probabilistic forecasting for resource allocation

Traditional forecasting approaches for network resource allocation are commonly based on deterministic and often univariate models [62]-[64], such as conventional LSTM architectures, which provide a single predicted value for future demand. Although these methods can learn temporal patterns from historical data, they generally analyze each resource stream independently and fail to capture the complex correlations that exist among multiple time series, corresponding to different O-RAN KPIs, cells or network slices. More importantly, deterministic single-point models do not provide information about the uncertainty associated with future observations [65], leading to overconfident predictions and potentially inefficient resource allocation decisions in highly dynamic wireless environments.

3.2.3.4 Container Forecasting Engine

The traditional approach to cloud network function (CNF) resource management is reactive: the orchestrator monitors current load and adjusts resource allocation when a utilisation

threshold is reached. This approach has two significant drawbacks. First, reactive scaling incurs cold-start penalties: when a container must scale up from scratch, it requires time to initialise and connect, during which traffic surges go unserved and QoS degrades. Second, because scaling actions occur only after demand changes are observed, operators typically maintain conservative resource margins, idle replicas, or over-dimensioned allocations to absorb unexpected load variations and avoid SLA violations, ultimately increasing infrastructure energy consumption.

These shortcomings are widely reported in the auto-scaling literature, which contrasts reactive, threshold-based rules with predictive, learning-based strategies and highlights their impact on cost and energy efficiency [66]. Even when threshold values are optimally tuned, reactive controllers remain intrinsically limited by the fact that scaling decisions are taken after workload changes are observed [67]. A growing body of work addresses this by coupling workload forecasting with the autoscaler: time-series–driven predictive scaling in Kubernetes has been shown to cut cold-start time and QoS fluctuation substantially [68], while cold-start–aware schemes based on deep predictors anticipate function arrivals to pre-provision resources ahead of demand [69]. For virtualised network functions specifically, deep-learning predictors of CPU, memory and bandwidth enable proactive scaling of service function chains with high QoS at low cost [70], and proactive resource managers for containerised microservices forecast load variations to preserve service continuity [71]. The Container Forecasting Engine follows this proactive, forecast-driven paradigm, anticipating CNF demand to mitigate cold-start penalties and curb the over-dimensioning typical of reactive operation, with direct benefits for QoS and energy consumption.

3.3 ALGORITHMS DESCRIPTION

3.3.1 Network Orchestration

3.3.1.1 Agentic orchestration framework

UNITY-6G introduces a multi-agent orchestration framework grounded in Agentic AI principles where the end-to-end service orchestration tasks are distributed among specialized agents, each embodying the principles of Perceive, Act, Reason, Evaluate, and Sustain (PARES), enabling autonomous operation while maintaining alignment with service objectives and infrastructure constraints.

Specifically, the framework provides three key innovations: i) PARES framework, structured methodology for defining and assessing the capabilities required by autonomous orchestration agents. PARES establishes the minimum functional requirements for perception, reasoning, action execution, evaluation, and long-term adaptation, providing a foundation for trustworthy

autonomous behaviour.; and ii) Hierarchical graph-of-graphs coordination architecture, distributed orchestration architecture that decomposes service lifecycle management into specialized orchestration domains while maintaining coordinated decision-making across edge and cloud environments. This architecture enables scalable orchestration while reducing decision complexity and improving modularity iii) ActSimCrit (Action–Simulation–Critic) planning methodology that validates orchestration decisions before deployment. Candidate orchestration actions are first evaluated within a digital-twin environment and subsequently assessed by an AI-based critic mechanism to ensure feasibility, policy compliance, and intent satisfaction before execution.

The proposed orchestration framework, illustrated in Figure 2, consists of three main components. At its core lies the Multi-Agent Graph, which orchestrates four specialized agents through sequential decision-making phases: Intent processing, observability analysis, planning, and infrastructure action. Supporting this central component are the Agent Engineering module and the LLM Toolset, both of which provide the key capabilities required for effective multi-agent orchestration. Specifically, the Agent Engineering component manages the agent lifecycle and coordinates decision-making workflows while integrating several key functions: performance evaluation to assess metrics such as decision accuracy and response time; tracing to provide insights into LLM reasoning and decision pathways, including token-level reasoning chains; monitoring to track orchestration performance metrics like resource utilization and SLA compliance; and versioning to manage system rollbacks and maintain model checkpoints with associated performance histories. The LLM Toolset complements these by offering comprehensive AI capabilities and a communication infrastructure that enable seamless coordination across all agents.

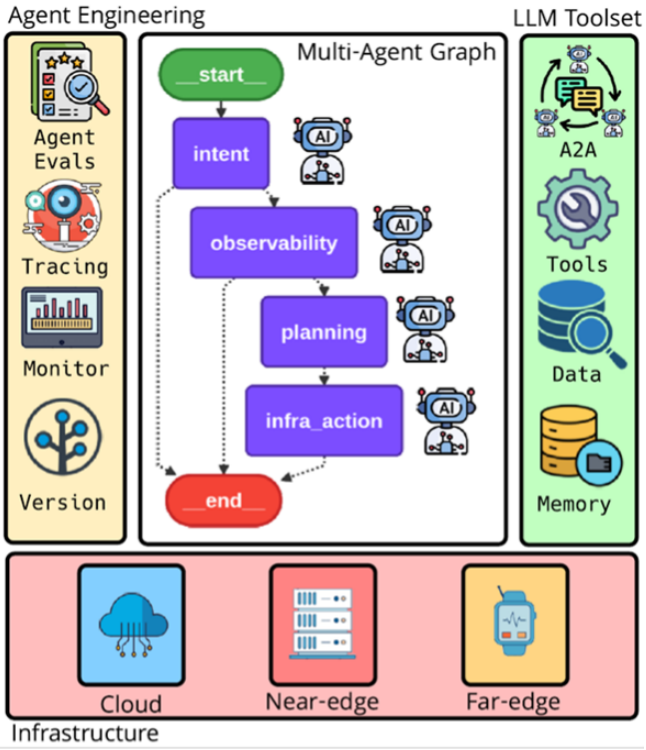


Figure 2: Agentic Orchestration framework

The multi-agent graph is composed of four agents, thus the Intent agent, the planning agent, the observability agent and the infra_action Agent. The Intent Agent acts as the interface between operators and the orchestration system, translating high-level user requirements into structured orchestration objectives and constraints that can be processed by the remaining agents. The Planning Agent transforms user intents into orchestration plans that satisfy service requirements while optimizing resource utilization. It leverages reasoning, historical knowledge, and real-time infrastructure information to generate candidate actions. To improve decision reliability, plans are validated through the ActSimCrit methodology before execution.

The Infrastructure Action Agent executes validated orchestration plans on the underlying infrastructure through platform-specific interfaces and APIs. It is responsible for carrying out deployment, migration, scaling, and lifecycle management actions while adapting to changing infrastructure conditions. The Observability Agent continuously monitors the infrastructure and service environment by analysing telemetry and operational metrics collected from distributed edge and cloud resources. It provides situational awareness and feedback to the Planning and Infrastructure Action Agents, enabling adaptive and proactive orchestration decisions.

To enable intelligent orchestration decisions, the agents leverage Large Language Models (LLMs) to process natural language inputs and generate orchestration actions. For an agent i, the interaction follows a standard input-output pattern, where the agent submits a prompt $p_i(t)$ and receives a response $y_i(t)$, such that $y_i(t) = LLM(p_i(t))$. The prompt $p_i(t)$ contains the current

system state, agent observations, historical context, and task-specific instructions, while the output $y_i(t)$ represents the reasoning outcome generated by the LLM.

Since unstructured LLM outputs are insufficient for infrastructure orchestration tasks that must satisfy strict operational constraints, the framework employs structured outputs validated against dynamically generated schemas. This validation mechanism ensures that orchestration decisions conform to the current infrastructure state and the responsibilities of the corresponding agent.

Specifically, a validation schema $V_i(t)$ is generated for each agent i based on the current system state $u(t)$ and the agent role $role_i$, such that $V_i(t) = Schema(u(t), role_i)$, where $u(t) = [u_1(t), u_2(t), ..., u_m(t)]$ represents the utilisation state of the infrastructure. The resulting schema defines the set of admissible outputs that can be generated by the agent under the current operating conditions. Only orchestration actions satisfying all schema constraints are accepted for execution, thereby ensuring safe, policy-compliant, and infrastructure-aware decision making.

3.3.1.1.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed multi-agent orchestration framework contributes to UNITY-6G AI-native management and orchestration layer by providing autonomous closed-loop operation across distributed cloud, edge domains.

From, KPI persepcetive, the solution contributes KPIs related to service continuity, adaptive resource management, energy-aware orchestration, and service management automation. In particular, the framework is primarily aligned with

- KPI #37: Guarantee 99.99999% service continuity through AI-driven proactive management of cloud-native services.

By continuously monitoring infrastructure conditions and service behaviour, the framework can proactively identify potential performance degradations and trigger orchestration actions such as workload migration, service scaling, resource reallocation, and automated recovery procedures before service disruption occurs

3.3.1.1.2 Preliminary results

To provide an initial assessment of the proposed Agentic AI orchestration framework, a proof-of-concept environment was developed emulating a distributed edge–cloud infrastructure. The evaluation considers the system model illustrated in Figure 3, where orchestration decisions are generated through the interaction of four logical layers. User requirements are expressed as intents, denoted by ζ(t), and are processed by the Agentic Framework following an Intent–Observe–Plan–Act workflow. Candidate orchestration plans are generated through LLM-based reasoning and subsequently validated using dynamic schemas V(t), ensuring that only

actions that are feasible within the current infrastructure context can be executed. The resulting validated actions are forwarded to the Service Orchestrator, which translates them into infrastructure-specific operations while continuously exchanging infrastructure state information u(t) with the agentic layer, thereby enabling closed-loop orchestration and adaptive decision making.

The underlying infrastructure is modelled as a graph G = (N, E), where N = {n_1, n_2, ..., n_m} represents the set of computational nodes and E denotes the communication links interconnecting them. Each node n_i provides a finite set of resources r_i, including CPU, memory, and storage capacity. The services to be orchestrated are represented by the set S = {s_1, s_2, ..., s_j}, where each service s_i is associated with a set of deployment requirements d_i, such as resource demands, placement constraints, and quality-of-service objectives. Service placement is represented through the mapping function $\pi(s_i) = n_k$, which assigns service s_i to infrastructure node n_k. Communication relationships between nodes are represented by $\ell(n_i, n_j)$, capturing the connectivity characteristics of the distributed edge infrastructure, including latency and network reachability. Together, the service set S, infrastructure graph G, resource availability r_i, deployment requirements d_i, placement mapping $\pi(s_i)$, and system state information u(t) define the operational environment over which the agentic orchestration framework performs reasoning, planning, and optimization.

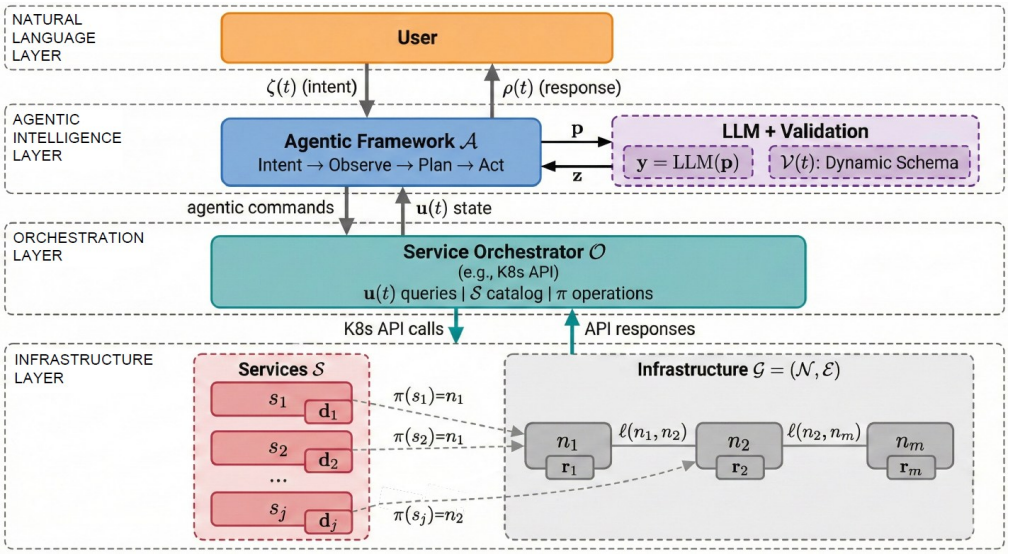


Figure 3 : Agentic Orchestration system model

For the preliminary evaluation, the infrastructure consists of two edge nodes (Edge-0 and Edge-1) hosting four services, namely MainWebApp, DatabaseService, APIService, and Monitoring. The objective is to assess the ability of the proposed framework to autonomously generate valid orchestration decisions under resource constraints, service placement

requirements, and energy-efficiency objectives. Specifically, three system state scenarios are considered as below:

- LPM + Scale (Deadlock) scenario: This scenario starts with Edge-0 using 1.92 out of 4.00 CPU cores for MainWebApp and Monitoring, while Edge-1 uses 0.40 out of 4.00 CPU cores for DatabaseService and APIService. The goal is to put Edge-1 into Low-Power Mode and at the same time scale the APIService to 2.4 CPU cores. This creates a shortage of 0.72 CPU cores, so the orchestrator must find a non-trivial solution such as redistributing services, partially scaling the API service, or splitting services across nodes.
- LPM + Scale scenario: In this scenario, Edge-0 starts with 1.92 out of 8.00 CPU cores allocated to MainWebApp and Monitoring, while Edge-1 uses 0.40 out of 4.00 CPU cores for DatabaseService and APIService. The objective is again to place Edge-1 in Low-Power Mode and scale the APIService to 2.4 CPU cores. However, scaling cannot be carried out on a node that is in Low-Power Mode, so the successful outcome requires migrating workloads before entering Low-Power Mode while still balancing performance and energy-efficiency goals.
- Constrained deployment scenario: This scenario begins with Edge-0 fully occupied, using 4.00 out of 4.00 CPU cores for WebApp, DatabaseService, and APIService, while Edge-1 is empty and has 0.00 out of 6.00 CPU cores in use. The challenge is to deploy a new service requiring 2 CPU cores on Edge-0, even though that target node has no spare capacity. A successful solution therefore requires freeing at least 2 CPU cores on Edge-0, typically by migrating one or more services to Edge-1.

Given the above scenarios, the performance of the agentic pipeline was evaluated in terms of success rate which reflects the ability of the agents to yield the desired predefined state as shown above. Specifically, the success rate is defined as the percentage of orchestration runs that achieve a valid target configuration. After each orchestration attempt, the final infrastructure state produced by the system is compared against a predefined set of acceptable end states. A run is considered successful if the resulting configuration matches any of these acceptable outcomes; otherwise, it is marked as a failure. The success rate is then computed as the ratio of successful runs to the total number of runs, expressed as a percentage.

In Figure 4, the performance across different LLM models serving as the brain of the different agents is shown, demonstrating Deepseek-R1's superiority with a 78.3% success rate, outperforming Llama-4 (73.3%) and significantly outperforming Qwen-3-32B (45%). Notably, Qwen-3-32B's lower performance highlights the critical importance of systematic model evaluation in orchestration systems.

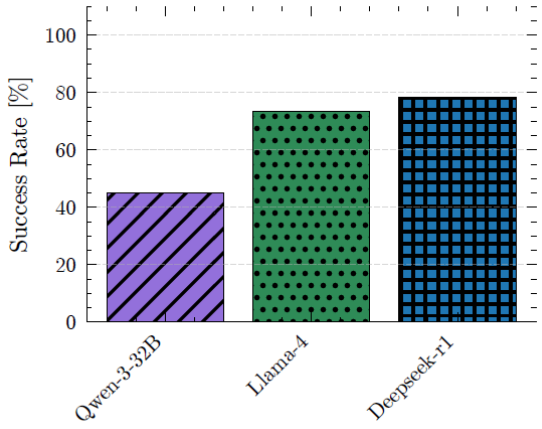


Figure 4: Orchestration success rates for the different models

Figure 5 presents the performance comparison between the proposed multi agent framework and single-agent orchestration approaches in terms of success rate and execution runtime. The results show that the multi-agent approach significantly outperforms the single-agent baseline, achieving a success rate of approximately 78%, compared to only about 22% for the single-agent system. This corresponds to an improvement of roughly 3.6× in successful orchestration decisions.

In terms of runtime, both approaches exhibit comparable performance, with the multi-agent system completing tasks in slightly less time than the single-agent approach. This indicates that the substantial gain in success rate does not come at the cost of increased execution time.

Overall, the figure highlights the effectiveness of the multi-agent architecture in handling complex orchestration scenarios, demonstrating that distributing intelligence across specialized agents leads to more reliable decision-making without introducing additional runtime overhead

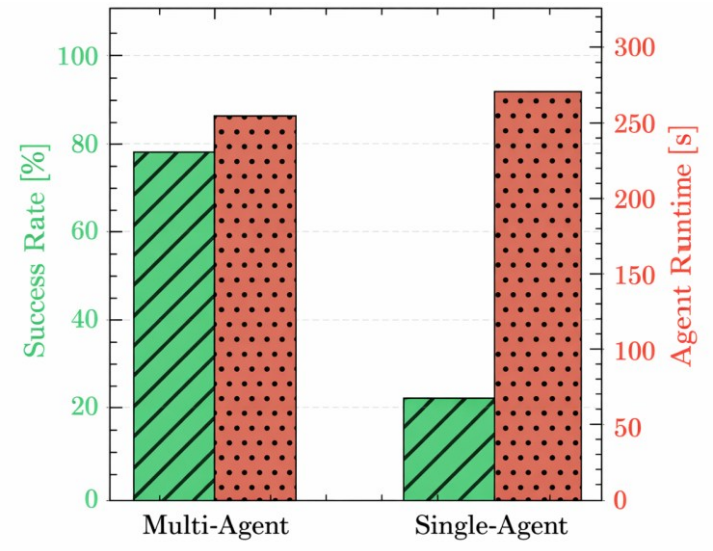


Figure 5: Performance of Multi-Agent and Single-Agent

3.3.1.2 Green orchestration through multi criteria decision analysis

We propose an architecture that bridges this gap by introducing an energy-aware, multi-criteria decision logic within the hierarchical UNITY-6G architecture composed of an IDMO layer, DMO, and IDMs integrated with an energy management system adopting similar principles.

The intelligence of the proposed orchestration architecture does not rely on a single, monolithic algorithm. Instead, it utilizes a distributed ecosystem of algorithms and predictive models divided into two main operational phases: (i) the provisioning and placement phase (multi-criteria decision analysis) and (ii) the runtime adaptation phase (automated closed-loop control). However, before any placement algorithm can execute, the system must establish an accurate algorithmic representation of current and future energy states. This intelligence gathering is delegated to the inter-domain Energy Management System (iEMS) and local Virtual Power Plants (VPPs).

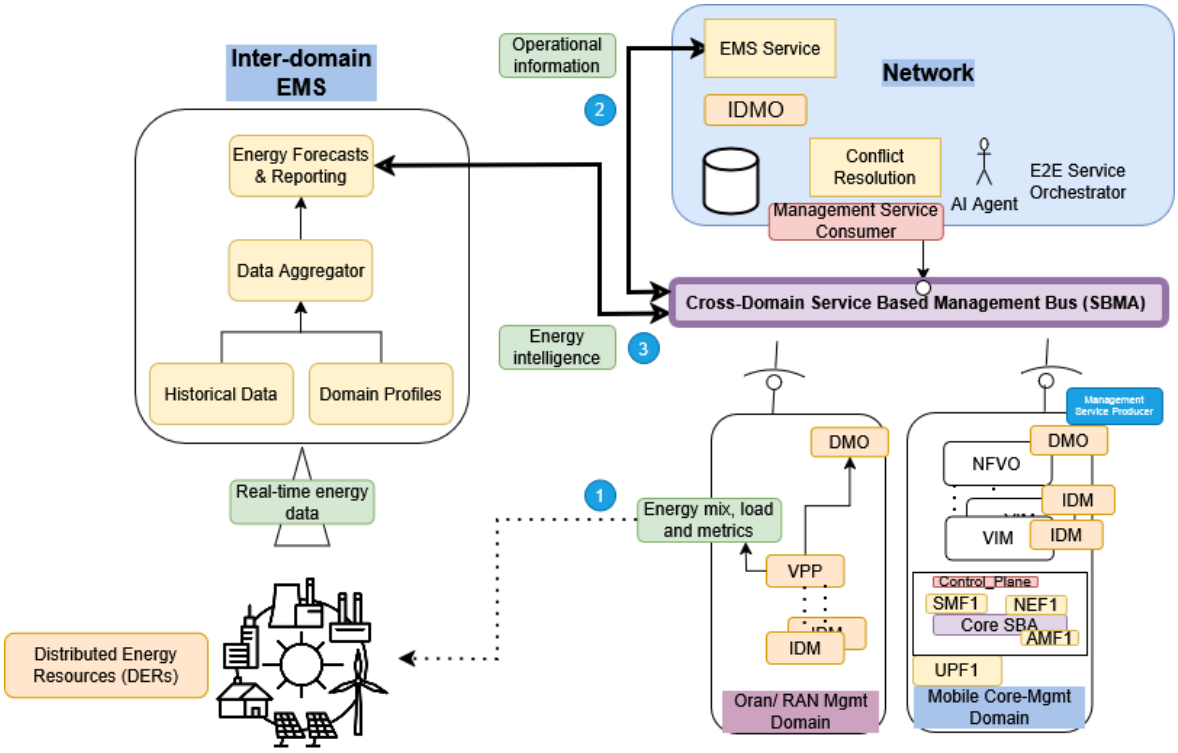


Figure 6: Proposed architecture based on multi criteria decision

Figure 6 illustrates the proposed architecture accordingly. At the local level, VPPs act as data aggregation entities, reporting the collective energy status of all local Distributed Energy Resources, storage systems, and flexible loads associated with the IDMs. The VPP algorithms consolidate this real-time telemetry into a unified representation of the domain's state, capturing net load profiles and localized generation capacity. The inter-domain EMS ingests these real-time data streams alongside extensive historical domain profiles. The EMS may apply advanced forecasting algorithms to generate both deterministic and probabilistic predictions for future decisions regarding the metrics of generation availability (e.g., solar panels attached to a base station), carbon intensity (e.g., how green a site is), and turn it into an indicator for greenness.

The core principle of this algorithm is that it follows the UNITY-6G Architecture within the IDMO's placement engine. When the IDMO receives a high-level E2E service intent (e.g., a request specifying target latency), it must decompose the request and map it to physical infrastructure. During this process, the EMS provides an additional criterion to distinguish candidate selections from an environmental perspective. The overall system evaluates the current percentage of renewable energy, carbon intensity, and the computed green index to prioritize greener domains.

The centerpiece of the orchestration logic is the placement engine embedded within the IDMO. When triggered by a high-level E2E service request, the IDMO initiates an automated workflow spanning Intent Translation, Candidate Discovery, **Multi-Criteria Optimization**, and Distributed Instantiation. The envisioned algorithm is titled the Multi-Criteria Decision Analysis (MCDA) algorithm. The algorithm ranks candidate domains by evaluating a weighted cost function that integrates multiple, often competing, constraints. The multi-criteria logic systematically evaluates candidate locations based on the ordered importance of (i) service capability and availability, (ii) resource sufficiency, (iii) latency and coverage suitability, and (iv) sustainability. Through this weighted evaluation, the algorithm inherently executes intelligent trade-offs. For example, it may securely route a core network function to a geographically distant domain (incurring higher, but still acceptable, latency) if that domain currently possesses a robust surplus of renewable energy.

To demonstrate the operational flow, Figure 7 illustrates how the prototype models the initial E2E service request from a network provider, mapping out the semantic requirements and constraint parameters before the MCDA engine begins its evaluation.

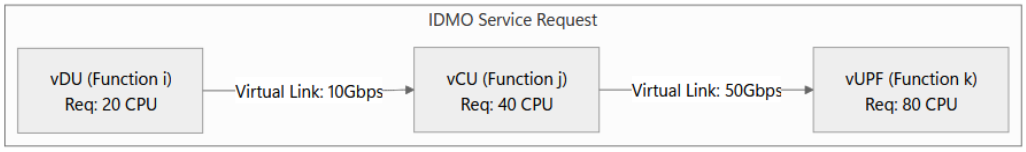


Figure 7: Service request for IDMO

Following the successful execution of the MCDA algorithm and the subsequent Distributed Instantiation phase, the network functions are pushed to their respective domains. Figure 8 shows the final deployed architecture, detailing how the instantiated service components are distributed across the optimal candidate domains, interconnected via secure VPN tunnels, and placed under the continuous supervision of the local DMO entities.

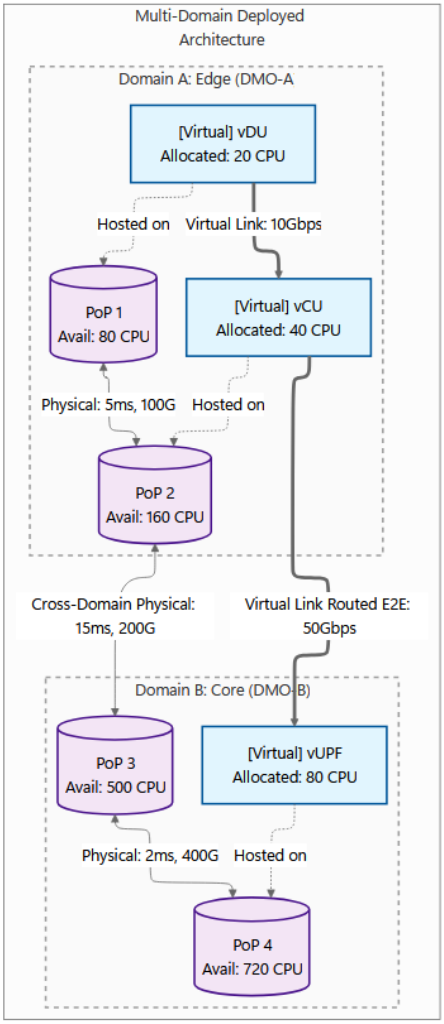


Figure 8: Sample multi-domain architecture

3.3.1.2.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed energy awareness architecture is based upon the hierarchical, multi-domain framework established by the UNITY-6G project. This energy intelligence operates via the continuous exchange of telemetry over the SBMA, driving automated sustainability actions and allowing the IDMO to make intelligent placement decisions.

Within PoCs, this architecture directly addresses several energy-related KPIs. It targets increasing the usage of electricity from renewable sources by up to 20% (KPI#14) via the AI/ML subsystem and native DMO policy enforcement for dynamic resource scaling and workload migration. Furthermore, the framework aims to save up to 25% in energy costs by significantly reducing CNF consumption (KPI#5) and cutting radio link power consumption by up to 30% per link (KPI#16).

Ultimately, the PoCs prove that these energy optimizations can be achieved without sacrificing network reliability, ensuring cross-domain end-to-end latency remains strictly below a required

threshold (KPI#7). This validates the core premise of the UNITY-6G architecture: that computational flexibility, driven by hierarchical orchestration (IDMO/DMO) and intelligent decision-making, can successfully manage the performance-to-sustainability trade-off.

3.3.1.3 Agentic Intelligence for Non-Terrestrial Networks

The proposed AI framework introduces an agentic intelligence architecture for NTNs, where each satellite beam behaves as an autonomous AI agent capable of perception, reasoning, prediction, anomaly detection, and resource allocation. Unlike conventional NTN optimization systems that rely on static thresholds or centralized orchestration, the proposed solution combines Federated Kolmogorov-Arnold Networks (Fed-KAN), LLMs, ReAct reasoning, and episodic memory into a unified autonomous control loop.

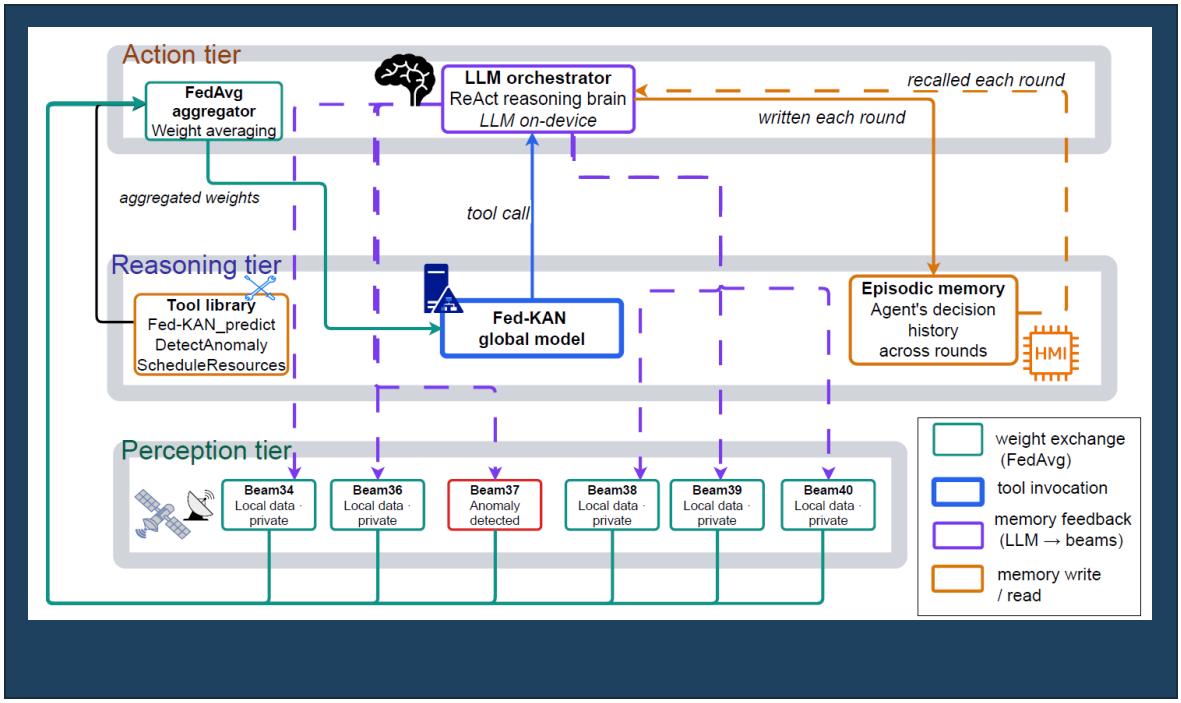


Figure 9: The overall system architecture of Agentic AI-enhanced resource management framework with federated-KAN

The architecture in Figure 9 is composed of three tightly integrated functional layers: perception, reasoning, and action. In the perception layer, each beam agent continuously monitors local traffic conditions including Return and Forward throughput measurements together with cyclic temporal features such as hour-of-day and day-of-week encodings. These inputs are processed locally to preserve privacy and avoid transmission of raw user traffic data. At the core of the framework lies the reasoning layer, powered by a locally hosted Llama 3 large language model running through the Ollama inference runtime. The LLM orchestrator acting rather than a simple chatbot, acts as network orchestrator so that it can interact with network decision. The network orchestrator within LLM orchestrator later executes a ReAct (Reason + Act) loop where the model first observes the current traffic state, retrieves previous

reasoning traces from episodic memory, selects the appropriate analytical tool, interprets the outputs, and finally commits a resource management decision. The available tools include:

- Fed-KAN traffic forecasting,
- Isolation Forest anomaly detection,
- Resource scheduling and allocation.

The Fed-KAN model itself extends conventional FL by replacing standard activation functions with learnable spline-based KAN functions. This improves interpretability and allows the model to better adapt to non-stationary satellite traffic distributions. The model is collaboratively trained across multiple satellite beams using the FedAvg aggregation mechanism while ensuring that raw traffic samples remain local to each beam.

A major innovation of the framework is the transformation of Fed-KAN from a passive prediction model into a callable AI tool within an autonomous reasoning loop. Existing FL-based NTN approaches typically stop after model training and apply predefined threshold rules externally. In contrast, the proposed framework embeds the forecasting model directly inside the decision-making pipeline. The LLM dynamically decides when to prioritize forecasting, anomaly detection, or both, before generating an explainable resource allocation action such as:

- Scale-up,
- Maintain,
- Scale-down,
- Alert anomaly.

Another key innovation is the episodic memory mechanism. Each beam maintains a local memory buffer containing previous observations, reasoning traces, selected tools, and committed actions. During the next reasoning cycle, the LLM retrieves recent memory entries and uses them as contextual knowledge. This provides continual adaptation without any gradient updates or retraining. The mechanism is computationally lightweight and particularly suitable for NTN environments where on-board processing resources are constrained. The proposed framework contributes beyond the current SoA in several dimensions:

- *Integration of Agentic AI with Federated Learning:* Current NTN federated learning solutions focus mainly on prediction accuracy. The proposed framework introduces autonomous reasoning and action orchestration through LLM agents.

- *Explainable Autonomous Decision-Making:* Traditional AI models provide numerical outputs only. The proposed solution generates human-readable chain-of-thought reasoning traces that can be audited by network operators.
- *Privacy-Preserving Distributed Intelligence:* All raw traffic data remains local to satellite beams. Only model weights are exchanged through FedAvg, fully aligning with privacy-preserving 6G NTN principles.
- *Continual Learning Through Episodic Memory:* Unlike conventional FL systems requiring retraining, the proposed framework adapts continuously using memory-enhanced reasoning without modifying model weights.
- *NTN-Specific Safety-Constrained ReAct Loop:* The architecture extends the original ReAct framework by enforcing deterministic tool-chain completion and machine-parseable JSON outputs, ensuring operational safety in satellite networks.
- *Reduced Dependence on Cloud Infrastructure:* The use of local Llama 3 inference eliminates dependency on external cloud APIs, enabling operation in latency-sensitive or spectrum-constrained NTN deployments.

The framework establishes a new paradigm for autonomous NTN management where forecasting, reasoning, memory, and action are unified within a fully decentralized AI-native architecture.

3.3.1.3.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

NTN Agentic AI with Fed-KAN algorithm is most relevant to KPIs focused on *distributed AI, NTN reliability, resource orchestration, energy efficiency, multi-tenant scalability, and real-time control*. The algorithm maps to the UNITY-6G architecture as an *AI-native autonomous NTN orchestration layer*. Each NTN beam acts as a distributed intelligent agent with local perception, local reasoning, federated learning, episodic memory, anomaly detection, and autonomous resource allocation. The following KPIs are considered:

- *KPI #17* - Reduce CAPEX and OPEX with shared resources: The use of federated learning enables distributed NTN beams to collaboratively train a shared AI model without centralizing raw traffic data. This reduces the need for duplicated centralized monitoring and optimization infrastructure. Contribution is shared AI model, decentralized orchestration, lower operational complexity.
- *KPI #21* - Reduce performance penalty due to conflicts by 20%*:* The LLM-based reasoning layer can interpret multiple signals, including traffic forecasts, anomaly scores, and memory traces. This allows the agent to distinguish between normal congestion, abnormal behavior,

and conflicting resource situations before committing an action. Contribution is explainable conflict-aware decision support through agentic reasoning and anomaly detection.

3.3.1.3.2 Preliminary results

The proposed framework was evaluated using a real residential satellite traffic dataset containing six NTN beam footprints and 744 hourly traffic samples collected during January 2023. The evaluation considered both federated learning performance and autonomous agentic decision-making capability. The Fed-KAN architecture was compared against a baseline Federated Multi-Layer Perceptron (Fed-MLP). Results as shown in Figure 10 demonstrated that the proposed framework achieved significantly better prediction performance:

- Fed-KAN Mean Squared Error (MSE): 7.972
- Fed-MLP Mean Squared Error (MSE): 9.772
- Overall improvement: 18.4% lower MSE

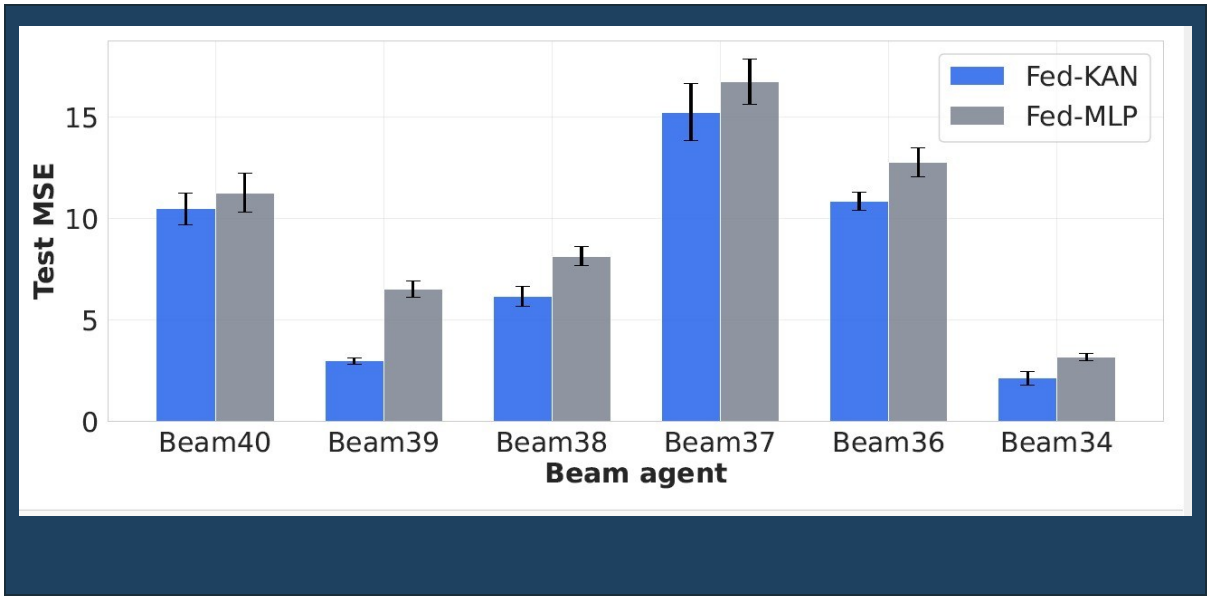


Figure 10: Per-agent test MSE (mean ± 1 std, 5 seeds)

The lower variance observed across multiple experimental seeds also demonstrated that the spline-based KAN representation generalizes more consistently across heterogeneous beam traffic conditions. The autonomous LLM reasoning engine generated context-aware resource management decisions across all beam agents. Four possible actions were supported:

- Scale-up,
- Maintain,
- Scale-down,

- Alert anomaly.

Experimental results showed 25 scale-up decisions (83.3%) and 5 anomaly alerts (16.7%). One beam (Beam37) exhibited anomalous traffic behavior detected through the Isolation Forest mechanism. The LLM correctly escalated this beam to an alert anomaly state rather than applying a standard resource scaling decision. This demonstrates that the reasoning engine can combine forecasting outputs and anomaly indicators to generate differentiated operational responses.

Another important result concerns explainability. Unlike black-box AI approaches, the proposed framework generated explicit chain-of-thought reasoning traces for every decision. Example observations included:

- Identification of asymmetric forward/return traffic patterns,
- Detection of fluctuating traffic conditions,
- Recognition of potential congestion trends,
- Justification of anomaly escalation decisions.

The episodic memory mechanism also demonstrated the feasibility of continual adaptation without retraining. By storing previous reasoning traces locally, the system was able to provide contextual continuity across federated rounds while preserving privacy.

From an implementation perspective, all experiments were executed using locally hosted Llama 3 inference with no dependency on external cloud APIs. This validates the feasibility of deploying agentic AI directly within NTN environments. The preliminary results validate the effectiveness of combining:

- Federated learning,
- KAN-based forecasting,
- Agentic LLM orchestration,
- Episodic memory,
- Explainable reasoning,

within a unified autonomous NTN management framework. The results indicate strong potential for future large-scale deployment across multi-orbit 6G NTN infrastructures and provide a solid baseline for future UNITY-6G demonstrations and proof-of-concept activities.

3.3.1.4 Intent-Based Transport Configuration

The proposed concept addresses a fundamental difficulty that is expected to intensify in 6G transport environments: the gap between high-level service objectives and the low-level configuration actions required to enforce them on heterogeneous transport infrastructure. Future 6G systems will combine disaggregated radio functions, distributed computing resources, slice-specific quality requirements, and a mixture of wired, microwave, millimetre-wave, and fronthaul or midhaul connectivity. In such an environment, it is no longer operationally efficient to expect an orchestrator, a domain controller, or a human operator to manually translate each service request into a sequence of vendor-specific parameter settings. The central idea of the present solution is therefore to introduce an intent-driven transport control mechanism in which the requesting entity expresses the desired operational outcome, whereas the system itself determines the most appropriate realizable transport configuration.

The need for such a solution arises from the fact that transport networks are no longer passive forwarding substrates. They actively determine whether stringent service targets can be satisfied, particularly for applications that impose tight latency, reliability, synchronization, or availability constraints.

The proposed approach is based on the principle that an intent should capture what must be achieved rather than prescribed in advance how the network should be configured. In this framework, an intent may express a required capacity level, a resilience target, a latency guardrail, a synchronization constraint, an energy-efficiency objective, or a traffic differentiation policy aligned with a service or network slice. The purpose of the intent solution is to interpret this statement, identify the relevant transport scope, determine which realizations are feasible for the available infrastructure, and generate a consistent realization plan. In this sense, the approach creates an intermediate reasoning layer between service-facing orchestration logic and transport control plane. It preserves abstraction at the northbound side while remaining tightly grounded in the actual capabilities of the underlying transport domain.

A second key aspect of the approach is that the solution does not treat all requests as simple one-shot translations. Instead, it treats transport decision making as an evidence-based process. The intent is parsed, normalized, and matched against contextual information such as topology structure, device roles, supported features, and recent telemetry. The resulting decision is not merely the product of a static rule engine; it is a structured process that can combine deterministic validation, hardware-aware reasoning, optimization logic, simulation results, and, when appropriate, advanced AI-assisted interpretation. In this way, the proposed concept is suitable both for formally specified requests generated by higher-layer systems and for more flexible operational requests expressed in descriptive terms by engineering personnel.

The overall ambition of the solution is therefore not only to automate configuration generation, but also to improve the technical quality of transport-domain decisions. By exposing transport management as an intent-driven capability, the system creates a mechanism through which high-level objectives can be transformed into safe, explainable, and infrastructure-aware actions. This is particularly valuable in 6G contexts where the network must support multiple domains, where transport resources are diverse and constrained, and where operational decisions should increasingly be derived from goals, policies, and measured network conditions rather than from manual low-level configuration procedures.

3.3.1.4.1 System Model and Solution

The system model underlying the solution assumes a layered transport-control architecture in which a requesting entity, such as an orchestration function, a service adaptation module, a digital twin component, or an operator-facing interface, submits a transport intent to an intermediate reasoning engine. This engine has visibility over the relevant transport topology, access to a structured knowledge base describing the capabilities and constraints of the managed infrastructure, and connectivity to the downstream transport control mechanisms that ultimately apply configuration actions. The managed domain may contain several classes of transport elements with different functional roles, including aggregation nodes, midhaul hubs, and strict point-to-point tails, each of which may expose a different set of admissible actions and performance envelopes. The system model therefore assumes heterogeneity by design and does not rely on uniform support across all infrastructure elements.

Within this model, the solution is organized around a sequence of functional stages. The first stage receives and interprets the intent, resolving the operational scope and identifying the primary objective category. The second stage validates the request against topology constraints, policy constraints, and hardware feasibility. The third stage constructs one or more candidate realization strategies and evaluates them according to the requested objective and the available evidence. The fourth stage generates the resulting realization plan as a set of transport-domain actions suitable for enforcement by the underlying control framework. The fifth stage verifies, through telemetry and state observation, whether the applied configuration continues to satisfy the originally requested objective. The entire solution can therefore be understood as a closed-loop control system in which the intent constitutes the desired state, the realization plan constitutes the control action, and the telemetry constitutes the feedback signal used to detect alignment or drift.

A central concept in the solution is the distinction between skills and tools. A skill is defined as a reusable technical capability of the intent engine, namely a bounded reasoning function that can be invoked to contribute to decision formation. Examples include intent parsing, semantic

classification, topology-aware scope resolution, capability matching, multi-constraint optimization, conflict detection, fallback selection, and closed-loop assurance. These are described as skills because they represent operational competencies of the system rather than individual software modules. They define what the system is able to do when it must infer the correct course of action from an objective expressed at a higher level of abstraction. In technical terms, a skill maps available evidence, constraints, and goals into a decision-relevant output that can be consumed by the subsequent reasoning stages.

The tools of the solution are the concrete algorithmic, analytical, and computational mechanisms through which these skills are executed. Depending on the situation, the system may use deterministic rule-based validators, hardware knowledge-base queries, search and optimization procedures, policy filters, constraint solvers, traffic and link-state analytics, telemetry-driven reconciliation logic, or machine-learning and AI-based components. In more advanced use, the tool set may also include simulation functions and what-if evaluation procedures that allow the system to assess candidate actions before selecting the final realization strategy. Accordingly, a skill such as optimization is not identified with a single algorithm; rather, it is treated as an operational function that may invoke one or more tools, including heuristics, exact optimization routines, predictive models, or simulation-assisted evaluation, to determine the best admissible transport action under the prevailing conditions.

This distinction is important because it allows the system to remain extensible while preserving a stable conceptual architecture. Skills define the persistent reasoning functions of the intent engine, whereas tools can evolve over time as the solution incorporates richer models, improved algorithms, or domain-specific AI and ML components. For example, a capability-matching skill may initially rely on static hardware profiles and deterministic constraints but later be augmented by predictive models that estimate the likelihood of successful realization under changing radio conditions. Similarly, a decision skill responsible for resilience optimization may invoke simulation-based tools to compare alternative protection strategies before recommending one of them for enforcement. The architecture (shown in Figure 11) therefore supports progressive enrichment without changing the fundamental interaction model presented to higher layers.

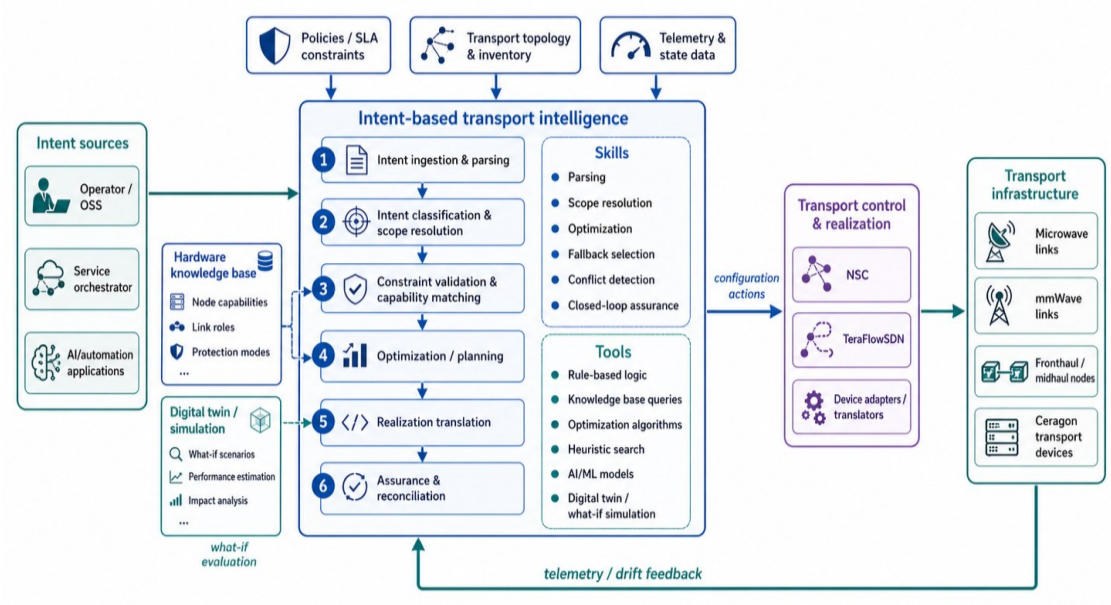


Figure 11: Proposed architecture for the intent based transport configuration

3.3.1.4.2 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The following KPIs are considered:

- KPI#6. Decrease service creation and termination time: Service creation is accelerated because the required transport resources, QoS treatment, protection configuration, and admissible path realization are determined and activated as one coordinated lifecycle operation. Likewise, when a service is withdrawn, modified, or expires, the corresponding transport resources and policies can be consistently released or reconfigured without prolonged dependency resolution across the transport domain. The result is a shorter end-to-end service activation and deactivation cycle, while maintaining compliance with capacity, latency, resilience, and isolation constraints.
- KPI#7. Cross-domain (cellular – IEEE 802.11) end-to-end latency smaller than 20 ms*:* It decomposes this target across the cellular access, transport, interworking, and Wi-Fi segments; admits only feasible service realizations; selects low-delay paths and local edge breakout where available; and continuously monitors the aggregate latency. When the latency budget is at risk, the system can re-steer traffic, adjust QoS treatment, or re-optimize the transport path to preserve compliance.
- KPI#8. Cross-domain jitter of max of 2 ms, with less than 10% of the packets missing deadline: It allocates jitter and delay budgets across the cellular, transport, interworking, and IEEE 802.11 segments; admits only feasible service paths; and applies suitable QoS, scheduling, and traffic-steering policies. Continuous telemetry verifies both jitter and

deadline-miss ratio, enabling corrective actions such as path re-selection, priority adjustment, or load rebalancing when either target is at risk.

- KPI#17. Reduce CAPEX and OPEX with shared resources: The solution reduces CAPEX and OPEX by treating transport capacity, protection resources, and edge-connectivity functions as shared pools that can be allocated dynamically across services and slices. This improves utilization, reduces the need for dedicated overprovisioned resources per service, and allows inactive or terminated services to release capacity for reuse. In operational terms, automated intent fulfilment and assurance also reduce the cost of repeated configuration, reconfiguration, and fault-handling activities across the shared infrastructure.
- KPI#20. Increase link utilization from 40% to 80%: The solution can increase average link utilization by replacing static, service-dedicated capacity reservations with telemetry-driven shared-resource allocation. It identifies underutilized links and available protection capacity, steers eligible traffic toward them, and continuously re-optimizes paths as demand changes. Capacity released by inactive, modified, or terminated services is returned to the common resource pool and reused by other services or slices. The target is achieved through controlled statistical multiplexing and policy-constrained traffic engineering, while retaining latency, resilience, and isolation margins so that higher utilization does not create unacceptable congestion or SLA degradation.
- KPI#21. Reduce the performance penalty due to conflicts by 20% by means of DLT, hierarchical AI algorithm and policy-based conflict resolution schemes: Maintaining a shared, tamper-evident view of active intents, resource commitments, and policy decisions, reducing conflicts caused by inconsistent or stale domain information. Hierarchical AI resolves routine conflicts locally and escalates only cross-domain or high-impact cases, while the policy engine applies predefined priorities, resource limits, and SLA-precedence rules to select a compliant outcome. This reduces repeated reconfiguration, resource contention, and service degradation.
- KPI#38. Reduce Operation expenditure (OPEX) by 30% due to the automation of service management and increase in link utilization (via planning annual frequency licensing): Automated intent fulfilment, assurance, modification, and termination reduce recurring operational effort for provisioning, monitoring, fault handling, and reconfiguration. In parallel, demand-aware capacity and frequency planning consolidates traffic onto underutilized links and frequency assignments, enabling resources to be reused and reducing the need to acquire or retain additional annually licensed spectrum capacity.
- KPI#43. Increase integrated network energy efficiency by a factor of 5 through distributed AI techniques, semantic communications, renewable energy usage, proactive strategies for

resource allocation, energy aware CNF placement and task offloading, and energy aware scheduling of AI training and inference: Selecting realizations that minimize energy consumption while satisfying capacity, latency, resilience, and synchronization constraints. The intent engine can steer traffic over energy-efficient paths, consolidate flows onto lightly loaded links, place unused transport interfaces or nodes into low-power states, and select routes or sites with greater renewable-energy availability. Distributed reasoning enables these decisions to be made close to the managed transport domain, while proactive telemetry-based planning avoids unnecessary link activation and overprovisioning. Transport-aware placement and offloading decisions may also be supported where they reduce the network energy required to reach edge or cloud functions.

The intent engine is running in the network operation center, monitoring the entire network. The analytical engine, comprising of an agentic skill based LLM service, interprets the data and the user intentions to provide guidance. The decision engine translates that to needed operations across the network, while the actuator is sending the instructions to the nodes. The actuations change the network setting, updating the monitored data. This is a closed loop tuple-based agent in charge of continues “zero-touch” optimizations of the transport network.

3.3.1.5 SLO-based Proactive Autonomous Network Orchestration

The rapid evolution of modern transport networks towards dynamic, programmable, and multi-tenant environments has introduced levels of complexity in network management and service provisioning. The current demand for services has significantly increased the demand for stringent SLOs, particularly for mission-critical and latency-sensitive services. In such contexts, ensuring deterministic performance across heterogeneous and distributed infrastructures remains a challenging task.

Agentic AI systems are capable of proactive and predictive decision-making, continuously learning from historical and real-time data, and executing autonomous control actions without the need for continuous human supervision. In contrast to static rule-based systems, these intelligent agents can anticipate potential failures, congestion events, or resource contention situations, and proactively adapt network behavior to mitigate adverse effects before SLO violations occur.

The key innovation of the proposed contribution lies not only in automating network management tasks, but in enabling proactive, predictive, and fully autonomous orchestration across the network infrastructure through distributed intelligence and coordinated multi-agent systems. This approach extends beyond centralized control paradigms by embedding intelligence across multiple layers of the edge-cloud continuum, thereby enabling scalable, resilient, and context-aware resource management.

In multi-tenant transport networks, operators provision differentiated network slices tailored to the specific requirements of diverse services and vertical applications. These slices are associated with strict SLOs that must be maintained throughout the service lifecycle. However, the dynamic nature of network conditions introduces multiple factors that may lead to performance degradation and SLO violations.

Such degradations may arise from a variety of causes, including but not limited to link congestion, physical layer impairments, node or link failures, sudden traffic surges, or topology changes due to dynamic resource provisioning. These events can significantly impact KPIs such as latency, jitter, packet loss, and throughput, potentially compromising the quality of experience for end users.

The fundamental challenge addressed in this contribution is therefore the design of a proactive and autonomous algorithm capable of

- Anticipating potential SLO violations before they occur
- Dynamically enforcing corrective actions on the network to prevent service degradation. This requires the integration of an autonomous AI agent for proactive network resource management

The overall workflow is structured into two tightly coupled phases: (i) monitoring, forecasting, and anomaly detection, and (ii) autonomous mitigation and control. These phases are supported by a distributed knowledge plane that integrates multiple data sources, including historical telemetry repositories, real-time monitoring systems, network inventory databases, and topology information.

The first phase of the proposed system focuses on continuous monitoring and predictive analysis of network conditions. The network infrastructure is instrumented with telemetry mechanisms that collect performance indicators, including latency, jitter, packet loss, and link utilization, across all relevant network segments.

Analytics modules process this telemetry in real time, complemented by historical data analysis to identify trends and deviations from normal behavior. Rather than relying solely on static thresholds, the system employs forecasting and anomaly detection algorithms capable of capturing temporal dynamics and context-dependent patterns. Indicative signals of potential SLO violations include increases in latency, rising jitter levels, abnormal packet loss rates, and link utilization approaching saturation. These indicators are analyzed in combination to detect early warning signs of degradation. When the predicted risk exceeds predefined confidence thresholds, proactive alarms are generated, triggering the autonomous mitigation phase.

Upon detection of a potential SLO violation, the system activates an autonomous AI agent responsible for executing corrective actions aimed at preserving service guarantees.

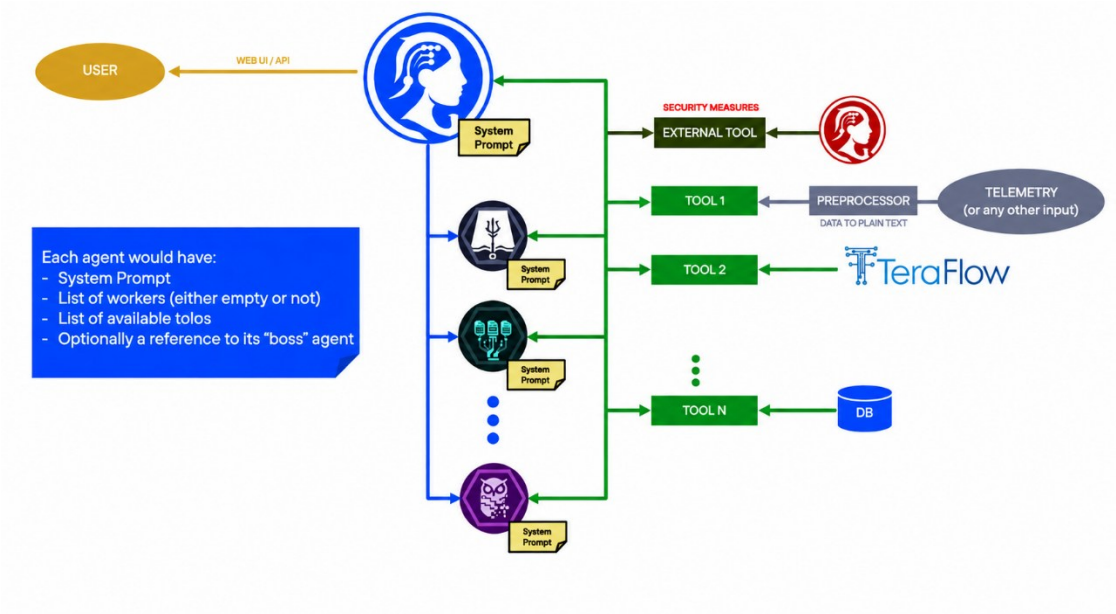


Figure 12: AI Agent architecture

In this system (Figure 12), a manager agent receives some input (either a user prompt, a certain API call, etc), and is capable of plan, organize and manage a list of subagents and produce a response with the help of one or more tools. On top of that, the above architecture has been designed to ensure maintainability and modularity, as each component is decoupled so that it can be easily replaced if needed.

The distributed nature of the architecture enables multiple specialized sub-agents to collaborate, each interacting with different data sources and control entities. These include interfaces to real-time topology information, active traffic data, network inventory systems, and orchestration platforms. Through coordinated decision-making, the multi-agent system ensures consistent and efficient network-wide optimization.

The mitigation strategies implemented by the agents are context-aware and are set to three kinds:

- Request a Path Computation Engine to compute an alternative path that satisfies the slice requirements
- Increase the bandwidth allocated to the affected slice, if feasible
- Redistributing lower-priority traffic flows away from congested paths

This autonomous closed-loop control paradigm eliminates the need for manual intervention in routine operational scenarios, significantly reducing reaction times and improving the robustness of network operations.

3.3.1.5.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

This algorithm operates in the Transport Management Domain as a submodule of the Domain Management Orchestrator, the Network Slice Controller

It is mapped with the following KPIs:

- KPI#19. Reduce time to manage network resources and management overhead and the reaction time by taking proactive actions in the network (time from identification to resolution via appropriate reconfigurations) with integrated design
- KPI#37. Guarantee 99.99999% service continuity through AI-driven proactive management of cloud-native services.

3.3.1.6 The aeriOS framework

The objective of this activity is centered on the implementation and adaptation of aeriOS framework, a next-generation Meta-OS specifically designed to orchestrate the hyper-distributed IoT-edge-cloud continuum. In the context of 6G, which requires the seamless integration of TN and NTN, aeriOS provides the essential abstraction layer that transforms fragmented hardware resources into a unified, programmable execution fabric. The framework is built on a peer-to-peer (P2P) decentralized topology where the continuum is organized into autonomous aeriOS Domains. Each domain is a self-sufficient administrative and technical entity that can operate independently or join a global federation as a peer, thereby eliminating the single-point-of-failure risks of centralized cloud controllers and minimizing the signaling overhead required for real-time orchestration at the extreme edge. The operational intelligence of aeriOS is realized through a multi-level orchestration architecture that decouples global strategic placement from local execution management. The primary cognitive engine is the HLO, which functions as the "brain" of the domain. The HLO's lifecycle begins with the ingestion of an Intention Blueprint, a high-level, goal-oriented manifest that specifies the desired outcome (e.g., "deploy an AI vision service with sub-10ms latency") rather than specific technical configurations. The HLO utilizes the HLO Data Aggregator to query the Data Fabric for real-time infrastructure state and then passes this information to the HLO Allocation Engine. This engine employs the ORIENT approach, a sophisticated AI model that leverages D3QL and GNN to solve the PIRA problem (Placement, Instance Assignment, Request Prioritization, and Allocation). Once a strategic decision is made, the HLO Deployment Engine generates a Decision Blueprint, which is then communicated to the LLO. The LLO, residing closer to the

infrastructure, translates these decisions into Implementation Blueprints, typically formatted as Kubernetes Custom Resources, to be executed on specific Infrastructure Elements (IEs) [72][73].

In terms of technical Layers and functional foundations, the aeriOS architecture is underpinned by three foundational "Fabrics" that ensure connectivity, data integrity, and autonomous operation:

- *The Data Fabric:* This layer provides the "contextual intelligence" for the entire continuum. It is built upon the ETSI NGSI-LD standard and utilizes the Orion-LD Context Broker to manage a real-time knowledge graph of all entities. The Data Product Manager component within this fabric ensures that telemetry from edge sensors and network links is processed and made available to AI consumers (like the HLO) in a standardized format, enabling "context-aware" orchestration.
- *Smart Networking Layer:* To ensure secure and high-performance communication across administrative domains, aeriOS implements a programmable networking stack. This includes eBPF-based (Cilium) packet management for low-latency intra-domain service meshes and WireGuard-based overlays for secure inter-domain connectivity. A software-defined control plane, powered by the ONOS controller using the OpenFlow protocol, allows the Meta-OS to dynamically configure network paths, ensuring that traffic isolation and Quality of Service (QoS) are maintained even during complex workload migrations [72][74].
- *Self- Capabilities:* Aiming for the "zero-touch" management paradigm of 6G, aeriOS includes a dedicated layer for Self-configuration, Self-healing, and Self-optimization. This is implemented via a lightweight json-rules-engine that monitors system telemetry (e.g., using PowerTOP for energy or psutil for CPU) and triggers autonomous recovery or re-orchestration actions if performance thresholds are breached or if a node becomes disconnected.

An aeriOS Domain constitutes a complete aeriOS runtime environment, formed by at least one or more IEs. There are two prerequisites that must be met for a set of connected compute and network resources to qualify as an aeriOS domain. The first prerequisite is that these compute resources are integrated as IEs, as described earlier. This integration ensures that they can support workload execution and provide a minimum set of aeriOS integration capabilities, such as manageable network functionality. These IEs act as the fundamental units within the domain, enabling the deployment and execution of containerized applications and services. The second prerequisite is the deployment of a core set of basic aeriOS services on top of these IEs. This means that an aeriOS domain can be defined as a group of IEs that share the same set and instance of basic aeriOS services, ensuring a cohesive and functional execution

environment [72]-[74]. The basic aeriOS services running within each domain are distilled from the previous discussions and include:

- Federation service, provided by the federator component enables the bidirectional exchange of information with the other domains of the continuum. Exchange of information regarding the status of domain and IEs facilitates resource sharing, workload distribution, and interoperability. It is responsible for managing subscriptions, and registrations to other aeriOS domains which are instrumented to discover resources across the continuum. It is built as a set of coworking microservices offered by aeriOS fabrics.
- Orchestration service which is crucial for managing the deployment of IoT services based on user defined intentions. This service translates user requests into actual deployment instructions and execute them on IEs. This orchestration process operates at two levels: the high level, which receives constraints and queries the continuum and handles decision-making complexities with the support of AI-enriched decision support systems, and the low-level which has the actual access to underlying IEs and translates decision to implementation directives. Based on the federation services, orchestration service within each domain acknowledges the state of IEs across the continuum and can produce decisions including remote domains resources.
- Data Fabric services facilitate seamless data access, for all aeriOS consumers. This ensures that data is readily available and accessible across the entire aeriOS ecosystem, supporting efficient and informed decision-making. Raw data concerning all aspects of aeriOS, including domain status and IoT applications, are ingested and transformed into a common, interpretable format. This standardized information forms the substrate for the federation service, which is responsible for sharing and propagating this data across the entire continuum.
- Cybersecurity services which enforce authentication, authorization, and access control policies based on roles and identities. These services ensure secure access to all domain resources and validate requests in close coordination with the Entrypoint domain's policies and users' registries. By integrating robust security protocols, they maintain the integrity and confidentiality of data and services across the aeriOS continuum.

Additionally, a range of aeriOS services integrated within each domain provide an intelligence layer to the aeriOS Meta-OS. These intelligent services enhance the system's ability to make informed decisions, optimize resource allocation, and improve overall operational efficiency.

- AI decision support services, which leverage data retrieved by the Data Fabric to provide critical input to the orchestrator. These services analyse and interpret the data to

recommend optimal workload placements, whether locally within the current domain or across other domains, ensuring efficient and effective resource utilization [75][76].

- Trust management services, which ensure the integrity and reliability of interactions within the aeriOS ecosystem. These services validate and monitor the trustworthiness of IEs within each domain and provide insights to the decision engine regarding the reliability of candidate hosting resources [73][74].
- Embedded analytic tools, which provide real-time data analysis and insights directly within the aeriOS ecosystem. These services enable continuous monitoring, assessment, and optimization of system performance and IoT applications, empowering stakeholders with actionable intelligence to enhance decision-making and operational efficiency.

Finally, each aeriOS domain, in addition to these core services, exposes a comprehensive and efficient API to facilitate communication with stakeholders, agents, and other domains. Figure 13 provides a comprehensive diagram reflecting the aeriOS architecture. This diagram illustrates the formation of the continuum through the seamless federation of aeriOS domains. It highlights the operations within each domain and the unified layer provided by the aeriOS Meta-OS, offering a cohesive environment for IoT developers [76].

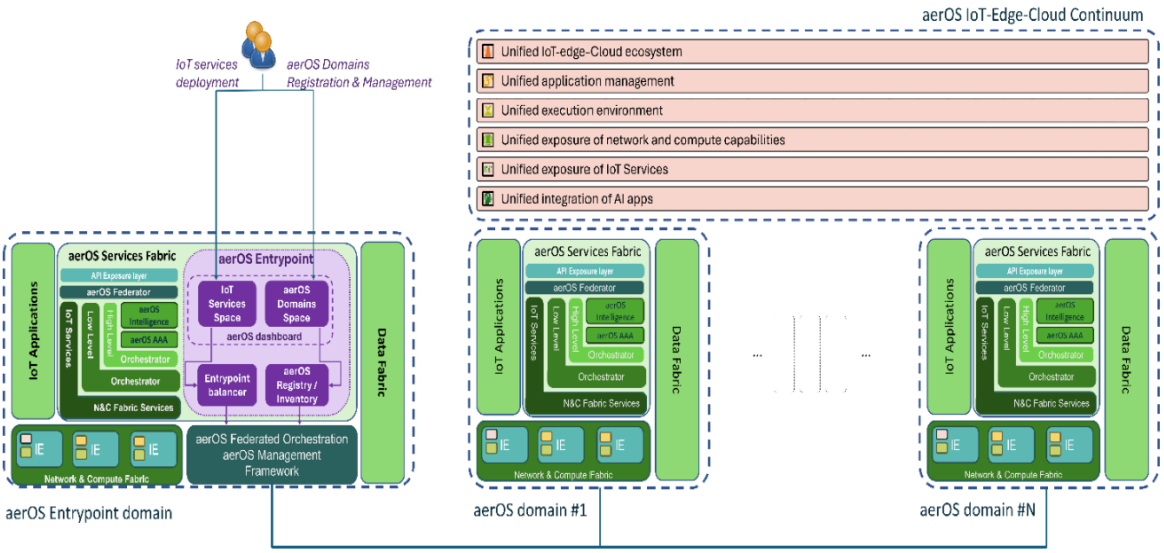


Figure 13: aeriOS Architecture

Standardized API Exposure and Common Frameworks Interoperability is a core pillar of the aeriOS framework, which is achieved through a unified and secure API strategy. Each domain features a KrakenD API Gateway that serves as the single point of entry for all Northbound interactions. The framework integrates natively with the 3GPP/ETSI penCAPIF, providing a standardized environment for service discovery and secure access for third-party

vertical applications [74]. aeriOS offers a comprehensive set of OpenAPI and AsyncAPI specifications to facilitate both synchronous and asynchronous interactions, including:

- HLO & LLO APIs: For service lifecycle management and orchestration requests.
- Data Fabric & Data Product Manager APIs: For accessing contextual information and managing data as a product.
- Self-Capabilities & IdM APIs: For managing autonomous rules and Identity/Access Management (based on Keycloak).
- IOTA & AsyncAPI Bindings: Supporting specialized communication for reliable services using DDS (Data Distribution Service) and ROS 2 bindings, which are critical for industrial and robotic 6G use cases.

In the context of Data Fabric, the IoT-Edge-Cloud continuum brings a highly distributed and dynamic data landscape. With the goal of providing a holistic view of all the data available in the IoT-Edge-Cloud continuum, whilst enabling data governance mechanisms that help ensure a responsible use of data, aeriOS aligns with the two recent data management approachesthat focus on decentralizing the management of data, namely, data mesh and data fabric. To cope with the complex data landscape of the continuum, aeriOS shifts the management of data close to their sources, i.e., to the data providing domains. In this sense, aeriOS embraces the data as a product thinking and the domain-oriented data ownership principle, as proposed by the data mesh paradigm [74]. Owners of data providing domains are responsible for turning their raw datasets into high-quality data products that data consuming domains can easily discover, understand, trust, and access. Nevertheless, building data products requires data engineering skills, as well as following standard data models and interfaces, which enable interoperability with data consuming domains. To help data owners in the creation of data products, aeriOS proposes a self-service data infrastructure that follows the architecture defined by the Data Fabric paradigm. The Data Fabric paradigm introduces a metadata-driven architecture that automates the integration of data from heterogenous sources and enables uniform access to the data through a standard interface. Aligning the Data Fabric architecture with the data mesh principles of data as a product and domain-oriented data ownership, results into the following novelties:

1. the Data Fabric, as the self-service data infrastructure, transforms the raw dataset of the data providing domain into a data product; and
2. the resulting data product can be accessed and shared with data consuming domains through the standard interface of the Data Fabric. Knowledge Graphs, represent a promising technology for the realisation of the Data Fabric architecture.

Knowledge graphs enable contextualized understanding of data by explicitly representing the facts (i.e., the data) in a graph structure, connected with the knowledge we have about them (i.e., the concepts). These explicit representations of knowledge, as concepts that relate to other concepts, in a formal way that can be understood by machines are known as ontologies, and these play a crucial role in the creation of the semantic layer [72]. Building on the principles of the knowledge graph, aeriOS proposes a definition of the data product as the transformation of a raw dataset into a semantically annotated data integrated in the knowledge graph, as depicted in Figure 14. This mapping between the conceptual level (represented by ontologies) and the physical level (represented by raw datasets such as data streams) is known as semantic lifting. In this semantic lifting process, the concepts extracted from the physical datasets are captured in the knowledge graph and linked with concepts from other physical datasets. As a result, each data product represents a subgraph of the whole knowledge graph created by the aeriOS Data Fabric.

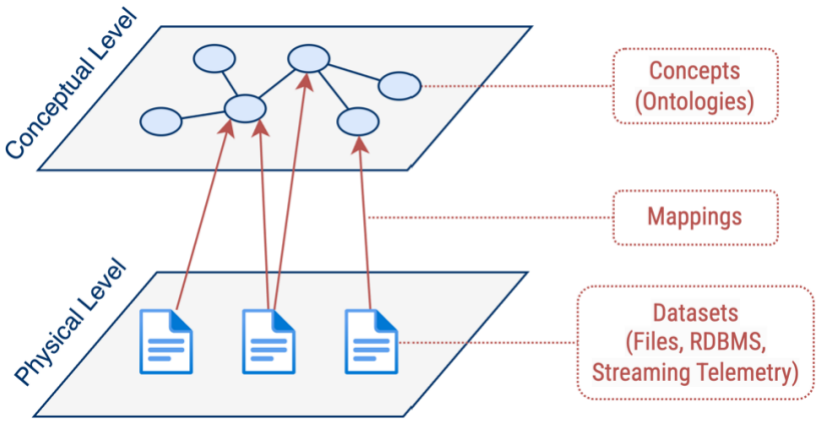


Figure 14: Semantic lifting based on mappings between the conceptual and physical levels.

To implement the knowledge graph, aerOS has adopted the ETSI NGSI-LD standard. NGSI-LD defines an information model derived from the property graph model, which additionally can reference ontologies. Thus, the NGSI-LD standard enables building property graphs where data are semantically annotated. In addition, the Representational state transfer (REST) API defined by NGSI-LD facilitates the management and interactions with the graph. The compact and natural information model of NGSI-LD, along with a friendly REST API, makes NGSI-LD a promising graph standard for implementing knowledge graphs, compared to other traditional graph standards such as the Resource Description Framework (RDF), which is deemed more verbose and entails a steep learning curve [75][76]. In terms of the NGSI-LD standard, the NGSI-LD Context Broker is pinpointed as the component that stores the knowledge graph. The NGSI-LD Context Broker, by means of the NGSI-LD API, allows data consumers to interact, query, and even subscribe to notifications in the stored knowledge graph. The aeriOS Data Fabric, as the self-service data infrastructure as per the data mesh paradigm, provides a

generic framework called Data Product Pipeline for data owners to build and onboard their own data products into the knowledge graph. The aeriOS Data Fabric, by means of the Data Product Manager, exposes an interface towards data owners to onboard new data products and orchestrate the pipeline that turns raw datasets into data products. These components are reflected in the high-level architecture of the aeriOS Data Fabric shown in Figure 15.

But, as illustrated in the high-level architecture, in addition to creating and sharing data products, the aeriOS Data Fabric also includes data governance mechanisms. aeriOS must govern a distributed mesh of data products scattered throughout the continuum; thus, mechanisms for cataloguing and controlling the consumption of data products must be put in place. In this regard, the aeriOS Data Fabric also relies on the knowledge graph to integrate the metadata that supports the data governance services. For example, when it comes to cataloguing data, the knowledge graph would contain metadata that describe the owner of a given data product, the domain that the data product belongs to, and even the concepts related to the data product. Similarly, for securing access to data, sensitive classification of data or access control policies based on the domain that provides a data product, could also be captured as metadata in the knowledge graph [75][76].

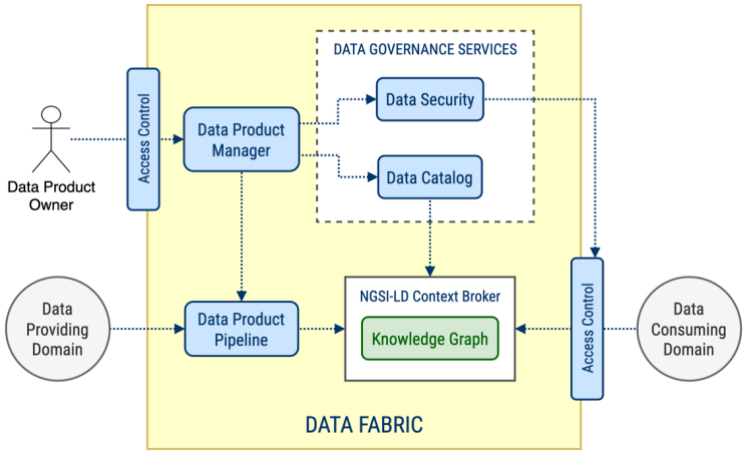


Figure 15: High-level architecture of the aerOS Data Fabric.

In the light of aeriOS Management framework, the Meta-OS developed in aeriOS must be managed by end users through the use of a common and recognisable interface, as in a traditional operating system, in which users can use a terminal or a command shell for this purpose. For instance, these tools allow users to install, uninstall and run programs or manage the set of users with the proper rights and permissions to interact with the operating system. In addition, OS have evolved to provide these users with a simpler and cleaner way of interacting with the system: a visual interface (e.g., a desktop) on top of the internal processes, so that they can manage the operating system using user-friendly “frontends”, while the actual processes are still being executed in the background, which means that they are hidden to the

users. In aeriOS, the intention is to follow the same approach that is fully adopted in traditional operating systems, so it have been decided to create one component to act as a single window for end users to manage the Meta-OS: the aeriOS Management Portal. This means that the portal is deployed in a unique domain, named as “Entrypoint domain”. However, the portal provides migration capabilities to be moved to another domain or IE due to a user requirement or an unexpected failure. Thus, this allows to avoid the loss of the unique management entrypoint, as is the case with high availability systems. As a global reflection, this approach is completely aligned with the principles of aeriOS of flexibility and decentralisation (“single entrypoint” does not mean a centralized computing approach). This user-friendly dashboard, developed as a modern Single-page web application (SPA), acts as a frontend for performing operations regarding the Meta-OS, which are finally performed by the aeriOS Basic Services in the background [72][74][76].

For example, an administrator can confirm the addition of a new domain or a new IE to the continuum using the portal, but the inclusion of this domain is managed by the aeriiOS Federator component. The interaction between the User Interface (UI) of the Management Portal and those basic services is undertaken by the backend of the aerOS Management Portal. Furthermore, this dashboard displays useful information gathered by these services to inform users of the status of the continuum (topology graph of the added domains, deployed services, state of domains and their IEs, etc) in real time. Finally, it is important to highlight that this portal does not add new capabilities to the Meta-OS, but leverages the capabilities offered by aeriOS to respond to user requests from a functional point of view. It, however, serves as the single window access for interacting with the continuum. This set of capabilities or actions which can be performed by users have been properly defined based on previously specified requirements by technical developers and potential end users of aerOS (e.g., use cases leaders). In that regard, a benchmarking tool has been identified as necessary to help this kind of end users to D2.7 – aeriiOS architecture definition improve their knowledge about the current performance of the aerOS Meta-OS. This tool is fed with data from the continuum to provide two main functionalities that complements the Management portal: 1) Benchmarking and comparison of IE/domains performance against known standards and methodologies, such as TPCx-IoT and RFC 2544. 2) Internal Technical KPIs dashboard, which provides a snapshot of these KPIs defined for aeriOS technical components in deliverable D5.2 and which are directly measurable from the aeriOS stack.

When it comes to Meta-OS end users’ management, aeriOS envisions the definition of a set of roles and permissions to be applied to them. Using a more practical example, the content that is displayed in the dashboard changes based on the role of the logged user, e.g., an administrator can only deploy certain services in a set of domains on which he has rights and

permissions, as an example. Still, the Entrypoint balancer is an important part of the aeriOS management framework (stuck to the Management portal) to avoid the addition of an additional centralization point, so that it emphasises the decentralised nature of aeriOS. After conducting a thorough review of the cutting-edge network load balancers, it has been determined that the most suitable LB algorithm in the case of the entrypoint balancer is the Least Connection type (or one of its modifications). This algorithm includes a straightforwardly updatable weighting function that is fed with metrics from the data fabric, such as the number of orchestration requests managed by each domain's HLO or IE capabilities. Moreover, more detailed technical information can be consulted in "D4.2 Software for delivering intelligence at the edge intermediate release" and future D4.3 [73][74][76]. However, the aeriOS Management Framework reaches beyond the capabilities offered only by the Management Portal. It also includes other management tools that oversee the creation and maintenance of the federation mechanisms between the multiple aeriOS domains that build the continuum. Here comes into the scene the aeriOS Federator, specifically the block related to the registration and discovery of these domains (Domain Registry and Discovery). The actions within the scope of this block are not performed directly by end users such as the ones of the Portal, but are smart, automatically conducted to react to some user actions, such as the creation of new domains or the modification of the IEs that belong to an existing domain. aeriOS is not a centralized solution, so by taking advantage of the aeriOS distributed state repository, this block is on charge of establishing the appropriate mechanisms to achieve a fully federated and decentralized architecture among the aeriOS domains. The set of distributed NGSI-LD Context Brokers, conforming the Distributed State Network of Brokers (DSNB), plays a major role in this architectural block. Figure 16 depicts the aeriOS management framework and its portal and federator.

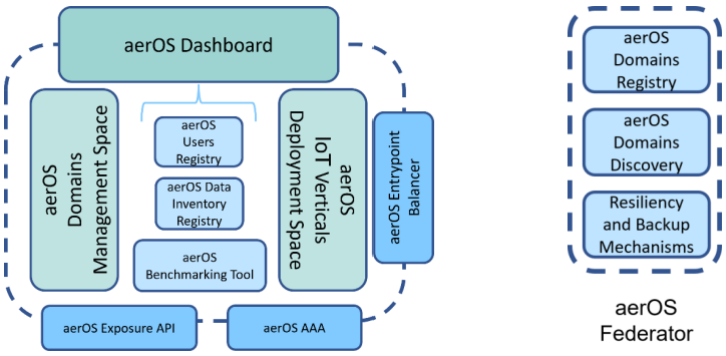


Figure 16: aeriOS Management Framework (left: aeriOS Management Portal, right: aeriOS Federator)

3.3.1.6.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The integration of the aeriOS framework within the UNITY-6G ecosystem is strategically targeted toward scenarios requiring advanced autonomy, edge-cloud continuity, and resource-efficient orchestration. Rather than serving as a monolithic solution for all project activities, aeriOS acts as a specialized distributed intelligence enabler for the following identified PoCs and Experimental Scenarios (ES), where its Meta-OS capabilities align most closely with the project's technical KPIs.

Support for PoC #1: AI-native Integrated Architecture for Sustainability and Resilience. The aeriOS framework is a potential contributor to PoC #1, which evaluates the synergy between AI-native management and infrastructure resilience. Within this PoC, aeriOS can support ES1 (Sustainable Networks for Disaster Handling) by providing the "Island Mode" orchestration logic necessary for autonomous edge operation when core connectivity is lost. The framework's ability to perform local re-orchestration ensures that critical situational awareness services remain active during disasters, directly addressing the KPI for service recovery time. Furthermore, aeriOS is able to support ES2 (NPN for Time-Sensitive and Sustainable Industrial Applications) by leveraging its LLO to provide deterministic resource provisioning for non-public networks. In this context, the framework's AI-driven scaling logic facilitates "Green AI" objectives by optimizing the energy-performance trade-off for industrial IoT workloads, ensuring that power consumption is minimized without violating latency constraints.

Support for PoC #2: Unified Management for Terrestrial and Non-Terrestrial Networks In PoC #2, which focuses on the seamless integration of TN and NTN components, aeriOS provides the underlying management plane for ES3 (TN-NTN Integration for Seamless Connectivity) and ES5 (Unified Resource Management in Multi-Domain Continuums). A major challenge in these scenarios is the dynamic nature of satellite-based links and the resulting variability in backhaul performance. aeriOS potentially could addresses this through its Context Broker (Orion-LD), which allows the orchestration layer to remain "state-aware." By monitoring the real-time quality of the NTN link, the HLO can proactively migrate stateful applications from the edge to the cloud (or vice-versa) to prevent service interruptions during satellite handovers or signal degradation. This ensures a transparent experience for the end-user, regardless of the underlying transport technology.

While aeriOS is primarily an orchestration and execution framework, it could provide critical data-plane support for ES4 (Digital Twin for Integrated 6G Network Evaluation) and ES6 (Advanced Services for Immersive Experiences). For the Digital Twin scenario, aeriOS is able to act as a telemetry aggregator, utilizing its NGSI-LD-based Data Fabric to feed real-time infrastructure state information into the virtual twin, enabling more accurate conflict resolution

and predictive modeling in WP4. For immersive XR services in ES6, aeriOS could provide the high-performance execution environment required for latency-critical rendering tasks. By utilizing eBPF-based networking, the framework minimizes the "jitter" and communication overhead between distributed microservices, providing the stable performance foundation required for the "holographic-grade" communication targets of UNITY-6G.

Beyond the state-of-the-art, the implementation of the aeriOS framework within UNITY-6G is able to shift from centralized, reactive cloud-native orchestration toward a truly AI-native, decentralized Meta-Operating System. While current industry standards (e.g., vanilla Kubernetes) are designed for stable data center environments, the NCSRD contribution introduces a hierarchical, intent-based orchestration logic that abstracts the entire Cloud-Edge-IoT continuum into a single, programmable execution fabric. This approach moves beyond the SotA in three critical dimensions: Resilient Autonomy, through "Island Mode" operations that sustain services during backhaul failure; Contextual Intelligence, by utilizing a distributed NGSI-LD Data Fabric to manage stateful migrations across dynamic NTN links; and Sustainable Orchestration, by employing the ORIENT algorithm (D3QL/GNN-based) to autonomously balance sub-millisecond latency requirements with aggressive energy-saving targets. By decoupling high-level intent from low-level execution, aeriOS provides the zero-touch management necessary for the integrated, multi-domain 6G architectures of the future.

The mapping of aeriOS assets to the UNITY-6G PoCs identifies the framework as a specialized enabler for distributed intelligence and resilience. In PoC-1, aeriOS provides the foundational autonomy for ES1, ES2, and ES3 by utilizing the LLO for localized "Island Mode" recovery and the HLO's ORIENT AI for energy-efficient, intent-driven resource allocation. In PoC-2, the framework ensures architectural convergence for ES4, ES5, and ES6; it employs eBPF-based networking for deterministic industrial control, the NGSI-LD Data Fabric for real-time Digital Twin telemetry, and cognitive placement to maintain the low-jitter environments required for immersive XR. This targeted deployment allows aeriOS to manage the 6G continuum as a unified, self-optimizing fabric across both terrestrial and non-terrestrial segments. Table 1 depicts the high-level mapping with UNITY-6G PoCs.

Table 1: aeriOS framework high-level mapping with UNITY-6G PoCs

PoC - Experimental Scenario	**aeriOS Key Component - Service**	**Technical Contribution to UNITY-6G**
PoC-1 ES1: Sustainable Networks for Disaster Handling	LLO & Self-Healing Engine	Enables "Island Mode" and autonomous local recovery for critical services.
PoC-1 ES2: 6G Integrated Network Orchestrator	HLO (ORIENT AI) & Data Fabric	Provides semantic-aware orchestration and AI-driven

		resource allocation.
PoC-1 ES3: Integrated Network Infrastructure for Sustainable Shared Resources	HLO Placement Engine	Optimizes energy-aware workload placement across shared infrastructure.
PoC-2 ES4: Multi-RAT O-RAN for Time Sensitive Services	LLO & Smart Networking (eBPF)	Facilitates deterministic management for 6G and IEEE 802.11 segments.
PoC-2 ES5: Digital Twin for Network Evaluation	Data Fabric (NGSI-LD Broker)	Aggregates real-time telemetry into the standardized models required for the twin.
PoC-2 ES6: Advanced Immersive XR Experiences	Smart Networking & HLO	Minimizes communication jitter and optimizes rendering placement for XR.

3.3.1.6.2 Preliminary results

To allow nodes that connect to the aeriOS compute continuum to be autonomous, they need to have certain capabilities. These features are offered through the aeriOS self-capabilities suite to all IEs that connect to the continuum, which are:

- *Self-awareness:* considered one of the main self-* capabilities of an autonomous system, this compo-nent analyses and obtains information from the node, continuously monitoring its health status and workload. Due to the need to offer real-time information on the status of the IE, this module is subdivided into two components, which are executed continuously. One (power_consumption) is in charge of obtaining the energy consumption of the node, which requires more computing time. The other (hardware_info) is responsible for obtaining the rest of the parameters. The component that obtains the power consumption needs an average of 20-25 seconds per execution to obtain new valid values and the other only needs about 3-15 seconds to update its information, depending on the amount of in-formation to be collected by the sub-module. This amount is specified by environment variables in the sub-module deployment files. The purpose is to provide updated information to the rest of the self-* capabilities as fast as possible to modify the operation of the IE, if necessary. Currently, this self-* capability is able to obtain the following information from each node: hostname, addresses (internal IP and MAC), CPU (architecture, number of cores, max frequency and current usage), RAM (total capacity, available capacity and current usage), disk (type, total capacity, available capacity and cur-rent usage), network (speed up, speed down, traffic up, traffic down and lost packages), power consumption (current and average), capability to execute workloads in real-time and operating system. To obtain all these parameters, both sub-modules use external packages and libraries such as PowerTOP, iproute2, psutil, getmac or speedtest-cli, as detailed below. On the other hand, each sub-

module has a REST API that allows the sampling frequency of the information in each node to be set independent-ly. This makes it possible to optimise the operation of the node within the domain to which it belongs and reduce the consumption of resources. Below are screenshots of the two sub-modules running on the continuous development and integration cloud infrastructure of the project (provided by partner CF).

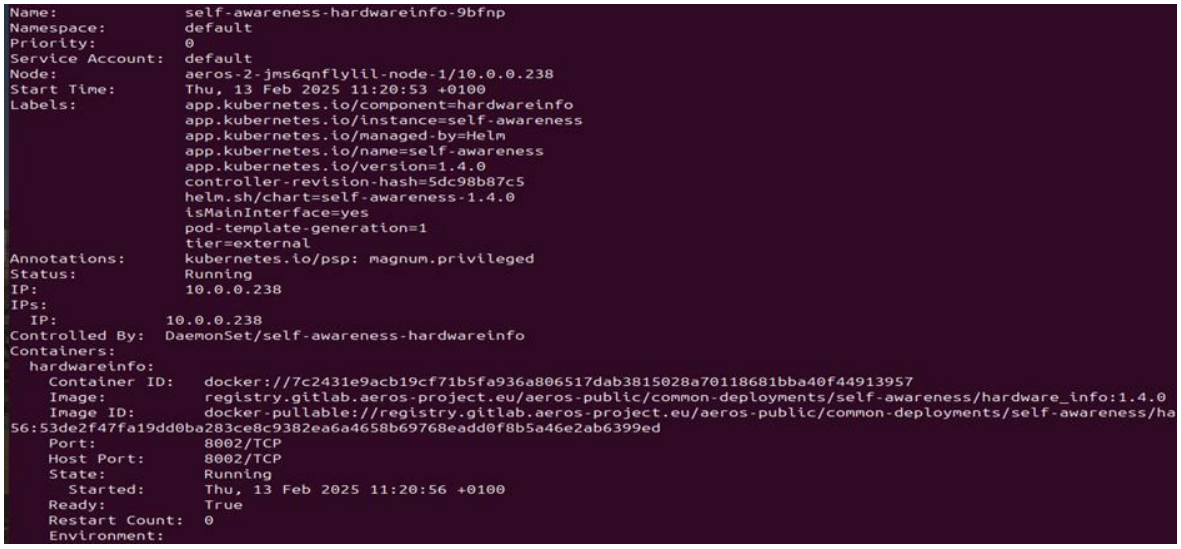


Figure 17: Hardware info sub-module running on a test cluster of infrastructure

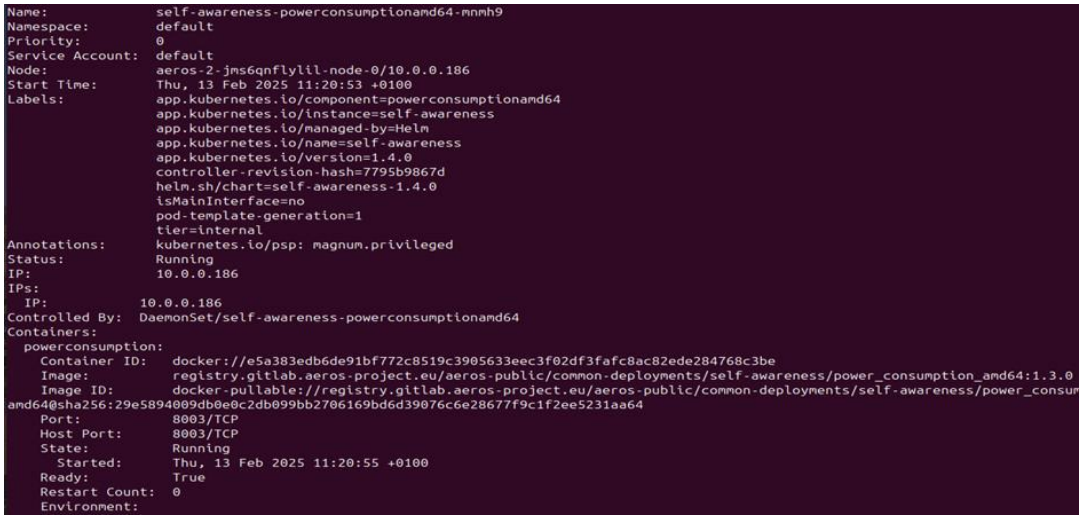


Figure 18: Power consumption sub-module running on a test cluster of infrastructure

- *Self-orchestrator:* considered one of the main self-* capabilities of an autonomous system, this component is composed by a Rules Engine, a Facts Generator, a Trigger and wrapped by a REST API. It is capable of managing facts, rules and alerts, obtaining information from the self-awareness, self-realtimeness, self-healing and self-optimisation and adaptation modules to send warnings about prob-lems in the IE to the aerOS EAT (Embedded Analytics Tool) of the domain, with the goal to improve the management and coordination of their own workloads. This improves the scalability of tasks and reduces the number of

errors that occur during task execution. This self-* capability uses libraries such as json-rules-engine or jsonschema, as detailed below. In order to be able to manage the rules for detecting faults, malfunctions or anomalous situations within a node, the module exposes a REST API that allows CRUD (Create, Read, Update and Delete) operations to be performed on these rules. Moreover, by means of persistent storage in the node, this module maintains an updated backup copy of the rules to restore them to their last state in case of failure in the IE. On the other hand, this REST API also allows to receive alerts in a predefined format coming from other self-* modules in order to carry out the necessary corrective measures through the aeriOS EAT. In addition, when a rule is triggered or an alert is received at the corresponding REST API endpoint, a message is sent to the domain's IOTA hornet node to register the event. Below is a screenshot of the module running on the continuous development and integration cloud infrastructure of the project (provided by partner CF).

```
Name:             self-orchestrator-orchestrator-b6dcv
Namespace:        default
Priority:         0
Service Account:  default
Node:             aeros-2-jms6qnflylil-node-0/10.0.0.186
Start Time:       Thu, 06 Feb 2025 10:51:54 +0100
Labels:           app.kubernetes.io/component=orchestrator
                  app.kubernetes.io/instance=self-orchestrator
                  app.kubernetes.io/managed-by=Helm
                  app.kubernetes.io/name=self-orchestrator
                  app.kubernetes.io/version=1.2.0
                  controller-revision-hash=657c67c45c
                  helm.sh/chart=self-orchestrator-1.2.0
                  isMainInterface=yes
                  pod-template-generation=4
                  tier=self
Annotations:      kubernetes.io/psp: magnum.privileged
Status:           Running
IP:               10.0.0.186
IPs:
  IP:           10.0.0.186
Controlled By:  DaemonSet/self-orchestrator-orchestrator
Containers:
  orchestrator:
    Container ID:   docker://27010fc4d52d8fd479f753a4bc79a9c05437a9e153c46a1a869fbd8906bd0767
    Image:          registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-orchestrator:1.2.0
    Image ID:       docker-pullable://registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-o
a9be96591684151ebdde18e40ef720a6a662181259c4d0b2a2788496
    Port:           8001/TCP
    Host Port:      8001/TCP
    State:          Running
      Started:      Thu, 06 Feb 2025 10:51:55 +0100
    Ready:          True
    Restart Count:  0
    Environment:
```

Figure 19: Self-orchestrator module running on a test cluster of infrastructure

- *Self-security:* developed using Suricata (open-source network analysis and threat detection software), it monitors traffic logs in real-time from the network card to detect threats and abnormal behaviours through Log Monitoring module. The ETL (Extraction, Transformation, Load) processing module then collects the security logs, converts them into structured JSON format and sends alerts to an end-point (Trust Manager). These alerts allow to discover different types of network attacks to detect vulnerabilities and threats at the IE level. An API has also been created so that the Trust Manager can make requests each

week, with the intention of collecting the alerts for the whole week and saving them in the history. The alerts generated by self-security are deleted once a week by a new service with the intention of minimising the space that this component occupies on disk.

Below is an example of the alerts generated by the self-security running on the continuous development and integration cloud infrastructure of the project (provided by partner CF).

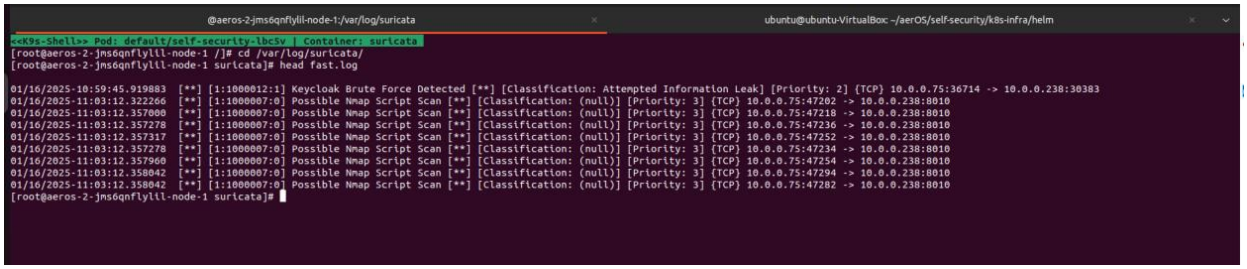


Figure 20: Self-security alert example

- *Self-API:* this self-capability consists of a global API deployed in each IE of the aeriOS continuum that exposes the functions that can be executed on the rest of the self-* capabilities installed on the node, controlling the input and output data flows.

Below is a screenshot of the self-API module running on the project's continuous development and integration cloud infrastructure (provided by partner CF):

```
Name:             self-api-api-6nc9s
Namespace:        default
Priority:         0
Service Account:  default
Node:             aeros-2-jms6qnflylil-node-8/10.0.0.158
Start Time:       Wed, 26 Feb 2025 15:46:05 +0100
Labels:           app.kubernetes.io/component=api
                  app.kubernetes.io/instance=self-api
                  app.kubernetes.io/managed-by=Helm
                  app.kubernetes.io/name=self-api
                  app.kubernetes.io/version=1.0.0
                  controller-revision-hash=5cdd9cd5fd
                  helm.sh/chart=self-api-1.0.0
                  isMainInterface=yes
                  pod-template-generation=1
                  tier=self
Annotations:      kubernetes.io/psp: magnum.privileged
Status:           Running
IP:               10.0.0.158
IPs:
  IP:           10.0.0.158
Controlled By:  DaemonSet/self-api-api
Containers:
  api:
    Container ID:   docker://44a9de63e9d3aeee0de295dfad45d63c0ac55e99a1e854a6d4399da84274f265
    Image:          registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-api:latest
    Image ID:       docker-pullable://registry.gitlab.aeros-project.eu/aeros-public/common-deployments
    Port:           8600/TCP
    Host Port:      8600/TCP
    State:          Running
      Started:      Wed, 26 Feb 2025 15:46:19 +0100
    Ready:          True
    Restart Count:  0
    Environment:
      PORT:                    8600
      URL:                     0.0.0.0
      SELF_ORCHESTRATOR_IP:     (v1:status.podIP)
      SELF_ORCHESTRATOR_PORT:  8001
      SELF_DIAGNOSE_IP:         (v1:status.podIP)
      SELF_DIAGNOSE_PORT:      8002
      SELF_SECURITY_IP:         (v1:status.podIP)
      SELF_SECURITY_PORT:      8000
      SELF_OPTIMIZATION_IP:     (v1:status.podIP)
      SELF_OPTIMIZATION_PORT:  8090
      SELF_HEALING_IP:          (v1:status.podIP)
      SELF_HEALING_PORT:       8500
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nknlv (ro)
```

Figure 21: Self-API module running on node-8 of test K8s cluster-2 of infrastructure

- *Self-healing*: capability of autonomously recovering affected parts of the system both at the hardware and software level caused by failures or abnormal states. It also can restart the system to pre-established routines scheduling, if necessary. This module detects and remedies abnormal states of the network, outlier values of sensors connected to the IE, and issues with the IE's power level. Since the self-healing module detects abnormal states or outlier values, it generates a JSON alert message and sends this message to the Trust Manager component for the health score calculation algorithm. The following is an example of a POST JSON message:

```
{
  "timestamp": "2025-02-04T11:00:00",
  "scenario": "Sensor Failure",
  "message": "Sensor measurement detected as an outlier. Exclude the sensor from the set of those that provide input to the system",
  "mac_address": "fa:16:3e:5e:25:ef"
}
```

Figure 22: Example of JSON alert from self-healing to Trust Manager

Furthermore, the module collects these alert messages generated from scenarios and exposes a GET API endpoint, which is consumed by self-API component. An example of these JSON alerts is shown below:

```
{
  "alerts": [
    {
      "timestamp": "2025-02-04T11:00:00",
      "scenario": "sensor_failure",
      "message": "Sensor failure detected",
      "mac_address": "fa:16:3e:5e:25:ef"
    },
    {
      "timestamp": "2025-02-04T11:00:00",
      "scenario": "network_violation",
      "message": "Network duty cycle exceeded",
      "mac_address": "fa:16:3e:5e:25:ef"
    }
  ]
}
```


Figure 23: Example of JSON alerts from self-healing to self-API

- *Self-scaling:* possibility of horizontally increasing or decreasing the hardware resources dedicated to workloads of a node running Kubernetes. These changes depend on the needs of each workload, are executed in real-time, and are based on time series inference and custom logic.
- *Self-configuration:* ability to maintain the desired state of the system with the help of an abstract and reactive management of its configuration. Both the configuration itself and its possible evolution can be defined/represented based on the concepts such as “resource”, “requirement”, and action/reaction. Development has focused on evolving an existing open-source tool (originated in H2020 project AS-SIST-IoT), by integrating the innovative needs of aerOS and incorporating automated configuration.
- *Self-optimisation and adaptation:* ability to optimise the dissemination of the data and control the performance of IE. On the one hand side, by using dynamic sampling techniques and the current metric streams (i.e. the IE’s operational data obtained from self-awareness), the component suggests optimal sampling periods, allowing control over the frequency of data monitoring by the self-awareness component. On the other hand, it incorporates an estimation model monitoring the shifts in the metric streams to detect data points with significant differences that may indicate potential anomalies, aiming to prevent the overutilization and underutilization of IE resources.
- *Self-realtimeness:* an experimental capability that continuously monitors the performance of real-time services using their time utility (TU) that degrades with the tardiness of deadline misses. The component automatically adjust the CPU time (quota) granted to a real-time

service every period to trade-off CPU utilisation and TU achieved on an IE. If a real-time service's TU degrades below a configurable threshold self-realtimeness issues a re-orchestration request as illustrated in the following figure:

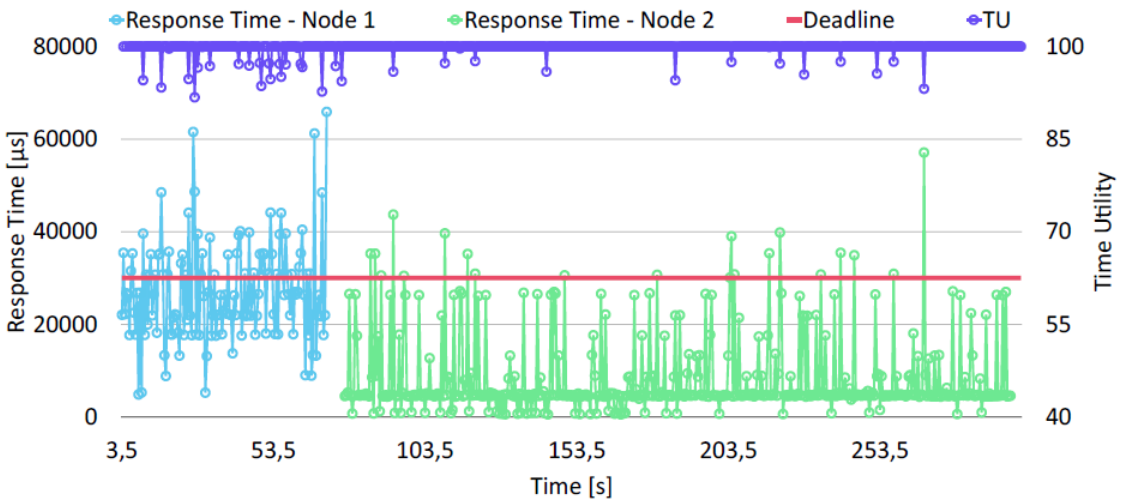


Figure 24: Self-real timeness relocating real-time workload with bad time-utility from node 1 to node 2 (2)

3.3.2 Resource allocation, traffic load balancing, and routing

3.3.2.1 TN-NTN Traffic Offloading

The proposed algorithm addresses uplink traffic offloading in a 6G network where the same area is served by both a TN and a NTN. The NTN, represented by a LEO node, is assumed to operate on a different frequency band from the TN. Both the TN and NTN gNBs are connected to a common core network (CN), which in turn is connected to a server hosting a RIC. The AI algorithm is deployed in the RIC, where it can gather information from the CN, which itself collects information from both the TN and NTN gNBs.

Users on the ground are assumed to be uniformly distributed across the coverage area. Each user measures the RSRP of the available gNBs and connects to the one with the strongest signal. Every UE also reports its buffer status report (BSR), indicating how much data it has to transmit. Based on the resources available, each gNB then decides which users to serve. The network is assumed to operate in resource-constrained conditions, *i.e.*, the number of requested Physical Resource Blocks exceeds the number that can be served simultaneously. The overall framework is reported in Figure 25 with served users being marked in red and unserved users in blue.

The objective of the algorithm is to jointly optimise the overall system throughput and the long-term system fairness across UEs: this is achieved by intelligently offloading uplink traffic between the TN and the NTN segments. Due to the high-dimensional nature of the decision problem, which make its solution through conventional optimisation methods unfeasible from

the decision latency perspective, an Artificial Intelligence approach based on Deep Reinforcement Learning is adopted. Specifically, a Deep Q-Learning method using a Double Deep Q-Network (DDQN) is designed to learn an offloading policy that improves both throughput and fairness.

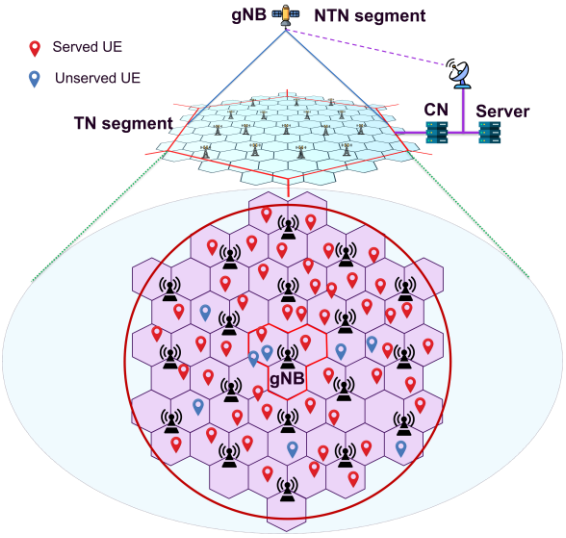


Figure 25: Overall framework for TN/NTN traffic offloading

3.3.2.1.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

Within the UNITY-6G architecture, the proposed algorithm runs as an xApp through the DE, physically placed on ground and connected to both the NTN and TN gNBs through the CN. The cross-domain coordination between the O-RAN domain and the NTN domain achieved through the UNITY-6G architecture ensures that 1) analytics from both domains can be used at the xApp, and 2) decisions can affect both domains concerned.

The algorithm targets KPI #20: increase link utilization from 40% to 80%.

3.3.2.1.2 Preliminary results

This section therefore presents the preliminary outcomes of the design phase, i.e., the formulation of the optimisation problem and the DDQN-based method developed to solve it.

Optimization problem formulation

The optimisation focuses on the two KPIs introduced above, system throughput and system fairness, which together capture both the system-level performance and the long-term performance experienced by each UE. These two metrics are jointly optimised through traffic offloading, as expressed in the following problem:

$$P = \max_{\mathbf{x}} (\alpha_1 T_{system} + \alpha_2 J_{system})$$

$$\begin{cases} C_1 : \sum_{g=1}^{N_{gNB}^{(TN)}} x_{i,TN_g} + x_{i,NTN} \le 1 \\ C_2 : \sum_{i=1}^{N_{UE}} \sum_{g=1}^{N_{gNB}^{(TN)}} x_{i,TN_g} \gamma_i \le N_{PRB,TN}^{\max} \\ C_3 : \sum_{i=1}^{N_{UE}} x_{i,NTN} \gamma_i \le N_{PRB,NTN}^{\max} \end{cases}$$

Here, the optimisation variable is the UE-to-gNB allocation $\mathbf{X} = \left[\mathbf{X}^{TN} \mid \mathbf{X}^{NTN} \right]$, where $\mathbf{X}^{TN} = \{x_{i,g}^{TN}\}_{i=1}^{N_{UE}, N_{gNB}^{(TN)}}$ and $\mathbf{X}^{NTN} = \{x_i^{NTN}\}_{i=1}^{N_{UE}}$. The coefficients α_1 and α_2 weight the relative importance of system throughput and system fairness in the objective. Constraint C1 guarantees that a UE is served by no more than one gNB, while constraints C2 and C3 ensure that the total bandwidth allocated within any gNB, whether TN or NTN, does not exceed its maximum capacity, expressed in terms of PRBs. This problem is high-dimensional and time-consuming to solve with conventional optimisation methods. Given the proven effectiveness of Deep Reinforcement Learning, and in particular Deep Q-Learning, for resource allocation in TN-NTN settings, a Deep Q-Learning-assisted method is designed to solve it.

Optimization problem formulation

The optimisation process is modelled as a Markov Decision Process (MDP) defined by the tuple of state space, action space, transition probability and reward function. The state provides the key information about the offloading scenario at each time step, namely the allocation state of the UEs, the traffic load of the gNBs, the signal-to-interference-plus-noise ratio of each active UE-gNB channel, the long-term service state of the UEs, and the progress of the current episode in time. The action represents the allocation decision for a specific UE: it can either leave the situation unchanged or change the allocation state of a chosen UE, provided enough bandwidth remains in the target gNB. The transition probability describes the likelihood of moving from one state to the next under a given action, and the reward function is built directly from the joint objective, continually encouraging the agent to reach higher system throughput and fairness.

At each time step, the agent observes the environment input and the associated reward, and selects an action accordingly. The environment then updates its state in response to that action

and returns the new input and reward to the agent. This interaction is repeated until the maximum number of steps in the episode is reached, as illustrated in Figure 26.

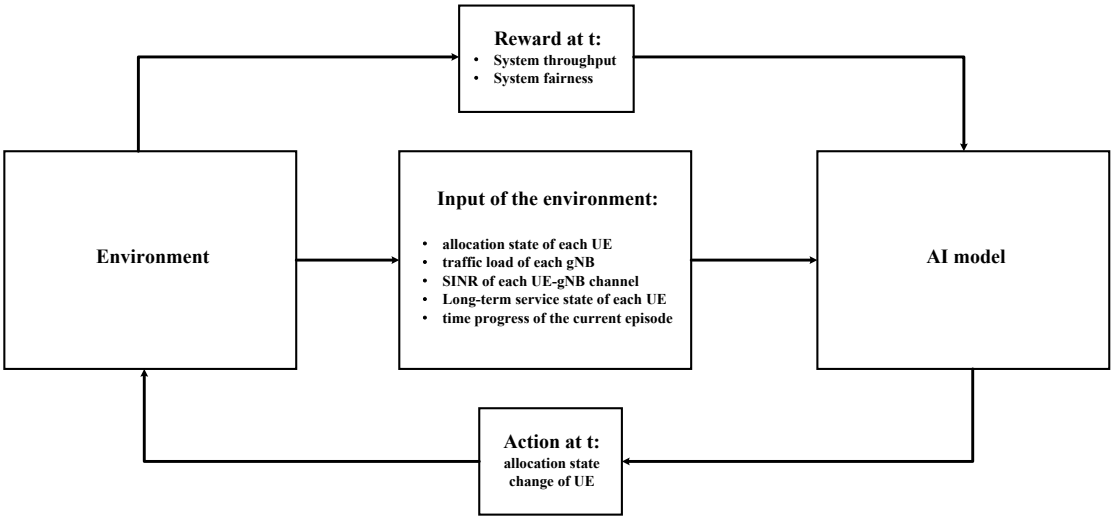


Figure 26: Proposed optimization framework

The algorithm relies on a DDQN, whose detailed structure is shown in Figure 27. A DDQN overcomes the tendency of a standard Deep Q-Network to overestimate action (Q) values by using two separate deep neural networks: an Online Network that determines which action to take, and a Target Network that evaluates how good the chosen action is. In this design, both networks share the same structure of three fully connected layers, each followed by a rectified linear unit (ReLU) activation. The input to the network is the state described in the MDP model, and the output is the allocation action for a specific UE. The reward computed from the joint objective then drives the learning process, steadily guiding the DDQN agent towards higher system throughput and fairness.

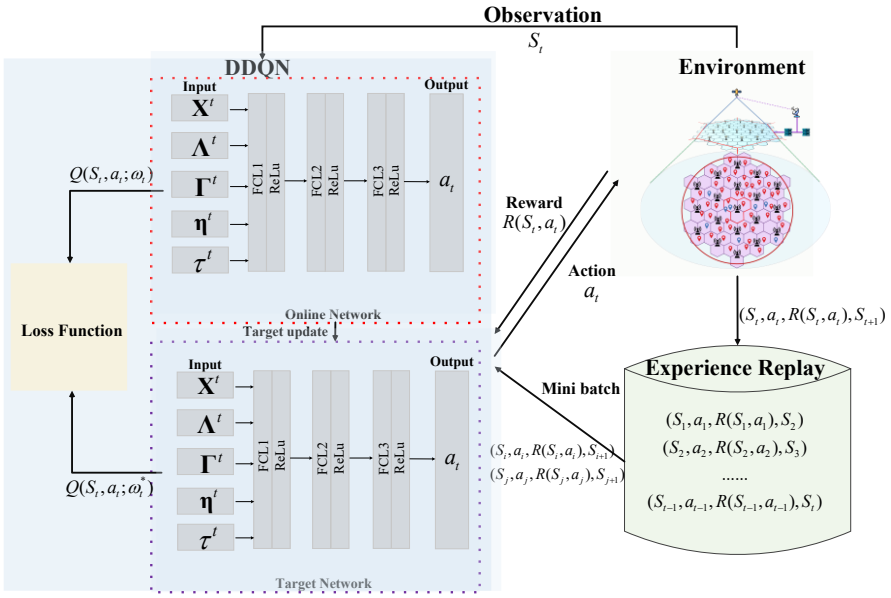


Figure 27: Double Deep Q-Network architecture

3.3.2.2 Energy-Efficient Decentralized Computation Offloading in the Continuum

The rapid growth of IoT, cyber-physical, industrial monitoring, real-time video analytics, autonomous systems, and extended reality services is creating a new class of workloads that are both latency-sensitive and energy-constrained. These workloads cannot be efficiently handled by a purely centralized cloud model, since cloud-only processing may introduce excessive communication latency, bandwidth consumption, and energy overhead. At the same time, processing everything locally at the edge may overload resource-limited edge nodes and increase the probability of deadline violations. The Cloud–Edge Continuum addresses this limitation by distributing computing resources across IoT, edge, and cloud layers, enabling workloads to be processed locally, offloaded horizontally to neighboring edge nodes, or offloaded vertically to cloud resources. In this context, task offloading becomes a core AI-driven orchestration function, because each incoming workload must be dynamically assigned to the most appropriate execution location according to current and predicted system conditions.

Task offloading in the Cloud–Edge Continuum is challenging because the decision depends on several coupled factors: workload size, processing density, deadline, current queue status, available edge/cloud capacity, communication rate, and energy consumption. The proposed DECOFFEE framework models this problem as a non-convex multi-objective optimization problem where the placement decision must jointly minimize workload execution delay, energy consumption, and workload drop rate due to deadline violations. The problem is further complicated by partial observability and decentralization: each edge node has only local information and shared telemetry, while the global system state evolves asynchronously due to stochastic workload arrivals and the decisions of other edge agents.

More analytically, DECOFFEE advances the state-of-the-art by introducing a decentralized reinforcement learning framework for time-critical and energy-efficient workload offloading across the Cloud–Edge Continuum. Its main novelty lies in the fact that each Edge Agent operates as an autonomous learning entity, while still exploiting shared telemetry and LSTM-based predictions of future load. Instead of relying on a centralized controller, DECOFFEE formulates the placement process as a collection of parallel agent-specific Markov Decision Processes. This allows the framework to scale across multiple edge nodes and to support local, horizontal, and vertical workload placement decisions under partial observability.

The algorithmic contribution is also multi-objective. Each agent learns a policy that balances three competing goals: reducing workload execution delay, reducing energy consumption, and avoiding workload drops caused by deadline violations. This is achieved through a weighted cost function where delay-awareness and energy-awareness coefficients can be tuned depending on the operational objective. Consequently, DECOFFEE can operate in different modes, including delay-aware, energy-aware, or balanced offloading. This flexibility is particularly important for continuum environments where operational priorities may vary depending on the use case, for example ultra-low-latency applications, battery-sensitive IoT deployments, or balanced service-level operation.

The considered system follows a three-tier computing continuum architecture composed of an IoT layer, an edge layer, and a cloud layer, depicted in Figure 28. In the IoT layer, devices generate computational workloads and are served by Radio Units. Each Radio Unit is associated with an Edge Agent, which provides edge computing capabilities for the corresponding cell. The cloud layer is represented by one or more Cloud Agents with higher computational capacity. The Edge Agents and Cloud Agent jointly form the set of available computing nodes across the continuum.

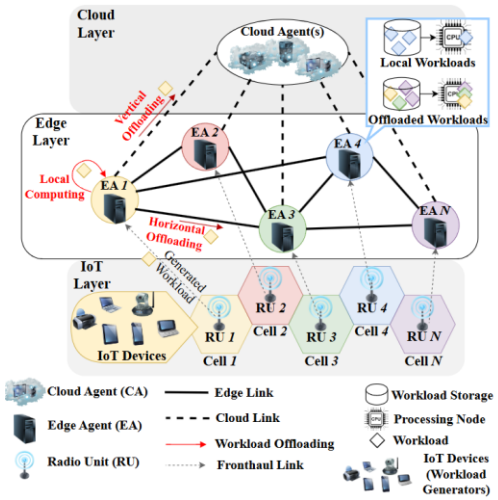


Figure 28: Three-tier computing continuum architecture considered the DECOFFEE framework.

Each incoming workload is atomic and non-partitionable, meaning that it must be executed entirely at one selected destination. Upon workload arrival, the corresponding Edge Agent must decide among three possible actions: local computation using its own private processing capacity, horizontal offloading to another Edge Agent, or vertical offloading to the Cloud Agent. Horizontal offloading enables cooperative use of neighboring edge resources, while vertical offloading exploits the larger processing capacity of the cloud at the cost of higher communication overhead. This three-action model is the core operational abstraction of the DECOFFEE framework.

The internal workload management model distinguishes between private, public, and offloading workload stacks. Each Edge Agent maintains a private workload stack for its locally generated and locally executed tasks, an offloading stack for workloads waiting to be transferred, and public workload stacks for tasks received from other Edge Agents. The Cloud Agent maintains public workload stacks for workloads offloaded from the edge. All workload stacks follow a FIFO discipline. This modeling choice allows the framework to explicitly capture queueing delays, execution delays, and deadline violations across all stages of workload placement.

Telemetry Agents are also included in the system model. They monitor load, CPU utilization, energy usage, and workload status at the edge layer. The Telemetry Agents maintain recent load histories of the continuum nodes and share this information with the local DECOFFEE agents. This telemetry mechanism enables each Edge Agent to make decisions that are not purely local, but enriched with partial continuum-wide awareness. In DECOFFEE, this information is further processed by an LSTM module to predict future node load and support proactive workload placement.

Algorithmic Framework: DECOFFEE formulates workload placement as a decentralized reinforcement learning problem, where each Edge Agent independently learns how to assign newly arrived workloads to the most suitable execution destination. At each time slot, an Edge Agent observes its local state, which includes the workload size, the waiting time in its private and offloading workload stacks, the status of public workload stacks associated with its offloaded tasks, and the predicted future load of the continuum nodes. Based on this state, the agent selects one of the available placement actions: local computation, horizontal offloading to another Edge Agent, or vertical offloading to the Cloud Agent.

The decision-making process is modeled as a set of parallel agent-specific Markov Decision Processes. Each agent receives a delayed cost signal after the workload is either successfully completed or dropped due to deadline violation. This cost combines execution delay and energy consumption, while deadline violations are penalized through a fixed drop penalty. In

this way, DECOFFEE jointly optimizes latency, energy efficiency, and workload completion reliability.

The internal AI model of each DECOFFEE agent combines an LSTM-based forecasting module with a Double Dueling Deep Q-Network, as shown in Figure 29. The LSTM module predicts short-term future load across the continuum, allowing the agent to anticipate congestion before selecting a placement action. The predicted load is then combined with local workload and queue-related information and passed to the DQN, which estimates the expected long-term cost of each possible action. The dueling architecture improves learning stability by separately estimating the value of the current state and the advantage of each action, while the Double DQN mechanism reduces overestimation during training.

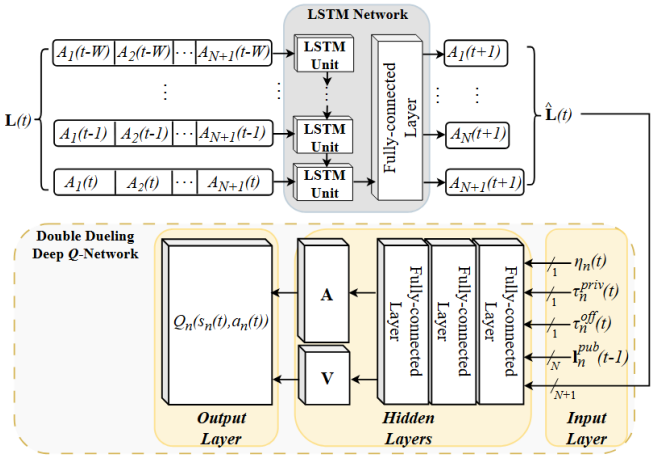


Figure 29: The internal architecture of DECOFFEE agent n, including stacked LSTM units and a Double Dueling DQN.

During inference, the trained DECOFFEE agent performs only a lightweight feed-forward pass through the neural network and selects the action with the lowest expected cost. This makes the framework suitable for real-time operation in Cloud–Edge Continuum environments, where workload placement decisions must be taken rapidly, under partial observability, and without relying on a centralized orchestrator.

3.3.2.2.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The DECOFFEE agents are just plug-in software agents running in the DU to assist compute workload placement. In this sense, the presented architecture is integrated within the ORAN domain of UNITY-6G and the DECOFFEE software can be part of ORAN DMO control functionalities (orchestrating how the workloads are allocated). In addition, the presented DECOFFEE framework is associated with KPI #40, which concerns the increase of edge computational resource utilization by 40% (relative to baselines). This KPI will be measured in a simulated environment consisting of several edge and one cloud servers. Computational tasks will be generated by the IoT/end device layer (the hierarchically lower layer) and will be

sent to the edge continuum for execution. The tasks (e.g., image processing, training of an ML model) will have parameters like deadline, size and required computational capacity. DECOFFEE models that are hosted in the edge servers will decide upon executing the task locally or offloading it either to other edge servers (horizontally) or to the cloud (vertically), conforming with the task latency constraints. Then, the ratio of the offloaded tasks to the edge vs cloud will be quantified in different scenarios and topologies (for different number of edge servers, varying the network topology and the links between them, etc.). The ML-assisted offloading scheme will be compared against rule-based approaches, e.g., “execute locally” or “offload to comply with the deadline” policies.

3.3.2.2.2 Preliminary results

For the preliminary evaluation of DECOFFEE, the system was configured with 20 interconnected edge nodes. Particular attention was given to the hyperparameters affecting the convergence and stability of the distributed DRL agents. DECOFFEE considers an application-agnostic workload model, where each workload is described by its size and processing density, without relying on application-specific assumptions. Each training episode includes 110 time slots, where the first 100 slots are used for workload placement decisions, while the final 10 slots allow pending workloads to be processed and delayed rewards to be collected.

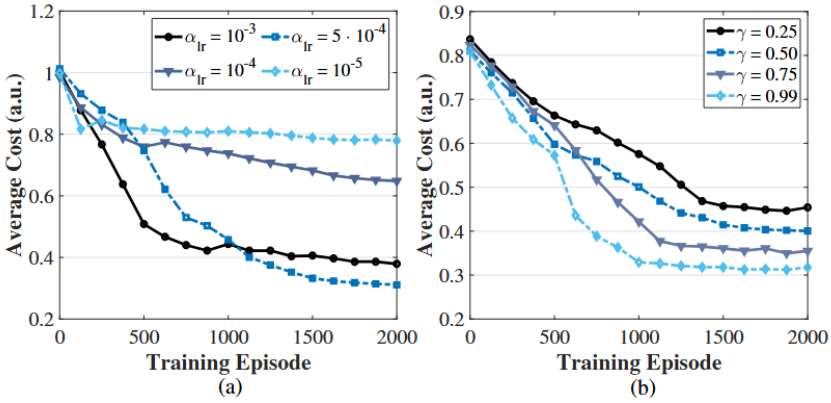


Figure 30: The learning curve of the average DECOFFEE agent for varying learning rate (panel a) and discount factor (panel b)

Figure 30 (a) evaluates the impact of the learning rate under heavy workload conditions (probability of task arrival is 0.7 per agent) and equal delay-energy weighting (weights for energy and delay awareness is 0.5). The results show that very small learning rates slow down convergence, whereas larger values may lead to unstable training. The best trade-off is achieved for 0.0001, which yields the lowest long-term cumulative cost. Figure 30 (b) examines the discount factor. The best performance is obtained for a value of 0.99, indicating that DECOFFEE benefits from emphasizing long-term system behavior. This is particularly relevant since offloading decisions influence not only immediate delay, but also future energy

consumption and queue evolution across the computing continuum. Therefore, the subsequent evaluations use the identified optimal learning configuration.

3.3.2.3 Clustered Federated Learning for Managed Wi-Fi

The pervasive deployment of Wi-Fi technology has established its critical role as a primary connectivity medium, particularly within residential and enterprise domains. This prominence is largely attributable to its inherent advantages, including superior indoor applicability, operational flexibility, and user-centric accessibility. Contemporary advancements in Wi-Fi standards, exemplified by Wi-Fi 7 and 8, are driven by the imperative to support an expanding array of demanding applications, such as immersive communications, digital twin implementations in manufacturing, and advanced healthcare solutions, all of which necessitate ultra-low latency and high reliability. Concomitantly, these technological evolutions have introduced considerable complexity into network management, thereby necessitating the development of sophisticated orchestration paradigms.

We advocate for the strategic integration and leveraging of CFL strategies within managed Wi-Fi solutions to significantly enhance prediction performances. To achieve this, we propose a novel CFL methodology that employs a sophisticated two-stage clustering algorithm. This algorithm is specifically designed to yield more informative and robust clusters compared to existing state-of-the-art approaches, thereby addressing the challenges posed by data heterogeneity across diverse APs.

In the first stage of our proposed method, a candidate set of clustering solutions is generated. These solutions are rigorously filtered based on a predefined minimum set of clustering criteria, ensuring that only high-quality and relevant groupings are considered for further analysis. This initial filtering step is crucial for pruning suboptimal solutions and focusing on those that exhibit foundational structural integrity. Following this, a subsequent and critical step involves performing a robust clustering decision. From the refined candidate set, we meticulously select the solution that maximizes the informativeness, quantified by the entropy, for the smallest cluster. This particular focus on the smallest cluster is a deliberate design choice, as these clusters are often more critical due to their inherent susceptibility to limited generalization capabilities, primarily stemming from the constrained number of AP models they encompass. Maximizing their informativeness ensures that even the most sparsely represented groups contribute meaningfully to the overall model accuracy and robustness.

The efficacy of this proposed CFL tool is rigorously evaluated using a comprehensive real-world dataset. This dataset is specifically utilized to address the complex problem of traffic load prediction within managed Wi-Fi systems, providing a realistic and challenging environment

for validation. The empirical results demonstrate that our proposal enjoys superior predictive performance when benchmarked against established state-of-the-art CFL strategies. Furthermore, our methodology exhibits tangible benefits in terms of operational efficiency, evidenced by significantly lower energy consumption and a substantially reduced communication footprint, making it a more sustainable and scalable solution for large-scale Wi-Fi network deployments.

3.3.2.3.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed CFL algorithm maps naturally onto the Unity architecture across two domains of the Unification Layer. On the access side, Wi-Fi APs reside in the Non-3GPP RAN Domain, which is specifically designed to manage and orchestrate IEEE 802.11 APs in multi-RAT deployments. This domain provides the infrastructure through which APs can expose traffic load telemetry and receive control instructions from the Infrastructure Domain Manager. Each AP locally trains a model on its own traffic observations and participates in the distributed learning process without transferring raw data to the cloud controller.

The cloud-side intelligence instead can be naturally hosted in the AI/ML Domain. Indeed, the AI/ML Domain is designed to incorporate distributed learning mechanisms such as federated learning, and to support privacy-preserving and scalable model training through these techniques. The CFL cloud controller logic -aggregating locally trained model updates, applying the two-stage entropy-based clustering to identify groups of APs sharing similar traffic distributions, and distributing cluster-specific models back to the APs for local inference - maps precisely onto the training, inference, and evaluation pipelines of this domain. Taken together, the AP-side and cloud-side components of CFL realise the MS-AE-DE-ACT closed-loop paradigm. Indeed, the APs act as the MS, the cloud-side aggregation and clustering constitute the AE, the selection and assignment of cluster-specific models can be done by the DE, and the deployment of those models for local inference at the APs can be done by the ACT. Finally, this end-to-end pipeline can be initiated through an ML intent issued by the IDMO in the Management and Orchestration Layer, with performance KPIs flowing back through the cross-domain service bus.

From a KPI perspective, the most direct contribution is KPI 43, Increase integrated network energy efficiency by a factor of 5 through distributed AI techniques. Cluster-specific models reduce memory and compute requirements compared to per-AP modelling. Exchanging compact model updates instead of raw traffic time series cuts control-plane traffic. More effective management is also more EE KPI#4, which calls for real-time reconfiguration of the IEEE 802.11 network domain in less than 10 ms from the moment a decision is taken. Meeting this target requires that decisions be prepared in advance rather than triggered by observed

failures: by providing accurate, AP-specific traffic load forecasts, CFL enables to anticipate demand changes and stage reconfiguration actions proactively, relieving the time pressure on the actuation path.

3.3.2.4 ML-Based Dynamic Network Routing

The proposed dynamic routing contribution builds on top of the Deep Predictive Interference-Aware Routing (DPIR) framework [77]. DPIR jointly considers traffic evolution, wireless interference, graph topology, route-level features, and utility-driven reinforcement learning. The objective is to select a routing profile for multiple concurrent flows that maximizes global network utility while accounting for interference coupling among the selected paths. The framework assumes that transmit power control is fixed and predetermined; hence, the main controllable decision variable is route selection.

DPIR advances beyond these approaches by combining four elements in a single closed-loop framework: (i) predictive traffic modeling using a Mamba-based sequence model, (ii) graph-native encoding of topology and wireless link state through message passing, (iii) explicit route-level representation using a path encoder, and (iv) reinforcement-learning-based route selection followed by local noisy-best-response refinement. This allows the routing policy to act proactively and to reason about interference-aware path diversity rather than only about path length or available capacity. The predictive metrics are adapted across multiple aspects and requirements.

3.3.2.4.1 System model

The network is represented as a wireless graph $G = (V, E)$, where V is the set of wireless nodes, and E is the set of wireless links. A set of active data flows $F = 1, \dots, N$ is considered. Each flow i has a source node s_i, a destination node d_i, and a traffic demand $D_i(t)$ at time t. For each flow, a finite set of k candidate routes is generated:

$$P_i = p_i^1, p_i^2, \dots, p_i^K.$$

The routing decision for flow i is the selection of one candidate route p_i in P_i. The joint routing profile is denoted by:

$$p = (p_1, p_2, \dots, p_N).$$

The key modeling aspect is the interfering-flow neighborhood. For a given selected route p_i, the set $N_i(p)$ contains the flows whose selected routes interfere with flow i. Therefore, the utility of flow i depends on both its own route and the selected routes of its interfering neighbors. This captures the fact that two routes may be individually attractive but jointly inefficient if they activate strongly interfering wireless links.

For each wireless link e, the achievable rate follows a Shannon-like expression:

$$R_e(p) = B_e \log 2\big(1 + SINR_e(p)\big),$$

where B_e is the link bandwidth. The signal-to-interference-plus-noise ratio is modeled as:

$$SINR_e(p) = S_e/\big(N_e + I_e(p)\big),$$

where S_e is the received signal power, N_e is the noise power, and $I_e(p)$ is the aggregate interference generated by other active links under the selected routing profile. The rate of flow i is determined by the bottleneck link on its selected path:

$$r_i(p) = \min_{e \in p_i} R_e(p).$$

The global objective is to maximize the sum of flow utilities:

$$\max i\, mize_p U(p) = \sum_{i=1}^{N} u_i\big(r_i(p)\big).$$

Different utility functions can be selected according to the desired network objective. For example, $u_i(r_i) = r_i$corresponds to sum-rate maximization, while $u_i(r_i) = \log(r_i)$ promotes proportional fairness among flows.

3.3.2.4.2 Algorithmic framework

The aldorithm operates periodically. The network update period is divided into smaller sub-slots, and a routing schedule is produced for the next update period. At the beginning of each decision cycle, the framework receives the topology graph, the set of active flows, the candidate routes for each flow, past traffic observations, and link-level features. The link feature vector may include normalized link capacity, estimated interference level, previous routing/action indicator, SINR, queue length, and link utilization. These features jointly describe the physical wireless state, the traffic state, and the recent routing state of the network.

A Mamba-based traffic forecasting module predicts future demand over the considered horizon from historical traffic volumes. The resulting demand forecast is included in the state representation, so that route selection is not purely reactive. This is important in bursty traffic scenarios, where a path that is acceptable at the current time may become congested or interference-limited shortly after the route is enforced.

The state representation is built using two neural encoders. First, a graph encoder based on message passing processes the topology and link features and produces link/node embeddings that capture multi-hop dependencies, congestion patterns, and wireless interference context. Second, a bidirectional GRU path encoder processes the ordered

sequence of embeddings along each candidate path. This produces a path-level representation that preserves information about bottleneck links, cumulative load, and interference exposure along the route. The use of path-level encoding is important because the routing action selects a complete path rather than a single next hop.

The policy network receives the candidate-path embeddings and produces a probability distribution over the K allowed paths of each flow through a softmax layer. The full joint action space has size K^N, which becomes intractable as the number of flows increases. DPIR therefore uses sequential allocation: after the route of one flow is selected, the network state is updated before selecting the route of the next flow. This allows the policy to account for the incremental load and interference introduced by earlier route choices within the same decision cycle.

Training is performed with a REINFORCE-with-baseline reinforcement learning procedure. The reward is the global utility achieved by the selected routing profile. The baseline reduces gradient variance and improves training stability. The policy is therefore encouraged to increase the probability of route profiles that improve sum-rate or proportional fairness, and to reduce the probability of route profiles that create severe bottlenecks, strong interference, or unfair capacity allocation.

After the neural policy generates an initial routing profile, DPIR applies a refinement stage based on Boltzmann noisy best response. In this stage, flows may probabilistically revise their selected route according to the utility improvement that would result from switching to another candidate path. The Boltzmann temperature controls the exploration level. This local refinement step is useful because the interference-aware utility of one route depends on the selections of other flows, and local revisions can improve the final routing profile without exhaustive enumeration of all K^N combinations.

To control complexity and improve the quality of the action space, DPIR includes a path-pruning stage before policy inference. Candidate paths are generated using a modified Dijkstra procedure that promotes spatial diversity. The goal is not only to provide short paths, but also to provide alternatives with different interference footprints. This increases the probability that the policy can find route combinations that reduce mutual interference among concurrent flows.

The deep predictive interference-aware routing framework is shown in Figure 31.

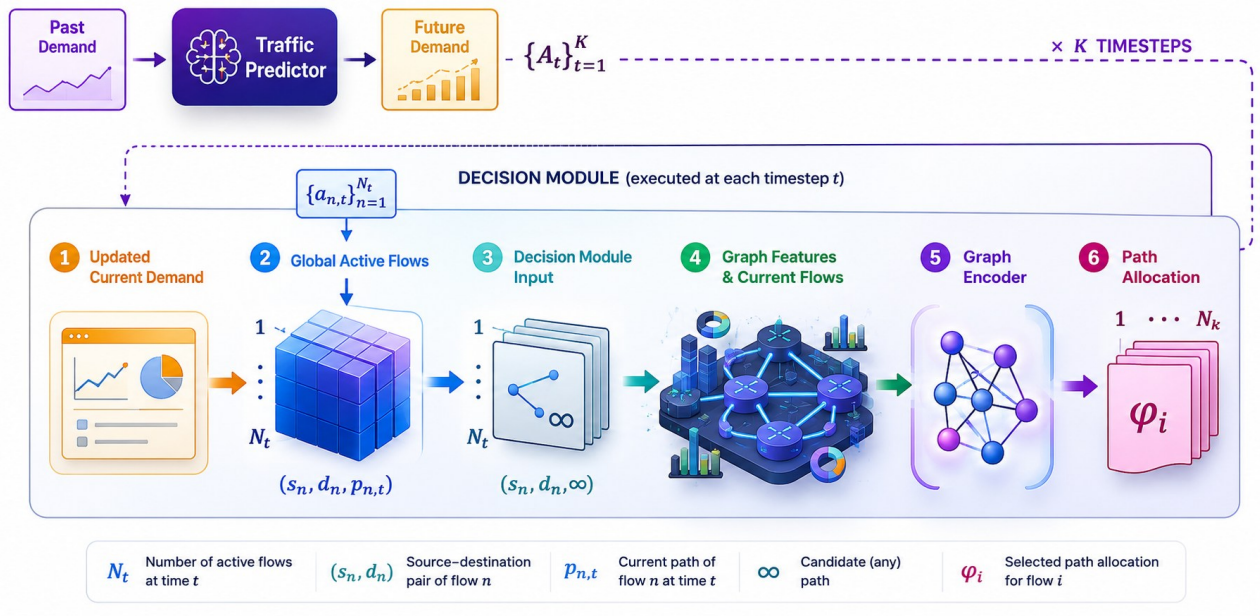


Figure 31: Deep Predictive Interference-Aware Routing framework

3.3.2.4.3 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The algorithm targets the following KPIs:

- KPI#7. Cross-domain (cellular – IEEE 802.11) end-to-end latency smaller than 20 ms dynamic-routing contributes to achieving cross-domain end-to-end latency below 20 ms by selecting, for each flow, paths that minimize predicted delay across cellular and IEEE 802.11 transport segments. By incorporating link delay, queue occupancy, utilization, interference, and forecast traffic demand into the routing state, the policy proactively avoids paths expected to become congested or interference-limited. A latency-aware utility function or SLA constraint can prioritize flows whose predicted end-to-end delay approaches the 20 ms threshold, thereby reducing the transport-domain contribution to the overall cross-domain latency budget.
- KPI#8. Cross-domain jitter of max of 2 ms, with less than 10% of the packets missing deadline: The routing mechanism contributes to the cross-domain jitter target by incorporating delay variation, queue dynamics, interference exposure, and predicted traffic demand into the route-selection utility. It preferentially selects paths with stable capacity and lower variability, avoiding routes likely to experience congestion or rapidly changing interference. By penalizing predicted jitter above 2 ms and packet-delay-budget violations within the SLA-aware reward function, the solution reduces the proportion of packets missing their deadline and supports the requirement that this remains below 10%.
- KPI#16. Reduce radio link power consumption up to 30% per link on average across a transport path (X-haul) due to full network coordinated load balancing, improved link

utilization and distributed AI algorithms (between components O-RU, O-DU, O-CU links in Open RAN): The solution contributes to reducing average radio-link power consumption across the X-haul path by distributing O-RU, O-DU and O-CU traffic over routes that achieve the required SLA with lower energy cost and more balanced link utilization. An energy-aware utility function can penalize the activation of highly loaded or inefficient links and favour routing profiles that consolidate traffic onto energy-efficient transport resources, enabling lightly used links to enter low-power or sleep modes where supported. Since DPIR coordinates route selection across concurrent flows and accounts for interference, it can reduce unnecessary retransmissions and avoid energy-intensive congestion conditions, supporting the targeted reduction in per-link power consumption.

- KPI#17. Reduce CAPEX and OPEX with shared resource: The solution supports CAPEX and OPEX reduction by enabling multiple services, RAN functions, and traffic classes to share the same transport resources more efficiently. Through coordinated, SLA-aware flow allocation, it increases utilization of existing links, avoids unnecessary capacity overprovisioning, and can defer infrastructure upgrades. Its automated adaptation to traffic, congestion, and interference also reduces operational effort associated with manual routing reconfiguration and reactive fault or performance management.
- KPI#20. Increase link utilization from 40% to 80%: The solution supports increased link utilization by continuously steering concurrent flows toward underused yet SLA-compliant transport paths, rather than concentrating traffic on the same preferred links. Its predictive traffic model anticipates demand growth, while the interference-aware and graph-based policy evaluates the residual capacity and expected quality of alternative routes. By using a utilization-balancing objective within the routing utility, the solution can redistribute load across available X-haul resources and move average link utilization toward the targeted increase from 40% to 80%, without violating latency, jitter, or reliability constraints.
- KPI#38. Reduce Operation expenditure (OPEX) by 30% due to the automation of service management and increase in link utilization (via planning annual frequency licensing): Automating SLA-aware routing and service-performance adaptation, reducing manual planning, reconfiguration, and troubleshooting effort. Its traffic forecasts and link-utilization analytics also provide evidence for annual frequency-licensing and capacity planning, helping operators reserve spectrum and transport resources according to expected demand rather than peak overprovisioning. By increasing utilization of already licensed links and balancing traffic across available resources, the solution can reduce recurring operational costs and support the targeted 30% OPEX reduction.

- KPI#42. DT-validated AI algorithms that do not break the time-sensitive operation when deployed in 99% of the outputs: The solution can be validated in a Digital Twin before deployment by replaying representative topology, traffic, interference, and failure scenarios and assessing the effect of each routing decision on time-sensitive SLA metrics. Only policies that consistently satisfy latency, jitter, deadline-miss, and stability constraints are released to the operational network, while unsafe or low-confidence actions are rejected or replaced by a baseline route. This DT-based validation and guarded deployment process supports the target that at least 99% of AI routing outputs do not disrupt time-sensitive operation.

Within the UNITY-6G architecture, DPIR maps naturally to the MS-AE-DE-ACT closed-loop pattern. The Monitoring System provides topology, traffic, queue, utilization, SINR, and interference-related observations. The Analytics Engine hosts the demand prediction and state-encoding functions. The Decision Engine executes the learned routing policy and selects the route profile according to the selected utility objective. The Actuator applies the routing schedule through the relevant control interface. Historical routing decisions, utility values, model versions, and telemetry traces can be stored in the Data Continuum for offline validation, retraining, and explainability.

3.3.2.4.4 Preliminary results

The output of DPIR is a per-flow routing schedule for the next network update period. During execution, routes are applied in a decentralized manner over the update sub-slots. Runtime telemetry, including measured SINR, utilization, queue state, achieved rates, and previous route indicators, is fed back into the following decision cycle. This forms a closed loop composed of monitoring, traffic prediction, interference-aware route selection, execution, and telemetry feedback.

The preliminary evaluation setup considers random wireless network topologies with 40 nodes and 70 edges and multiple randomly generated flows. Each flow is assigned a set of candidate routes. The main evaluation metrics are expected to include aggregate utility, sum-rate, proportional fairness, bottleneck flow rate, interference level, and routing decision time. DPIR should be compared against baselines such as shortest-path routing, load-aware routing, and non-predictive learning-based routing. Additional ablation studies should isolate the contribution of the Mamba demand predictor, the graph encoder, the path encoder, and the Boltzmann refinement step.

3.3.2.5 Network Recovery & Disaster Resilience via Multi-Agent Reinforcement Learning

The proposed solution applies Multi-Agent Reinforcement Learning (MARL) to enable self-healing behaviour in the surviving access and transport infrastructure and it is shown in Figure 32. Each programmable gNB and wireless transport relay node is modelled as an autonomous agent. Each agent observes its local radio and network environment, selects local recovery actions, and contributes to a shared network-level objective: restoring the maximum possible fraction of critical UE communication flows during Island Mode. The agents are trained over multiple simulated disaster scenarios so that the resulting policy generalises across different severance types, surviving topologies, and traffic distributions.

The MARL approach is particularly suitable for this problem because post-disaster recovery is inherently distributed and partially observable. No single surviving node can be assumed to possess a complete global view of the damaged network, and no central controller can be assumed to be reachable. Instead, each node must infer useful recovery actions from its own local measurements and limited neighbourhood information. Through cooperative training, the agents learn behaviours such as activating relay mode, adjusting transmission power, prioritising emergency traffic, allocating physical resource blocks, selecting scheduling modes, and supporting ad-hoc relay paths between isolated cells.

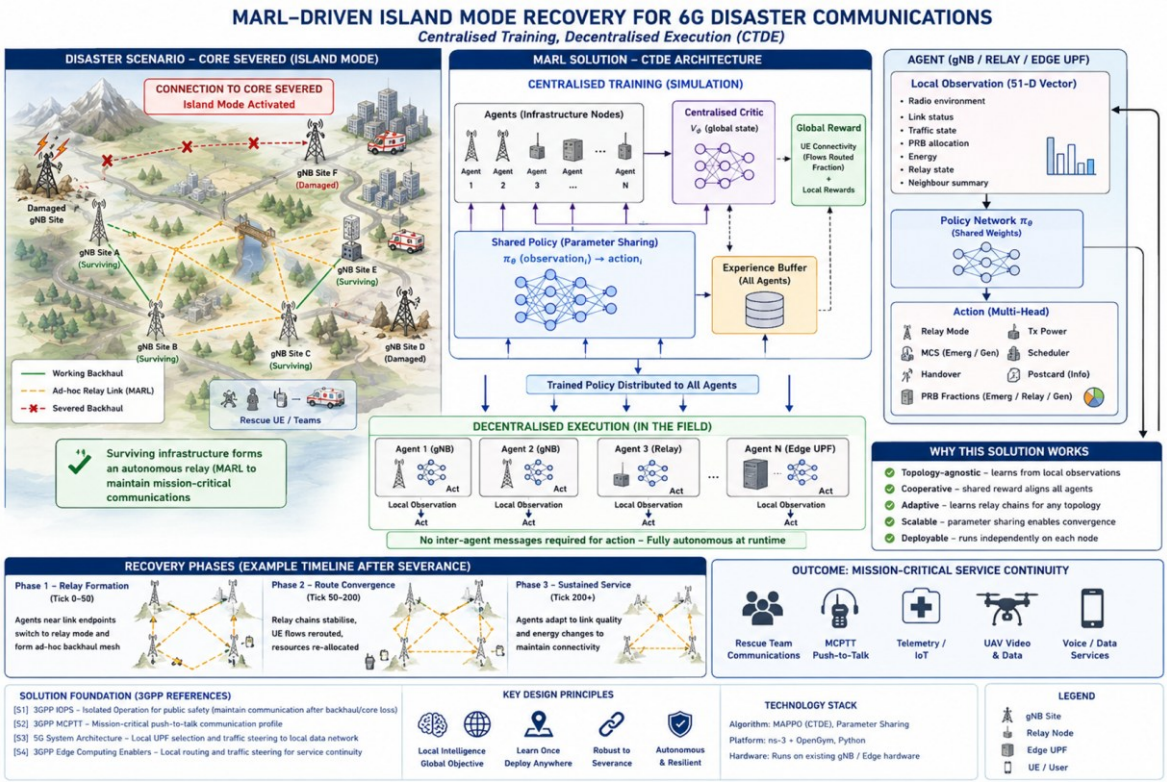


Figure 32: Network Recovery & Disaster Resilience via Multi-Agent Reinforcement Learning

3.3.2.5.1 System model

The proposed system is modelled as a post-disaster, partially connected 5G/6G access and transport network in which a subset of the infrastructure remains physically operational, but

the normal connectivity to the core network, centralised control functions, and remote service anchors may be lost. The model captures a representative integrated 6G environment composed of UEs, gNBs, wireless transport relay nodes, edge or local breakout functions, and core-network elements. The purpose of the model is to evaluate how surviving programmable infrastructure nodes can autonomously reconfigure themselves during a disaster in order to restore local and inter-cell communication, with particular emphasis on emergency and public-safety traffic.

The network is represented as a time-varying graph:

$$G_t = (V_t, E_t)$$

where V_t is the set of operational nodes at time t , and E_t is the set of available communication links at time t . The node set is partitioned into functional classes:

$$V_t = V_t^{UE} \cup V_t^{gNB} \cup V_t^{TR} \cup V_t^{Edge} \cup V_t^{Core}$$

where V_t^{UE} denotes the set of user equipment nodes, V_t^{gNB} denotes the set of surviving gNBs, V_t^{TR} denotes the set of wireless transport relay nodes,V_t^{Edge} denotes the set of edge or local breakout components, and V_t^{Core} denotes the set of core-network components, including UPF or gNB-core nodes. The transport relay set V_t^{TR} includes relay nodes that provide wireless backhaul or IAB-like relay functionality, and are therefore capable of forming temporary transport paths between otherwise disconnected radio cells.

A disaster event is represented as a disruption process that removes or degrades a subset of nodes and links from the graph. Let D_t denote the disaster state at time t . Before the disaster, the network operates in a normal connected state, where most access nodes have at least one valid route to the core or to a service anchor. At the severance time t_s, the disaster process affects the graph by disabling a subset of nodes $V_t^{fail}$and a subset of links E_t^{fail}. The post-disaster operational graph is therefore:

$$V_t = V_0 \, V_t^{fail}$$

$$E_t = E_0 \, E_t^{fail}$$

for $t \geq t_s$, where V_0 and E_0 denote the pre-disaster node and link sets. The failed elements may include core nodes, fibre backhaul links, wireless backhaul links, relay nodes, edge-UPF nodes, or entire geographic zones. This allows the model to represent different disaster patterns, including full core severance, partial core failure, zone-level outage, and cascading failures in which additional nodes or links fail after the initial disaster event.

The post-severance state is referred to as Island Mode. Island Mode is triggered when one or more groups of surviving infrastructure nodes become disconnected from the required core-network or service component. More formally, let $P_t(i,j)$ denote the existence of a valid communication path between node i and node j at time t. A surviving access node $i \in V_t^{gNB}$ is considered to be in Island Mode if there is no valid path from i to the required core or service anchor:

$$P_t(i,c) = 0 \; \forall \; c \; \in V_t^{Core}$$

or, in cases where local edge service continuity is considered, if no valid path exists from i to an acceptable service anchor in $V_t^{Core} \cup V_t^{Edge}$. The graph may then be decomposed into disconnected components:

$$C_t = C_t^1, C_t^2, \dots, C_t^K$$

where each C_t^k represents a connected island of surviving nodes. The recovery problem is to use the surviving gNBs and transport relay nodes to reduce the effective fragmentation of the graph and to restore routability for as many critical UE flows as possible.

Each programmable infrastructure node that can participate in recovery is modelled as an autonomous MARL agent. The agent set is defined as:

$$A = a_1, a_2, \dots, a_N$$

where each agent a_i is associated with a surviving gNB or transport relay node. The agents operate at discrete simulation ticks. At each tick, each agent observes local and neighbourhood-level information, selects a local action, and contributes to a shared network-level recovery objective. The model does not assume that each agent has access to a complete global topology during deployment. This is important because, in a disaster scenario, the centralised management system, SDN controller, or orchestration platform may be disconnected from part of the damaged network.

The local observation of agent a_i at time t is represented as:

$$o_i(t) = [m_t, e_i(t), p_i(t), n_i(t), c_i(t), \tau_t, h_i(t)]$$

where m_t is the Island Mode indicator, $e_i(t)$ represents the local energy or battery state of the node, $p_i(t)$ represents PHY/MAC state information, $n_i(t)$ represents aggregated neighbour radio information, $c_i(t)$ represents local and inferred connectivity state, τ_t represents temporal information, and $h_i(t)$ represents optional coordination or policy hints. In the implemented

system, this observation is encoded as a fixed-dimensional local state vector, enabling the same policy architecture to be applied across different node roles and topological positions.

The Island Mode indicator m_t is a binary variable that indicates whether the network, or the local connected component of the agent, is operating under post-severance conditions. The energy state $e_i(t)$ includes the state of charge or available power budget of the node, which is important because sustained relay operation may consume significant energy. The PHY/MAC state $p_i(t)$ includes local queue lengths, PRB allocation, interference measurements, estimated capacity limits, scheduler state, and local traffic load. The neighbour radio summary $n_i(t)$ includes compact information about neighbouring nodes, such as observed SINR, received signal quality, transmit power levels, relay participation state, and limited coordination information received through local broadcasts or postcard-like summaries. The connectivity state $c_i(t)$ includes information about UE reachability, known or inferred relay paths, local island size, and the number or fraction of flows that appear routable from the local perspective. The temporal state τ_t includes the current tick and the elapsed time since the severance event. Optional hints $h_i(t)$ may represent last-known management instructions, local policy constraints, or coordination information received before or during Island Mode.

Each agent selects a structured multi-head action at every tick. The action of agent a_i at time t is represented as:

$$a_i(t) = \left[a_i^{relay}, a_i^{power}, a_i^{mcs_{emg}}, a_i^{mcs_{gen}}, a_i^{sched}, a_i^{ho}, a_i^{postcard}, a_i^{prb}\right]$$

where a_i^{relay} selects the relay operating mode, a_i^{power} selects the transmit-power adjustment, $a_i^{mcs_{emg}}$ selects the modulation and coding scheme for emergency traffic, $a_i^{mcs_{gen}}$ selects the modulation and coding scheme for general traffic, a_i^{sched} selects the scheduling policy, a_i^{ho} indicates whether handover or rerouting support should be triggered, $a_i^{postcard}$ indicates whether the node broadcasts a compact local coordination summary, and a_i^{prb} defines the allocation of physical resource blocks among emergency traffic, relay traffic, and general traffic.

The relay-mode action allows a node to remain inactive as a relay, operate in a normal relay mode, support local rerouting, boost relay capacity, or assist device-to-device and peer-relay communication where applicable. The transmit-power action allows the agent to increase or decrease its transmit power within allowed operational limits. This is required because higher power can improve relay reachability but may also increase interference to other active links. The MCS actions allow differentiated treatment of emergency and general traffic. For example, emergency flows may require more robust coding under poor channel conditions, while general

traffic may use more aggressive coding when the link quality is sufficient. The scheduling action allows the node to choose between policies such as round-robin, proportional fairness, emergency-first scheduling, or maximum-SINR scheduling. The PRB allocation action is a continuous or discretised resource split that determines how much radio resource is assigned to emergency, relay, and general traffic classes.

The joint action of all agents is defined as:

$$a(t) = [a_1(t), a_2(t), \dots, a_N(t)]$$

and the global network state evolves according to the current graph, the disaster process, the traffic process, and the joint actions selected by the agents. The transition from one state to the next can be expressed as:

$$s(t+1) = f(s(t), a(t), D_t, X_t)$$

where $s(t)$ is the global simulation state, $a(t)$is the joint action, $D_t$is the disaster state, and $X_t$represents exogenous processes such as UE mobility, traffic arrivals, channel variation, weather effects, additional failures, or battery depletion. During training, the simulator has access to $s(t)$, while during deployment each agent only receives its own local observation $o_i(t)$.

Wireless transport links are modelled using radio-aware link-quality and capacity estimates. A candidate wireless link between nodes i and j is denoted as $e_{ij} = (i, j)$. The link is available at time t only if both endpoints are operational, the channel conditions are sufficient, and the link is not removed by the disaster process. The achievable capacity of a wireless relay link is modelled using a Shannon-like formulation:

$$C_{ij}(t) = B_{ij}(t) \log_2 \left(1 + SINR_{ij}(t)\right)$$

where $C_{ij}(t)$ is the achievable link capacity, $B_{ij}(t)$ is the allocated bandwidth, and $SINR_{ij}(t)$ is the signal-to-interference-plus-noise ratio on the link. The SINR can be expressed as:

$$SINR_{ij}(t) = P_i(t)G_{ij}(t) / \left(N_0 B_{ij}(t) + k \sum_k P_k(t) G_{kj}(t)\right)$$

where $P_i(t)$ is the transmit power of node i, $G_{ij}(t)$ is the channel gain between nodes i and j, N_0 is the noise spectral density, and the summation term represents interference from other simultaneously transmitting nodes k. This formulation captures the fact that relay activation is not always beneficial. Activating more relays may create additional paths, but it may also increase interference, consume more energy, and reduce the effective capacity of

neighbouring links. The agents must therefore learn when relay activation improves the network and when it degrades the overall recovery objective.

Traffic is modelled as a set of active UE communication flows:

$$F_t = f_1, f_2, \dots, f_M$$

Each flow f_m is defined by a source node, a destination node or service endpoint, a traffic class, a priority weight, a bandwidth or rate demand, and a routing requirement. This can be written as:

$$f_m = \left(u_m^{src}, u_m^{dst}, \kappa_m, w_m, r_m, d_m\right)$$

where u_m^{src} is the source UE or traffic origin, u_m^{dst} is the destination UE or service endpoint, κ_m is the traffic class, w_m is the priority weight, r_m is the required rate, and d_m represents delay or deadline requirements where applicable. Emergency, rescue, and mission-critical flows are assigned higher priority weights than general UE traffic. This enables the recovery policy to prioritise life-safety communication and public-safety coordination when the post-disaster network cannot serve all flows simultaneously.

A flow is considered served at time tt if there exists at least one valid path through the current access and relay topology from its source to its destination, and if the bottleneck capacity along that path is sufficient to support the required traffic. Let $path_m(t)$ denote a selected feasible path for flow f_m. The bottleneck capacity of that path is:

$$C_m^{path}(t) = \min_{(i,j)\in path_m(t)} C_{ij}(t)$$

The flow is served if:

$$C_m^{path}(t) \geq r_m$$

and if all nodes and links along the path are operational. The set of served flows at time t is denoted by F_t^{served}. The main system-level connectivity metric is the routed-flow ratio:

$$\rho_t = \left|F_t^{served}\right| / |F_t|$$

For emergency-aware evaluation, a weighted routed-flow ratio can also be used:

$$\rho_t^w = \left(\sum_m w_m \ 1_m^{served}(t)\right) / (\Sigma_m w_m)$$

where $1_m^{served}(t)$ is equal to 1 if flow f_m is served at time t, and 0 otherwise. The weighted formulation gives greater importance to emergency and rescue flows, which is more suitable for disaster-recovery evaluation than an unweighted average over all UE flows.

The global recovery objective is to maximise post-severance service continuity over the Island Mode interval. Let T_I denote the set of ticks during which the network is in Island Mode. The objective is to find a multi-agent policy π that maximises routed connectivity while limiting interference, energy consumption, queueing, and inefficient relay usage. This can be expressed as:

$$\max_{\pi} E\left[\Sigma_{t\in T_I}\gamma^t\left(w_c\rho_t^w - w_I I_t - w_E E_t - w_Q Q_t - w_R R_t^{idle}\right)\right]$$

where γ is the discount factor, ρ_t^w is the weighted routed-flow ratio, I_t is the interference cost, E_t is the energy or battery cost, Q_t is the queueing or service-degradation cost, R_t^{idle} is the cost of inefficient or unnecessary relay activation, and w_c, w_I, w_E, w_Q, and w_R are weighting coefficients. The first term rewards restored connectivity, while the remaining terms prevent the agents from producing unstable or inefficient recovery behaviour.

In the implemented reinforcement-learning formulation, the global objective is operationalised through a shaped reward that combines network-wide and local learning signals. The dominant reward component is the fraction of UE flows that remain routable during Island Mode. Additional positive terms encourage useful relay participation and successful emergency-flow support. Penalty terms discourage unnecessary relay activation, excessive interference, high energy consumption, and queue buildup. A representative reward structure is:

$$R(t) = \alpha\rho_t^w + \beta N_{relay}(t) + \delta F_{emg^{served}}(t) - \lambda_I I_t - \lambda_E E_t - \lambda_Q Q_t - \lambda_F\left(N_{frag}(t) - 1\right)$$

where $N_{relay}(t)$ is the number of useful active relay participants, $F_{emg^{served}}(t)$ is the number or ratio of served emergency flows, $N_{frag}(t)$ is the number of disconnected island fragments, and α, β, δ, λ_I, λ_E, λ_Q, and λ_F are design coefficients. The purpose of this reward design is to ensure that the agents primarily optimise connectivity restoration, while still receiving intermediate feedback that helps them learn how to create relay paths before full recovery is achieved.

The MARL formulation follows the Centralised Training and Decentralised Execution paradigm. During training, a centralised critic has access to broader simulation information, including global connectivity, full topology state, routed-flow status, and the joint actions of the agents. This critic is used to estimate the value of the joint behaviour and to reduce the variance of the policy-gradient update. During deployment, each agent executes the learned

policy using only its own local observation. This is essential for the target disaster scenario, because centralised control-plane connectivity may be lost exactly when the recovery policy is needed.

The policy of each agent is represented as:

$$a_i(t) \sim \pi_\theta\left(a_i(t) \middle| o_i(t)\right)$$

where π_θ is the learned policy parameterised by θ. In the current implementation, parameter sharing is used, meaning that all gNB and transport-relay agents use the same policy network. Parameter sharing improves sample efficiency because experience collected from all agents contributes to the same policy update. It also improves generalisation because the learned policy must operate across different node positions, roles, and surviving topologies. This is important for disaster recovery, since the exact set of surviving nodes and links is not known in advance.

The critic is represented as:

$$V_\varphi\left(s(t)\right)$$

where V_φ is the value function parameterised by φ. During training, the critic can use global state information from the simulator. During deployment, the critic is not required for action selection. The trained policy can therefore be executed locally at each infrastructure node, without requiring a central server, continuous inter-agent communication, or real-time access to the full network state.

The training objective is to maximise the expected discounted return:

$$J(\theta) = E[\Sigma_{t=0}^{T} \gamma^t R(t)]$$

where $R(t)$ is the shaped recovery reward. In the MAPPO-based implementation, policy updates are constrained using the PPO clipped objective in order to avoid destructive updates and improve training stability. The probability ratio between the updated policy and the previous policy is:

$$r_t(\theta) = \pi_\theta(a_t|o_t)/\pi_{\theta_{old}}(a_t|o_t)$$

and the clipped policy objective is:

$$L_{CLIP}(\theta) = E_t[\min(r_t(\theta)A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t)]$$

where A_t is the estimated advantage and ε is the PPO clipping parameter. Generalised Advantage Estimation may be used to compute low-variance advantage estimates over the

collected rollout data. The critic is trained to predict the expected return, while the actor is trained to select actions that improve the recovery reward without excessive policy shifts.

The system evolves in three main operational phases. In the first phase, immediately after severance detection, Island Mode is triggered and the agents enter rapid relay-formation behaviour. During this phase, nearby gNBs and transport relay nodes activate relay or bridge modes, adjust their transmission parameters, and begin creating temporary paths between isolated access nodes. In the second phase, the relay topology converges. Relay chains stabilise, UE flows are mapped through the emergent mesh, and radio resources are reallocated according to traffic priority and channel conditions. In the third phase, sustained service is maintained. The agents continue adapting relay assignments, scheduling, PRB allocation, MCS selection, and transmit power as link quality, traffic demand, interference, battery level, and additional failures evolve.

This system model is naturally aligned with cooperative MARL because the recovery task is distributed, dynamic, partially observable, and topology-dependent. The agents share a common network-level recovery goal, but each agent acts from a limited local view. The learned policy therefore captures local behaviours that collectively produce global recovery, such as forming relay chains, preserving emergency traffic, reducing island fragmentation, and avoiding excessive interference. The approach is also aligned with side-link and locally assisted communication principles, where communication can be preserved even when conventional base-station or core-network connectivity is unavailable [78]. In the proposed model, this principle is extended to infrastructure-assisted Island Mode recovery, where surviving gNBs and transport relay nodes cooperate to form temporary communication structures for emergency and local service continuity.

3.3.2.5.2 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The following KPIs are considered:

- KPI#19. Reduce time to manage network resources and management overhead and the reaction time by taking proactive actions in the network (time from identification to resolution via appropriate reconfigurations) with integrated design:. The MARL-based recovery solution reduces resource-management overhead and recovery time by enabling surviving gNBs and transport relays to detect Island Mode locally and execute coordinated reconfigurations in parallel. Using local observations and learned policies, agents can rapidly activate relay paths, adapt power, MCS, scheduling and PRB allocation, and prioritise critical flows without waiting for a central controller or manual intervention. This integrated, distributed decision process shortens the time from disruption identification to service restoration while limiting control-plane signalling.

- KPI#31. Reduce service recovery time below 180s: The MARL-based Island Mode recovery mechanism supports service recovery within 180 seconds by allowing surviving gNBs and transport relays to initiate local recovery actions immediately after severance detection. Agents execute relay activation, neighbour-assisted path formation, traffic prioritisation, radio-resource allocation, and link-parameter adaptation in parallel, without waiting for centralised orchestration or manual intervention. Offline training across representative disaster scenarios enables the policy to select effective recovery actions rapidly, reducing convergence time toward restoration of critical communications.

3.3.3 Forecast and prediction

3.3.3.1 Hy-DREAM for Anomaly Detection

Maintaining service level agreements through proactive measures necessitates the identification of anomalies early enough to intercept violations regarding throughput, loss, and latency, which facilitates remediation while it can still have a meaningful impact.

Because anomalies are often evolving, diverse, and not present during initial training phases, network operators require detection systems that provide high trustworthiness for automation through low false positive rates, sufficient speed to allow for fast preventive action, and be lightweight enough to scale across numerous network slices and KPIs.

To address this, we proposed a dual-model framework, that improves robustness to previously unseen anomalies and supports low-latency, lightweight detection suitable for closed-loop, proactive assurance.

DREAM (Dual foREcAsting Model for Network Anomaly Detection) is a dual-model framework designed to enhance both the speed and accuracy of anomaly detection within time series data by training two distinct forecasting models: Model NB, which captures expected behavior from normal sequences, and Model MB, which uses mixed sequences containing unlabeled anomalies. Detection is achieved by comparing the outputs of these two models and identifying segments where their divergence surpasses a set threshold. By predicting major deviations from normal states before they fully manifest, this dual-model approach minimizes detection latency and offers a flexible response to the varied dynamics of real-world anomalies.

Despite these advancements, the framework can still encounter difficulties with entirely novel anomalies, whereas traditional forecasting often catches a wider spectrum of issues by comparing predictions directly to actual observations. On the other hand, DREAM can

generate these traditional predictions without increasing the computational burden, as comparing the normal-data model's forecast to the ground truth provides an additional indicator for free. This synergy led to a hybrid approach that combines both results to maximize coverage while maintaining high precision and speed.

To execute this, HyDREAM was developed as a hybrid framework as showcased in Figure 33 using a rule-based module to merge these complementary indicators into a unified decision. It utilizes the existing components of the DREAM architecture at no extra cost, generating a traditional anomaly score by comparing Model ND's predictions directly to the observed data. Rather than using simple averaging which might slow down the response, HyDREAM employs a rule-based module that makes immediate decisions when both models agree. If they disagree, it calculates a confidence score based on the distance to their respective thresholds and defers to the model with the higher relative confidence.

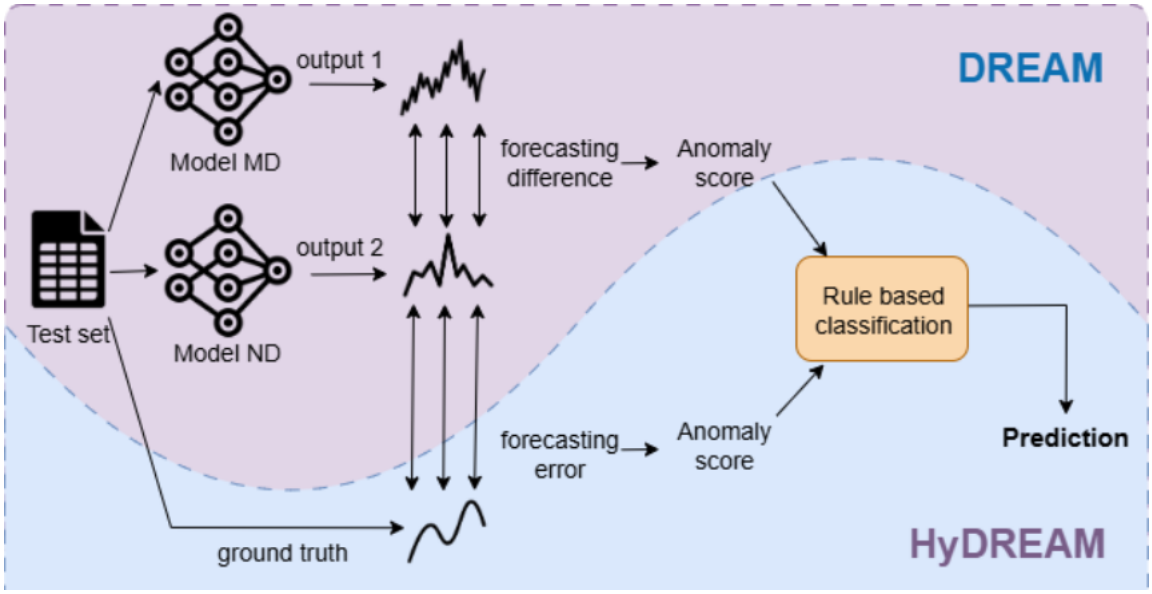


Figure 33: Architecture of the DREAM and HyDREAM approach.

3.3.3.1.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed model aims to achieve KPI#31, which sets the objective of reducing service recovery time to below 180 seconds. This is a critical metric for ensuring high-quality service and minimizing downtime in 5G networks. The initiatives Dream and HyDream focus on enhancing the speed and accuracy of anomaly detection, which is a fundamental step before initiating any recovery procedures.

A key challenge in network management is the timely identification of issues, as delays in detection can significantly extend recovery times and impact user experience. Our approach emphasizes rapid detection of anomalies, enabling faster response and minimizing service disruption.

In the preliminary results, we will demonstrate that the Delay Rate was successfully reduced to below 0.5%, indicating a high level of precision in anomaly detection. To achieve this, we propose a detection time of approximately 300 milliseconds within the 5G dataset, which represents a 60% improvement in speed compared to conventional solutions. This rapid detection capability is crucial for meeting the 180-second recovery window.

This algorithm will be integrated into the AI/ML subsystem domain in the UNITY-6G architecture. Our ongoing efforts are focused on further optimizing the detection algorithms to reduce the detection time even more, thereby ensuring that anomalies are identified and addressed within the desired timeframe. Ultimately, this will enable a more resilient and efficient 5G network, capable of maintaining high service quality and customer satisfaction even in complex and dense network environments.

3.3.3.1.2 Preliminary results

We evaluated the proposed framework using three multivariate datasets: SMD and PSM, which monitor server machine metrics, and 5G3E, a large-scale dataset capturing thousands of features from a 5G network environment. The analysis utilized two specific metrics to account for detection speed: the Delay Rate, which measures the percentage of an anomaly's duration that passes before it is detected, and the F-rate, which provides a harmonic mean of precision and detection timeliness. DLinear was chosen as the forecasting model due to its efficiency and superior speed compared to LSTM and transformer-based alternatives. The experiments were conducted across a threshold range from 1% to 99% to find the optimal balance for each method. DREAM and HyDREAM utilize identical hyperparameters and the same evaluation framework to ensure a fair comparison.

Table 2 **Error! Reference source not found.**compares the performance of Traditional, DREAM, and HyDREAM models across the SMD, PSM, and 5G3E datasets.

Table 2: Performance evaluation of models

Model	SMD F1-score	SMD F-rate	PSM F1-score	PSM F-rate	5G3E F1-score	5G3E F-rate
Traditional	29.29	36.09	53.28	62.03	77.42	86.13
DREAM	41.22	**52.77**	**54.03**	72,42	96.59	98.64
HyDREAM	**41.73**	51.43	54.02	**73.95**	**96.62**	**98.85**

Table 3 details the Delay Rate (DR), representing the average percentage of an anomaly duration that has elapsed by the time of detection, and its corresponding precision. A lower DR value indicates faster detection.

Table 3: Delay Rate (DR) of models

Model	SMD DR (%)	SMD Precision	PSM DR (%)	PSM Precision	5G3E DR (%)	5G3E Precision
Traditional	56.81	31.00	36.61	60.73	3.87	78.02
DREAM	54.13	62.13	33.75	79.87	0.37	97.68
HyDREAM	55.91	61.71	32.50	81.04	0.35	98.06

The data demonstrates that DREAM and HyDREAM consistently outperform traditional methods in both timeliness and accuracy across all tested environments. Particularly in the 5G3E dataset, both models achieved Delay Rates below 0.5% with nearly 98% precision, whereas traditional methods were significantly slower and less precise. This indicates that the dual-model strategy successfully reduces detection delay and performing up to ten times faster than standard approaches.

In conclusion, both DREAM and HyDREAM provide improvements for network anomaly detection through high precision and rapid response times. While DREAM establishes a robust dual-forecasting foundation, HyDREAM offers a balanced solution that captures the strengths of both dual-forecasting and traditional methods. This synergy allows HyDREAM to address the challenge of novel anomalies effectively, such as identifying packet loss incidents in the 5G3E dataset that were missing from the training set faster than other methods, making both models practical tools for ensuring reliable network service.

3.3.3.2 Radio Failure Prediction and Localized Recovery Estimation for Transport Wireless Links

The proposed approach is context-aware and fault-source-aware: it combines radio PMs, internal device-health indicators, alarms, topology data, traffic behavior, operational history, and external weather data to infer both the likelihood of failure and the most plausible reason for the predicted condition. Traditional threshold-based monitoring is useful for detecting that a monitored metric has crossed a configured limit, but it is mostly reactive. It provides limited support for identifying whether the root cause is external weather, internal hardware degradation, site-level equipment behavior, traffic/topology stress, or a compound event. It also provides limited support for estimating time-to-failure, expected recovery duration, or whether a field intervention is required. In a 6G context, where the transport domain is expected to support AI-native, intent-driven and resilient operation, the management system requires a predictive layer that can reason over heterogeneous evidence before the failure becomes service impacting. The application, therefore, targets a predictive-maintenance and resilience function for the transport segment. Its objective is not merely to detect that a link has failed, but to anticipate failure, classify the likely failure reason, estimate the expected time-to-

failure and recovery time, and support a decision on whether the condition can be tolerated, automatically mitigated, monitored, remotely handled, or escalated for field intervention.

3.3.3.2.1 System model

The system is modelled as a set of transport radios and radio links operating over a physical and logical transport topology. Let $G = (V, E)$ denote the transport graph, where V is the set of radio nodes, sites, or managed transport entities, and E is the set of radio links or logical transport connections between them. A given managed entity i may represent a radio unit, a modem, an RF chain, a link, a site-sector, or another operational unit selected according to the granularity required by the deployment.

During field inference, however, the deployed runtime does not assume access to the complete live graph or to a centralized observation state. Instead, it uses the locally available or cached view of the relevant managed entity and its operational neighborhood, denoted by G_i^{loc}. This local graph view may include the radio itself, its directly associated link, selected neighbouring entities if their measurements are locally available, and static or periodically updated topology/configuration metadata received through approved model or configuration updates.

$$x_{i,t}^{loc^{=}}[p_{i,t}, h_{i,t}, c_{i,t}^{loc}, a_{i,t}, e_{i,t}^{loc}, m_{i,t}]$$

In this formulation, $p_{i,t}$ denotes the performance-measurement vector of entity i at time t, including measurements such as received signal level, modulation state, capacity, throughput, error counters, retransmissions, availability indicators, latency-related counters, utilization, and other radio or transport KPIs available through the management point. The vector $h_{i,t}$ denotes internal device-health information, including temperature indicators, power-related measurements, fan or cooling behaviour when available, hardware alarms, reset patterns, firmware or software events, board or card status, RF-chain indicators, and other signs of internal degradation. The vector $c_{i,t}^{loc}$ denotes local or cached configuration and topology context, such as operating profile, channel configuration, link role, protected path information, locally known neighbour relation, and configuration age. The vector $a_{i,t}$ denotes alarms and events generated by the device or local management function. The vector $e_{i,t}^{loc}$ denotes environmental context available without assuming live Internet connectivity, including embedded regional weather priors, cached weather observations or forecasts, local environmental sensors where available, seasonal indicators, and explicit freshness or missingness flags. The vector $m_{i,t}$ represents metadata required for robust offline inference, including model version, feature availability indicators, confidence flags, and local data-quality indicators.

The model does not operate on $x_{i,t}$ alone. . It operates on an ordered history of locally available observations over a look-back interval of length L . This time-window representation allows the model to distinguish between instantaneous noise and meaningful temporal behaviour while preserving the requirement that inference can continue without centralized observation. The input sequence for entity i at time t is:

$$Xi, t = \left[x_{i,t-L+1}^{loc}, x_{i,t-L+2}^{loc}, \ldots, x_{i,t}^{loc}\right]$$

The prediction target is multi-task in nature. The model estimates the probability of each failure reason, the expected time to failure, the expected time to recovery, and the probability that human or automated intervention is required. Let R denote the set of supported failure reasons, such as weather-induced attenuation, wind or alignment-related degradation, internal RF-chain degradation, hardware or board instability, power or cooling problems, software or firmware instability, cabling or connector degradation, topology-induced overload, traffic-induced stress, and unknown or mixed causes. Let L_{loc} denote the set of possible localization targets, such as device, link, site, RF chain, modem, interface, or network segment. The inference function is expressed as:

$$F_\theta(X_{i,t}, G) \rightarrow P(r|X_{i,t}, G), P(l|X_{i,t}, G), TTF_{i,t}, TTR_{i,t}, P(I = 1|X_{i,t}, G) F\_\{\theta\}(X_{i,t}, G)$$
$$\rightarrow P(r|X_{i,t}, G), P(l|X_{i,t}, G), TTF_{i,t}, TTR_{i,t}, P(I = 1|X_{i,t}, G)$$

Here, θ represents the model parameters, r is a failure reason in R , l is a localization target in L_{loc}, $TTF_{i,t}$ is the estimated time to failure, $TTR_{i,t}$ is the estimated time to recovery, and I is a binary or graded indication of whether intervention is required. The local graph view G_i^{loc} is included because a radio entity may be affected by directly associated link conditions, cached neighbour context, known shared-site dependencies, local traffic rerouting, common power conditions, or locally visible cascading effects. The algorithm is therefore allowed to use topology-aware context where it is available, but it must degrade gracefully to per-device or per-link inference when neighbouring measurements or centralized state are unavailable. For example, if several locally visible adjacent links show correlated attenuation during a rain event, the model should reason differently than when one link alone shows progressive degradation while neighbouring links remain stable. If only the target device time series is available, the same model package can still operate using missingness indicators and local uncertainty estimation.

One class of time-series encoders used by the pipeline is based on recurrent neural networks, especially Long Short-Term Memory models. For a sequence $X_{i,t}$, the LSTM processes each sample in temporal order and maintains an internal memory state that can retain long-range

degradation patterns. For each time step k within the observation window, the LSTM update is:

$$f_k = \sigma\big(W_f[h_k - 1, x_k] + b_f\big)$$

$$i_k = \sigma(W_i[h_k - 1, x_k] + b_i)$$

$$o_k = \sigma(W_o[h_k - 1, x_k] + b_o)$$

$$c_{\sim k} = \tanh(W_c[h_k - 1, x_k] + b_c)$$

$$c_k = f_k * c_k - 1 + i_k * c_{\sim k}$$

$$h_k = o_k * \tanh(c_k)$$

The hidden representation h_k summarizes the temporal evolution of the measurements up to time k . In practice, the final hidden state or an attention-weighted aggregation of hidden states is used as the embedding $z_{i,t}$ for the current prediction window. This embedding captures patterns such as slowly decreasing received level, increasing error counters, periodic instability, correlated degradation under weather events, sudden shifts after a configuration change, or recurring hardware alarms. The same framework can be implemented using GRU encoders when a lower-complexity recurrent model is required or using Temporal Convolutional Networks when efficient causal inference is preferred. A causal temporal convolution can be represented as:

$$z_{i,t} = TCN\ \phi\big(X_{i,t}\big)$$

In this case, the convolutional filters learn local and dilated temporal patterns over the observation window while preserving causal ordering. Transformer or attention-based encoders may also be used where interpretability and long-range dependency modelling are important. The attention mechanism computes relationships between time samples and feature dimensions, allowing the model to emphasize the observations that are most relevant to the predicted failure mechanism:

$$Q = X^{loc}W_Q, K = X^{loc}W_K, V = X^{loc}W_V$$

$$\text{Attention}\big(X^{loc}\big) = \text{softmax}\left(QK^{T\sqrt{d}}\right)V$$

The sequence encoder is complemented by reasoning mechanisms that improve the diagnostic value of the prediction. Anomaly reasoning compares the current window to the expected normal behaviour of the entity, region, and configuration. A reconstruction-based anomaly score can be represented as:

$$A_{i,t} = \left|\left| X_{i,t}^{loc} - \widehat{X_{i,t}^{loc}} \right|\right|_2^2$$

A high anomaly score indicates that the current behaviour deviates from the learned normal operating manifold. Change-point reasoning detects abrupt shifts in the statistical properties of the time series, which may indicate a new failure process, a configuration change, a traffic transition, or the onset of hardware instability. Survival reasoning estimates the probability that the entity will continue operating without failure for a future horizon. If $\lambda_{i,t}(\tau)$ is the estimated hazard of failure at future offset tau, the survival probability over horizon H can be expressed as:

$$S_{i,t}(H) = \exp\left(-\sum_{\tau=1}^{H} \lambda_{i,t}(\tau)\right)$$

$$P\left(TTF_{i,t} \leq H\right) = 1 - S_{i,t}(H)$$

The estimated time to failure can be derived from the predicted survival curve or from a dedicated regression head. The estimated time to recovery is learned from historical recovery outcomes, maintenance records, alarm-clear times, weather-clearance behaviour, and service-restoration observations. Weather-related failures may have recovery patterns linked to the expected duration of the environmental condition, while internal hardware problems may have recovery patterns linked to replacement, reset, reconfiguration, or technician intervention. This distinction is one of the reasons the model must combine time-series encoders with cause reasoning rather than relying on a single failure-probability score.

Graph-aware reasoning is used to account for spatial and topological dependencies between managed entities when such information is available in the local or cached operational view. In the training environment, the full graph G G may be used to learn these dependencies. In deployed offline inference, the model uses G_i^{loc}, which is the locally available or cached neighbourhood of entity i. Each entity embedding $z_{i,t}$ can bei updated using messages from neighbouring entities that are included in that local view. A generic message-passing step is:

$$m_{i,t}^{loc} = \sum_{j \in N_i^{loc}} \alpha_{i,j,t}\, \phi\left(z_{j,t}, e_{i,j,t}^{loc}\right)$$

$$g_{i,t} = \psi\left(z_{i,t},\, m_{i,t}^{loc}\right)$$

In this formulation, N_i^{loc} is the set of neighbouring entities visible or represented in the local graph view of i , $e_{i,j,t}^{loc}$ denotes locally available or cached relationship features between entities

i and j, $\alpha_{i,j,t}$is a learned or rule-constrained importance weight, and is the graph-aware representation used for final prediction. This mechanism allows the model to identify whether degradation is isolated to a single radio entity, shared across locally visible links in the same weather region, correlated with rerouted traffic, or propagated through a topology change. The mechanism is not a requirement for continuous centralized observation. Where neighbour measurements are not locally available, their corresponding inputs are masked and the inference function relies on the device-local time-series evidence and cached context.

$$P\left(r\middle|X_{i,t}^{loc}, G_i^{loc}\right) = \text{softmax}\left(W_r g_{i,t} + b_r\right)$$

$$P\left(l\middle|X_{i,t}^{loc}, G_i^{loc}\right) = \text{softmax}\left(W_l g_{i,t} + b_l\right)$$

$$TTF_{i.t} = f_{fail}\left(g_{i,t}\right)$$

$$TTR_{i.t} = f_{rec}\left(g_{i,t}\right)$$

$$P(I = 1|X_{i,t}^{loc}, G_i^{loc}) = "\{sigmoid\}(W_l g_{i,t} + b\_I)$$

Training is performed centrally or at the customer premises using historical and asynchronously collected data. The preparation process aligns all measurements to a common time base, handles missing samples, filters corrupted records, normalizes features by device type and region, constructs sliding windows, and associates windows with failure labels, alarm outcomes, trouble tickets, recovery observations, and maintenance actions. This training environment may use broader observations than those available to a disconnected device, including fleet-level PM histories, network management records, external weather data, topology snapshots, and maintenance outcomes. The resulting model is then constrained and packaged so that the deployed inference runtime uses only the feature set that is expected to be available locally, together with explicit missing-data and data-quality indicators. The generalized model is trained over all available regions and deployment conditions. Its objective combines failure-reason classification, localization, time-to-failure estimation, time-to-recovery estimation, intervention prediction, anomaly modelling, and regularization:

$$\theta_0 = \text{argmin}\,\theta \sum_{rgn \in RGN} \sum_{(X,y)\in D_{rng}} L_{total}\left(F_\theta(X,G), y\right) L_{total}$$
$$= \lambda_1 L_{reason} + \lambda_2 L_{location} + \lambda_3 L_{TTF} + \lambda_4 L_{TTR} + \lambda_5 L_{intervention} + \lambda_6 L_{anomaly}$$
$$+ \lambda_7 \left|\left|\theta\right|\right|_2^2$$

After the generalized training stage, the model is adapted to a target region using local data. The regional model θ_r is obtained by fine-tuning or calibrating the generalized model while constraining it not to overfit the smaller regional dataset. This may be represented as:

$$\theta_r = \text{argmin}_\theta \left[\sum_{(X,y)\in D_r} L_{total}(F_\theta(X, G_r), y) + \beta\left|\left|\theta - \theta_0\right|\right|_2^2\right]$$

The regularization term keeps the regionalized model close to the generalized model unless local evidence justifies specialization. This is important in operational networks because some failure classes may be rare in a specific region, while the fleet-level model may contain valuable prior knowledge about those events. Regionalization can also include calibration of decision thresholds, adaptation of weather priors, normalization statistics, local topology constraints, and region-specific uncertainty handling.

During field operation, the deployed inference runtime continuously forms the latest local observation window $X_{i,t}^{loc}$ and applies the regional model θ_r using the last approved model package. The operational output at time t is:

$$\widehat{y_{i,t}} = r_{i,t}^*, l_{i,t}^*, TTF_{i,t}, TTR_{i,t}, I_{i,t}, U_{i,t}$$

The output contains the most likely failure reason r^*, the most likely localization target l^*, the estimated time to failure, the estimated time to recovery, the intervention indication, and an uncertainty score $U_{i,t}$. The uncertainty score is required because field data may be incomplete, especially during disconnected operation or when external weather feeds are unavailable. The intervention decision can be derived through a cost-aware policy that compares the expected operational cost of available actions:

$$u_{i,t}^* = \text{argmin}_{u\in U} E\left[C(u, r, l, TTF, TTR)\middle|X_{i,t}^{loc}, G_i^{loc}\right]$$

The action set U may include monitoring only, raising an early warning, triggering a preventive maintenance recommendation, initiating remote diagnostics, recommending configuration adjustment, escalating to a technician, or requesting spare-part preparation. The model itself does not have to directly actuate network changes. Depending on the integration policy, it may provide recommendations to the management system, to an operations dashboard, to a digital-twin validation environment, or to a higher-level intent or decision engine. This design keeps the algorithm compatible with controlled operational deployment while still enabling automation where permitted.

The deployed device or local controller is not assumed to be connected to the Internet, to the cellular network, to a centralized observation service, or to the training pipeline during inference. It performs deterministic local inference using the last approved model package and the measurements available at the deployment point. If the network segment becomes isolated from higher-layer services, the inference process continues locally using the available PM stream, alarms, device-health measurements, stored topology and configuration state, cached or embedded environmental context, and missingness indicators for unavailable inputs.

Predictions generated during disconnected operation can be stored locally according to the available storage policy. Once management connectivity is restored, accumulated PMs, alarms, predictions, and outcomes can be uploaded to the network management entity and later used by the central or customer-premises pipeline for retraining, recalibration, regionalization, validation, and release of a future model update. This separation between local inference and asynchronous model lifecycle management is a central design principle of the solution.

The expected functionality of the application is therefore broader than binary failure prediction. It provides early warning of upcoming degradation or outage, classification of the most probable failure reason, localization of the affected device, link, interface, RF chain, or network segment, estimation of time to failure, estimation of time to recovery, assessment of whether intervention is required, and support for model updates as operating conditions evolve. By combining generalized fleet-level learning, regionalized adaptation, local time-series inference, graph-aware context, and centralized retraining, the solution enables predictive resilience for Ceragon transport radios in 6G-oriented deployments.

3.3.3.2.2 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

KPI #17. Reduce CAPEX and OPEX: The solution reduces OPEX by replacing predominantly reactive troubleshooting with condition-based and predictive maintenance. Early identification of likely degradation, its probable root cause and affected location reduces manual alarm correlation, unnecessary site visits and prolonged engineering investigation. Time-to-failure and intervention predictions allow maintenance teams to prioritize only assets with a credible and time-critical risk, while preparing the appropriate technician skills and spare parts in advance. Root-cause differentiation also reduces unnecessary replacement of radios, RF components or other equipment when the actual source is weather, traffic stress, configuration, or a transient condition. Local inference can continue during management disconnection, limiting dependence on continuous cloud connectivity and supporting efficient distributed operation. Over time, the accumulated prediction and outcome history enables more accurate spare-part planning, lifecycle management and targeted infrastructure investment.

KPI #19. Reduce time to manage network resources, management overhead and reaction time through proactive actions: The solution continuously converts heterogeneous PMs, alarms, health indicators, topology context and environmental evidence into a consolidated operational assessment, rather than requiring operators to investigate multiple independent dashboards. It identifies the likely failure reason and localization target, estimates TTF and TTR, and recommends an appropriate operational response, such as monitoring, remote diagnostics,

preventive maintenance, reconfiguration, escalation or field intervention. This shortens the path from anomaly detection to an actionable decision and reduces the management overhead associated with manual data collection, alarm correlation and fault isolation. Confidence and data-quality indicators ensure that uncertain cases are escalated appropriately instead of triggering unreliable automation. When integrated with approved network-management or intent-control workflows, the output can trigger validated proactive reconfigurations before the condition becomes service-affecting.

KPI #31. Reduce service recovery time below 180 seconds: The solution contributes to sub-180-second recovery by detecting degradation before a full outage occurs and identifying the likely fault source and affected link, device, RF chain or site. This reduces the diagnostic phase that typically delays restoration after an alarm.

For conditions that can be remotely mitigated, the prediction output can be passed to AI-assisted recovery algorithms in WP4, such as rerouting, configuration rollback, radio-parameter adaptation, controlled reset or traffic rebalancing.

Predicted TTR supports selection of the fastest feasible mitigation path and distinguishes conditions likely to self-recover, such as weather-related attenuation, from faults requiring active intervention.

Local runtime inference remains available during partial disconnection, allowing diagnosis and recovery recommendations even when central management access is degraded.

Achievement of the 180-second target should be validated end-to-end, including detection, inference, decision approval, actuation and service verification time.

KPI #37. Guarantee 99.99999% service continuity through AI-driven proactive management of cloud-native services: The solution supports service continuity by identifying emerging transport degradation before it propagates into cloud-native service disruption, loss of connectivity, packet loss, excessive latency or capacity shortage.

Its predictions can be exposed to the cloud-native service-management and orchestration layer as risk, root-cause, localization, TTF, TTR and intervention-confidence indicators.

These indicators enable proactive actions such as traffic steering, workload migration, service-instance scaling, redundancy activation, path reselection or preventive maintenance before service-level objectives are breached.

The model's uncertainty output is important for ultra-high-availability operation, since low-confidence predictions can be routed to conservative policies or human validation rather than causing unsafe automated changes.

The solution alone cannot guarantee seven-nines continuity; that requires resilient cloud-native architecture, redundancy, validated orchestration policies and closed-loop recovery mechanisms. It provides the predictive transport-intelligence layer needed to trigger those mechanisms early.

This is a two parts solution, the offline training and the online inference. Training takes place in the AI/ML domain where data in collected by a unified monitoring system, transformed and prepared for training. The model is trained in the ML subsystem using a pipeline that repeats the training when the model diverges or in need of updating. The trained model then is distributed to the devices and the network operations to be used for inference. The inference requires the latest data samples as input, using the model to infare, and the output is used to make a decision on the actuations, if any. This is framed using an AI-native agent which comprises of the four tuples interconnected as a closed loop. Monitoring retrieves the input data, the analytics engine is the model forward pass, the decision engine uses the inferred output data to do rule-based decision making, the actuation sends the translation of the decision into actionable instructions (e.g., call for maintenance).

3.3.3.3 Multivariate probabilistic forecasting for resource allocation

Effective resource allocation in telecommunication networks can benefit from accurate demand forecasting to optimize performance and prevent inefficiencies such as resource over-provisioning or shortages. Traditional forecasting approaches often fail to capture uncertainty and multivariate interdependencies. This frequently results in suboptimal decision-making, particularly in dynamic, multi-tenant environments such as O-RAN. To mitigate this issue, multivariate probabilistic forecasting provides a more robust approach by capturing the interdependencies among multiple resource time series demands, including but not limited to PRB, and by providing confidence intervals for predictions, enabling more adaptive and reliable resource management in O-RAN. A novel cloud-native resource allocation framework for O-RAN is proposed, integrating advanced probabilistic forecasting models within RICs. We implemented and evaluated Gaussian Process Vector Autoregression (GPVAR) and Temporal Fusion Transformer (TFT), comparing their performances against multivariate LSTM networks. The proposed solutions has been analyzed, containerized and integrated with Swagger's REST API to facilitate network exposure and seamless deployment within the O-RAN framework.

To overcome these limitations, this work adopts a multivariate probabilistic forecasting framework. Unlike traditional single-point estimators, probabilistic forecasting aims to estimate the full probability distribution of future resource demands rather than only their expected values. This paradigm shift enables the model to quantify uncertainty through prediction intervals and confidence ranges, providing a more reliable representation of future network behavior. Additionally, multivariate probabilistic models jointly process multiple correlated time series and contextual features, such as time-of-day patterns, and weekly traffic dynamics. By learning both temporal dependencies and inter-series relationships, these models such can better characterize complex network conditions and improve forecasting robustness. As a result, probabilistic forecasting supports more adaptive and risk-aware resource management strategies. These models estimate “what is the probability distribution of future demand?”, enabling operators and decision-makers to make allocation decisions based not only on expected utilization but also on the associated uncertainty and risk levels.

The primary contributions of this study are as follows:

- State-of-the-art multivariate probabilistic forecasting models (such as GPVAR and TFT) have been considered for to forecast resource requirements for Non-Real-time RIC (rApp).
- The AI solution has been containerized and integrated with Swagger’s RESTful APIs for effective network exposure and seamless interaction.
- The performance of probabilistic estimators is compared with multivariate LSTM predictors through quantitative traditional error metrics (RMSE, MAE, and MAPE) and computation complexities (training time, prediction time, CPU usage, and memory usage).
- The TFT multivariate probabilistic estimator is evaluated to illustrate its ability to model long-term dependencies and handle complex input time series data.
- High-dimensional GPVAR multivariate forecasting is analyzed more deeply through a low-rank Gaussian copula process to interpret and understand the scalability, robustness, and impact of different ranks on the performance of GPVAR. GPVAR performance is quantified through traditional error metrics, training time, and advanced performance metrics like train NLL (Non-Negative Log-Likelihood), test NLL, normalized deviation, and CRPS (Continuous Rank Probability Score).

Traditional forecasting approaches, such as multivariate LSTM estimators, typically generate deterministic single-point predictions of future resource demands. Although effective at capturing temporal dependencies, these methods do not explicitly model forecasting uncertainty, which may lead to overconfident predictions and inefficient resource allocation decisions in dynamic wireless network environments.

Probabilistic forecasting overcomes this limitation by estimating predictive probability distributions instead of single future values. This allows the model to quantify uncertainty and provide confidence information alongside the forecast, enabling more robust and adaptive decision-making. Among recent Multivariate probabilistic forecasting methods, the GPVAR model [79] captures dependencies among multiple correlated time series through probabilistic autoregressive modeling. Similarly, the TFT architecture [80] combines recurrent processing and attention mechanisms to learn temporal patterns and generate interpretable probabilistic forecasts. Both models are well suited for high-dimensional multivariate forecasting tasks due to their ability to jointly model temporal correlations and uncertainty. In the multivariate PRB allocation forecasting problem, the system state at each time instant is represented by a vector containing multiple correlated input variables. The objective is to model the conditional probability distribution of future observations based on historical data, enabling the prediction of future resource demands together with their associated uncertainty estimates.

GAUSSIAN PROCESS VECTOR AUTO REGRESSIVE (GPVAR)

Resource allocation in O-RAN can be formulated as a Multivariate time series forecasting problem. The time series are represented as $y_{k,t}$, with $k \in \{1,2,\dots,K\}$ indexing each time series (i.e., the resource demands for k-th tenant/slice) component at time t. The observation vector $\mathbf{y}_t \in \mathbb{R}^{K\text{x}1}$ uses history PRB data $\mathbf{y}_1, \dots, \mathbf{y}_{t_0-1}$and predicts future PRB demands for the multi-step ahead from t_0 up to T for estimating the joint conditional distribution $\{\mathbf{y}_{t_0}, \cdots, \mathbf{y}_{t_0+\text{T}}\}$. GPVAR focuses on optimizing multivariate normal distribution losses and provides confidence intervals for the resource demands on the multi-step ahead time horizon prediction.

Forecasting large collections of correlated time series resource demands or network KPIs is a critical task in O-RAN. Deep GPVAR is a novel autoregressive deep learning model designed to address this challenge by jointly modeling a large number of time series using low-rank Gaussian Processes. Unlike traditional global forecasting approaches that treat time series independently, Deep GPVAR explicitly captures inter-series dependencies, leading to improved accuracy and generalization. The algorithm assumes a multivariate Normal distribution and is able to learn potential correlation between the multiple input time series. Next, the key advantages of Deep GPVAR are presented: i) Covariate Integration: Ability to incorporate multiple time series alongside external covariates. ii) High Scalability: High efficiency through low-rank approximations of large covariance matrices. iii) Multi-dimensional Dependency Modelling: In contrast to single variate probabilistic forecasting models like DeepAR, which assume conditional independence among series, Deep GPVAR leverages a low-rank estimation of the covariance structure and employs Gaussian Copulas for adaptive

transformation and scaling. This architecture allows Deep GPVAR to learn richer temporal dynamics and shared patterns across series, establishing it as a powerful tool for high-dimensional time-series forecasting.

Low-Rank Gaussian Approximation

Estimating the covariance matrix is one of the most effective ways to capture relationships among multiple time series. However, scaling this process to a high number of series poses significant challenges due to memory constraints and numerical instability. Moreover, covariance estimation is computationally intensive, as it must be performed every time window during training. To overcome these limitations, the authors of GPVVAR propose simplifying the estimation through a low-rank approximation of the covariance matrix.

Recall the basic structure of a multivariate normal distribution $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} \in \mathbb{R}^{K\times 1}$ is the mean vector, $\boldsymbol{\Sigma} \in \mathbb{R}^{K\times K}$ is the covariance matrix, Kdenotes the total number of time series. The probability density function is defined as:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{K/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

A key challenge lies in the size of the covariance matrix $\boldsymbol{\Sigma}$, which grows quadratically with the number of time series K in the dataset. This scaling makes direct computation impractical for large K. To address this issue, the covariance structure can be approximated using a low-rank Gaussian approximation. This approach, rooted in factor analysis and conceptually related to Singular Value Decomposition (SVD), significantly reduces computational complexity. Rather than computing the full $K \times K$ covariance matrix, we approximate it by estimating a lower-dimensional representation consisting of a Diagonal matrix plus a low-rank matrix:

$$\boldsymbol{\Sigma} \approx \mathbf{W}\mathbf{W}^{\top} + \mathbf{D},$$

where $\mathbf{W} \in \mathbb{R}^{K\times r}$ captures the low-rank structure (with $r \ll K$), $\mathbf{D} \in \mathbb{R}^{K\times K}$is a diagonal matrix accounting for residual variances. Since $r \ll K$, it has been proven that the Gaussian likelihood can be computed using $\mathcal{O}(Kr^2 + r^3)$ operations, instead of $\mathcal{O}(K^3)$ [79].

GPVAR Architecture overview

The Deep GPVAR employs a Long Short-Term Memory (LSTMs) network and autoregressive (AR) architectures. Suppose K input time series, corresponding to the PRB demands of different tenants, indexed by $i = 1, \ldots, K$. At each time step t, the model follows this process:

1. Input: An LSTM cell receives the target variable $z_{t-1}$from the previous time step for a subset of time series, along with the hidden state $\mathbf{h}_{t-1}$ from the previous step.

2. Processing: The LSTM processes this information and produces an updated hidden state $\mathbf{h}_t$.

3. Parameterization: The hidden vector $\mathbf{h}_t$ is then used to parameterize a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. This distribution is a specific case known as a Gaussian copula. Mathematical algorithmic details are omitted herein but, Gaussian Copulas is a type of function that allows modelling the dependencies of multiple variables, even when these variables have different or non-Gaussian marginal distribution.

with transition dynamics φ parameterized by LSTM. Parameters $\theta_h$are shared for all time series, but the LSTM is applied separately to each time series. With state values $\mathbf{h}_{k,t}$known for all time series $k = 1,2,\ldots,K$, and representing the gathered state values at time $t$as $\mathbf{h}_t$, the joint distribution is parameterized using a Gaussian copula.

Time Fusion Transformers (TFTs)

TFT [80] is a specialized, attention-based deep learning architecture designed for multi-horizon time-series forecasting. It excels at capturing complex temporal dependencies across high-dimensional, multivariate datasets while simultaneously providing robust uncertainty quantification (i.e., across the multiple input data streams corresponding to the time-varying resource demands, e.g., PRBs, of the different tenants/slices).

The operational efficacy of the TFT relies on five primary architectural mechanisms:

- Gating Mechanisms: These networks dynamically adjust computational complexity by skipping unused or redundant components based on the dataset. This adapts the model's depth, optimizes resource management and prevents overfitting.
- Variable Selection Networks: To handle high-dimensional data, these networks dynamically isolate and select the most relevant features at each time step, filtering out noisy or irrelevant inputs.
- Static Covariate Encoders: These encoders transform time-independent variables into context vectors, conditioning the temporal dynamics of the network with stable baseline context.
- Hybrid Temporal Processing: Integrates recurrent sequence-to-sequence structures to capture localized, short-term temporal dynamics alongside interpretable multi-head attention to map long-range historical dependencies.

- Quantile Forecast Output: Instead of producing a single point forecast, the model outputs prediction intervals in the form of multiple quantiles, enabling explicit confidence and uncertainty quantification for each step in the prediction horizon.

We evaluated the performance of TFT for forecasting future PRB demand in a high-dimensional multivariate setting. We investigated the ability of TFTs to capture complex temporal dependencies and quantify prediction uncertainty. The probabilistic forecasting objective is modeled by estimating the joint conditional distribution of the future targets. For a given quantile q at a τ-step-ahead horizon.

Loss function: The model is optimized using the average quantile loss function. The quantile loss function for quantile q, which is denoted by L_q, is defined as follows:

$$L_q\left(y_{k,t}, \hat{y}_{k,t}^q\right) = max\left(q(y_{k,t} - \hat{y}_{k,t}^q), (1-q)(y_{k,t} - \hat{y}_{k,t}^q)\right),$$

where $y_{k,t}$ denotes the actual value of the resource demand for k-th tenant/slice and $\hat{y}_{k,t}^q$ represents the is the predicted quantile value for k-th data stream at time t.

TFT is trained by jointly minimizing the average quantile loss, summed across all the quantile outputs and all the input time series, corresponding to O-RAN KPIs, i.e., PRB demands for the different tenants/slices.

$$L = \frac{1}{|Q|}\sum_{k=1}^{K}\sum_{q=1}^{Q}\sum_{t=1}^{T} L_q\left(y_{k,t}, \hat{y}_{k,t}^q\right)$$

3.3.3.3.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

KPI #17. Reduce CAPEX and OPEX: Probabilistic forecasting of PRB utilization using GPVAR and TFT enables mobile network operators to predict not only expected traffic demand but also the likelihood of different demand scenarios. GPVAR & TFT capture temporal relationships and traffic dependencies between the resource demands of multiple tenants/cells/slices. By producing uncertainty-aware forecast distributions (e.g., P50, P90, P95) instead of a single forecast value, these models provide a more realistic view of future network load and congestion risk. This uncertainty-aware forecasting helps reduce CAPEX by identifying sites where sustained high PRB utilization is highly probable, allowing proactive resource increase (i.e., capacity expansions, carrier additions, or site upgrades) to be prioritized only where they are truly needed. At the same time, it reduces OPEX by enabling proactive network optimization, dynamic resource allocation, energy-saving actions during low-demand periods, and earlier mitigation of potential congestion issues before they impact service quality. The

result is more efficient use of existing network assets, improved customer experience and lower overall operating and investment costs.

The proposed solution is a rApp that lies in the AE (Analytic Engine) within the Non-real time RIC. Monitoring System (MS) and is in charge of getting the corresponding metrics of E2 nodes via E2 interfaces and the corresponding probabilitistic forecasts are then send to a Decision Engine (DE) who finally actuates over the PRB resources.

3.3.3.3.2 Preliminary results

A PyTorch forecasting setup has been considering using a 1-month hourly dataset across PRB demand time series representing different tenants/slices, evaluating a 72h context length with a 48h prediction window. For the purpose of the analysis, a scenario with network coverage by several base stations is considered. Based on the available dataset, K = 29 input data streams representing the PRB demand of K tenants/slices are provided as input for the multivariate forecast estimators. Table 4 summarizes the configuration parameters of the considered algorithms: multivariate LSTM, GPVAR and TFT.

Table 4: Model configuration parameters.

Parameter	LSTM	GPVAR	TFT
Batch Size	128	128	128
Max Epochs	150	150	150
Layers	3 Dense Layers	2 RNN Layers	Attention Head, Hidden Continuous Size: 1,8
Optimizer	Adam	Adam	Adam
Loss Function	Mean Square Error (MSE)	Multivariate Normal Distribution Loss (Rank-28)	Quantile Loss
Dropout Rate	0.1	-	0.1
Gradient Clipping	-	0.1	0.1
Regularization	L2 Regularization	-	-
Context Length	72	72	72
Time Varying Known Reals	-	Hours, Weekdays, Weekends	Hours, Weekdays, Weekends
Hyperparameter Tuning	Optuna for selecting optimal learning rate and hidden size		

Multivariate probabilistic forecasting vs univariate probabilistic forecasting

To show the limitations of using univariate probabilistic estimators like DeepAR as a baseline, their performance is compared with that of the multivariate GPVAR forecasting model. DeepAR treats each tenant's PRB time series independently and requires a separate model for each series for both training and inference. GPVAR, on the other hand, being a multivariate probabilistic model, processes all tenant's information jointly and explicitly models their dependencies and relations, which is crucial for resource optimization in a multivariate O-RAN environment. As we consider K = 29 tenants in this work, running 29 DeepAR models in parallel to process 29 tenants' PRB data produces significant computational overhead and does not provide additional methodological benefits due to its inability to exploit cross-tenant correlation. Therefore, for a fair comparison between the univariate and multivariate probabilistic

estimators we considered PRB demands of K = 9 tenants (i.e., 9 time series data streams). Table 5 shows the performance comparison of DeepAR and GPVAR on 9-Tenant's PRB demand with regards to multiple evaluation metrics. Compared with GPVAR, DeepAR shows lower prediction accuracy, with RMSE, MAE, and MAPE of 39.70, 38.91, and 15.23, respectively. For those metrics, such as they are single point metrics the median both probabilistic techniques are considered (50% percentile). Additionally, it indicated a weaker calibration and less reliable uncertainty quantification with a high normalized deviation of 19.01 and a CRPS of 0.505, unlike GPVAR, which showed superior performance with normalized deviation and CRPS of 16.31 and 0.402, respectively. These results clearly show that multivariate frameworks such as GPVAR (and TFT in our broader evaluation) are more reliable and robust for predicting future resource demands while simultaneously capturing uncertainty and cross-dependencies with high accuracy.

Table 5. Performance conmparison of GPVAR vs DeepAR

Evaluation Metrics	**DeepAR**	**GPVAR**
RMSE	39.70	24.11
MAE	38.91	11.83
MAPE	15.23	8.06
Normalized Deviation	19.01	16.31
CRPS	0.505	0.402

Performance and computational complexity analysis

The CPU and memory usage recorded and analyzed in this section highlight the processing overhead during training and inference, independent of GPU acceleration. The estimator's computational footprint on the system is emphasized here, along with the relative efficiency of all models under practical conditions. The consideration of CPU and memory utilization by estimators, together with their accurate prediction capability, is essential for real-world deployment. Figure 34 and Figure 35 help us to better understand the complexity of the model. In all figures, the x-axis shows the time for training/prediction in seconds, while the y-axis shows the CPU utilization per percentage. The memory usage percentage of all models during training/prediction is shown in circle color in the form of a heat map, and the CPU usage percentage is depicted by the varying size of the circle. Initially, Figure 34 explains the CPU

and memory usage during the training phase of the model over time. The time during training includes the time required for preprocessing the data, splitting, training the model, and storage. CPU utilization fluctuates throughout the runtime due to the complex structures and resource requirements of all probabilistic estimators. The maximum CPU utilization for both GPVAR and TFT is between 46.8% and 48.9%, while LSTM requires very little Central Processing Unit (CPU) at around 10.5%. Despite the sequential processing of data, LSTM is also the fastest in terms of training time due to its simple architecture and fewer calculations.

A similar performance is evident in Figure 34 in relation to memory utilization. GPVAR requires the highest amount of memory around 97.5% during training due to the modeling of the multivariate distribution and temporal dependencies. TFT, on the other hand, exhibits stable behavior with moderate memory requirements but high training time due to the attention mechanism, variable selection network to select relevant inputs at each step, and temporal fusion. The Multivariate LSTM memory requirement is the lowest, and the training time is very low due to its lightweight nature, smaller number of parameters, and simpler architecture, which illustrates the trade-off between computational complexity and network efficiency.

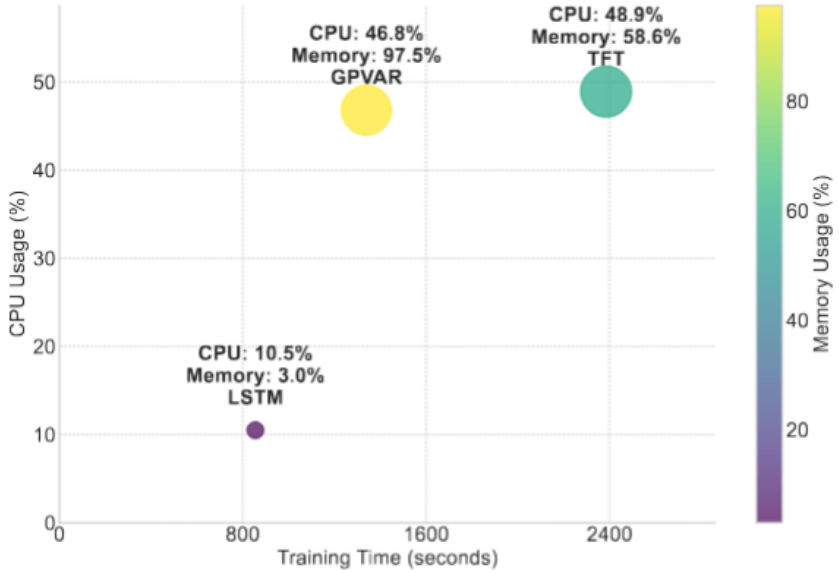


Figure 34. CPU and memory usage during training

Figure 35 shows the CPU and memory utilization during PRB prediction. The prediction time is the sum of the loading time of the data and the trained model, the prediction time and the post-processing time. TFT requires the lowest CPU of 10% for the prediction. GPVAR and LSTM require almost the same amount of CPU, about 37% to 40%, due to the RNNs involved, but the time in which the CPU is occupied is comparatively high for LSTM due to the

serializationof data processing. This metric demonstrates the model's effectiveness in processing new data and making accurate predictions.

Figure 35 also shows memory usage during the tests for all Multivariate LSTM, GPVAR and TFT estimators. TFT and GPVAR require a moderate amount of memory, about 40% to 42%, for a lower duration than LSTM. The memory requirement depends on the complexity of the estimator's architecture. LSTM requires a considerable amount of memory for sequential processing, additional memory for storage due to multivariate processing, large input embeddings and the calculation of internal gates. On the contrary, a sophisticated TFT framework requires the lowest memory and makes faster predictions due to the parallel processing of all time steps by the attention mechanism.

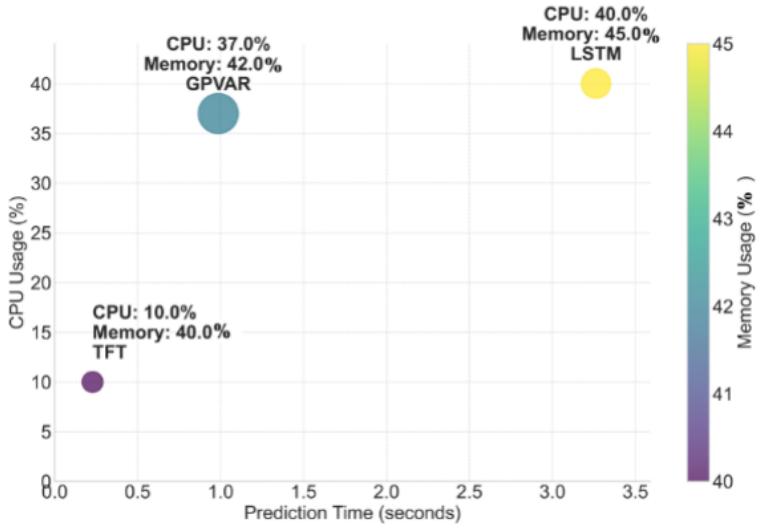


Figure 35. CPU and memory usage during inference phase

Low rank approximation analysis in GPVAR

The ranks determine the dimensions of the covariance matrix approximation of the multivariate Gaussian process in GPVAR estimators for high dimensional inputs. A low rank reduces the complexity of the model, but higher ranks can capture the temporal relationships in the data more efficiently. Analyzing and evaluating the trade-off between the model's complexity and performance scalability through different ranks enables a better understanding of how to optimize its use in real-world scenarios.

Next Table presents a quantitative analysis of the effect of rank on the performance of the GPVAR model using several evaluation metrics, including Negative Log-Likelihood (NLL), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Normalized Deviation (ND), and Continuous Ranked Probability Score (CRPS). The results indicate that the model achieves its best overall performance at Rank 28, yielding

the lowest Test NLL of 231.34, RMSE of 20.017, MAPE of 4.06%, MAE of 9.387, and ND of 15.31. In addition, Rank 28 achieves a competitive CRPS value of 0.37 and the lowest Train NLL of 466.81. In contrast, lower ranks, such as Rank 1, exhibit substantially poorer performance, with a Test NLL of 298.62, RMSE of 24.69, and MAE of 11.69. These results suggest a weaker ability to generalize to unseen data. As the rank increases, probabilistic forecasting performance improve noticeably, as evidenced by the decreasing values of Test NLL, RMSE, MAE, MAPE, ND, and CRPS. Rank 28 stands out as the optimal configuration, balancing prediction accuracy and probabilistic modelling capability. The low Train NLL achieved at this rank indicates an improved ability to capture the underlying probabilistic structure of the data. Overall, increasing the rank enables GPVAR to model more complex temporal relationships, leading to enhanced prediction accuracy and forecasting robustness. However, the improvements diminish as the rank approaches its maximum value (Rank 29). Ranks 1 and 10 consistently demonstrate the weakest performance across most evaluation metrics, highlighting their limited capability in PRB demand prediction. Although increasing the rank generally improves forecasting performance, performance begins to deteriorate when the rank becomes very close to the number of input data series. This degradation is likely due to increased model complexity and the tendency to capture excessively detailed patterns that do not generalize well. The trends observed in this Table clearly demonstrate the impact of rank selection on model accuracy and computational complexity. The rank selection process can be automated through an early-stopping strategy that progressively increases the rank while continuously monitoring model performance. Once no further improvement is observed, training can be terminated to avoid unnecessary computational overhead. This approach facilitates an effective trade-off between computational cost and prediction accuracy, making it particularly suitable for real-time O-RAN deployments. Furthermore, the reliability of the selected rank with respect to scalability, varying data distributions, and model generalization can be further validated through cross-validation using NLL and by employing statistical model-selection criteria such as the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC).

Table 6. Effect of GPVAR rank on the forecasting performance

Rank	Train NLL	Test NLL	RMSE	MAE	MAPE	Normalized Deviation	CRPS
1	488.32	298.62	24.69	11.69	4.88	18.78	0.39
2	470.21	272.102	23.50	10.42	4.67	17.76	0.38
5	469.88	275.60	24.36	11.55	4.82	18.05	0.39
10	477.47	285.97	23.53	10.87	4.59	17.44	0.38
16	472.00	252.06	21.07	10.17	4.21	16.32	0.38

20	470.47	237.66	21.36	10.02	4.11	16.07	0.38
24	473.52	235.34	21.14	9.95	4.09	15.97	0.38
26	469.30	239.73	21.93	9.89	4.07	15.88	**0.37**
28	**466.81**	**231.34**	**20.017**	**9.387**	**4.06**	**15.31**	**0.37**
29	467.27	240.61	21.55	10.26	4.28	16.47	**0.37**

Next two figures illustrate the CPU and memory usage of GPVAR during the training and prediction phases for different covariance ranks. As shown in Figure 36, during training, CPU utilization increases noticeably with higher ranks, rising from 56.4% at rank 1 to approximately 97.5–98.1% at ranks 28 and 29. This increase is expected because higher-rank tensor operations require greater computational effort during GPVAR training. Memory usage follows a similar pattern, growing from 18.7% at rank 1 to around 47% at rank 29. The higher memory demand results from the need to store larger low-rank covariance structures and additional model parameters as the rank increases. These results demonstrate that the choice of rank has a significant impact on training complexity, affecting both computational and memory requirements. Therefore, selecting an appropriate rank is essential to achieve a balance between computational efficiency and model performance.

On the contrary, predictions are less resource-intensive than training, regardless of the tensor rank, as shown in Figure 37. CPU usage remains relatively stable between 37% and 38%, while memory usage varies moderately around 42%. This behavior is expected during inference of future resource demands, since the prediction horizon is fixed at 48 hours and the covariance parameters are already computed during the training phase. Such consistent computational requirements during inference make the approach suitable for deployment on edge systems or Near-RT RIC, where it can support delay-sensitive decisions in O-RAN resource management.

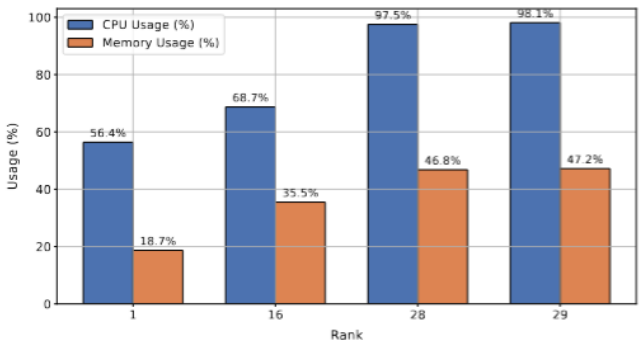


Figure 36. CPU and memory usage of GPVAR during training phase as function of the rank.

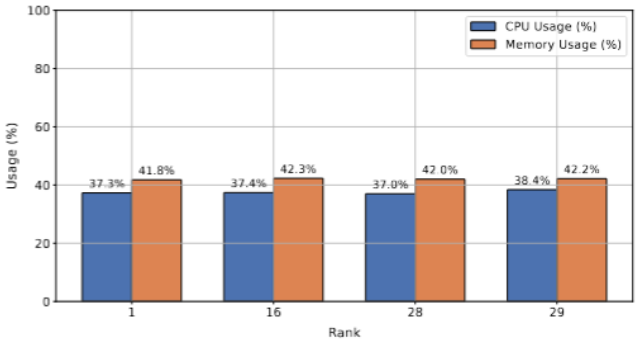


Figure 37. Rank effect analysis of CPU and memory usage in prediction (inference phase).

3.3.3.4 Container Forecasting Engine

Performance optimisation and energy efficiency are increasingly critical concerns in 6G networks. As 6G is being designed as a cloud-native system, network functions are expected to be fully containerised (CNFs), dynamically scheduled, and distributed across edge and cloud nodes. The orchestrator responsible for managing these CNFs must continuously decide how much CPU and memory to assign to each function as workloads evolve.

Proactive orchestration addresses these limitations by forecasting future workload conditions and allocating resources ahead of demand peaks. This enables smoother scaling transitions, reduces the need for reserved spare capacity, and improves overall energy efficiency by aligning resource allocation more closely with actual network demand. However, the effectiveness of such predictive orchestration strategies strongly depends on the accuracy and robustness of the underlying forecasting process. In highly dynamic 6G environments, workload patterns may show heterogeneous temporal behaviours, including non-stationarity and abrupt regime changes. As a consequence, selecting the forecasting model that best matches the current characteristics of the observed data becomes a critical challenge. This challenge is further complicated by the phenomenon of concept drift (also referred to as covariate shift), whereby the statistical properties of the time series evolve over time: a linear model may produce superior results during stable periods, while a non-linear model becomes preferable during regime changes. It's clear then how one, single, pre-selected forecasting model cannot adapt to such dynamics.

To address these challenges, the proposed approach, i.e., **Container Forecasting Engine (CFE),** proactive allocates CNF resource. Unlike traditional reactive approaches, the CFE enables the 6G infrastructure to allocate resource in advance, making them available right when they are needed, avoiding cold start delays and over-provisioning.

The CFE ingests raw time-series telemetry from an observability subsystem (typically a Prometheus-based stack in Kubernetes-native environments, tracking CPU and memory utilisation, application metrics, and energy consumption for pods, containers, running CNFs), processes it through an ensemble of specialized autonomous agents, generates resource usage forecasts, and produces a recommended scheduling action for the CNF orchestrator, which implements the actual resource adjustments on the Kubernetes control plane.

The contribution beyond the state of the art is twofold: (i) automated, adaptive model selection driven by LLM-based reasoning over statistical data profiles, eliminating the need for manual model engineering; and (ii) a structured multi-agent decomposition of the forecasting and allocation workflow, where each agent encapsulates a distinct cognitive function and agents coordinate via the A2A protocol.

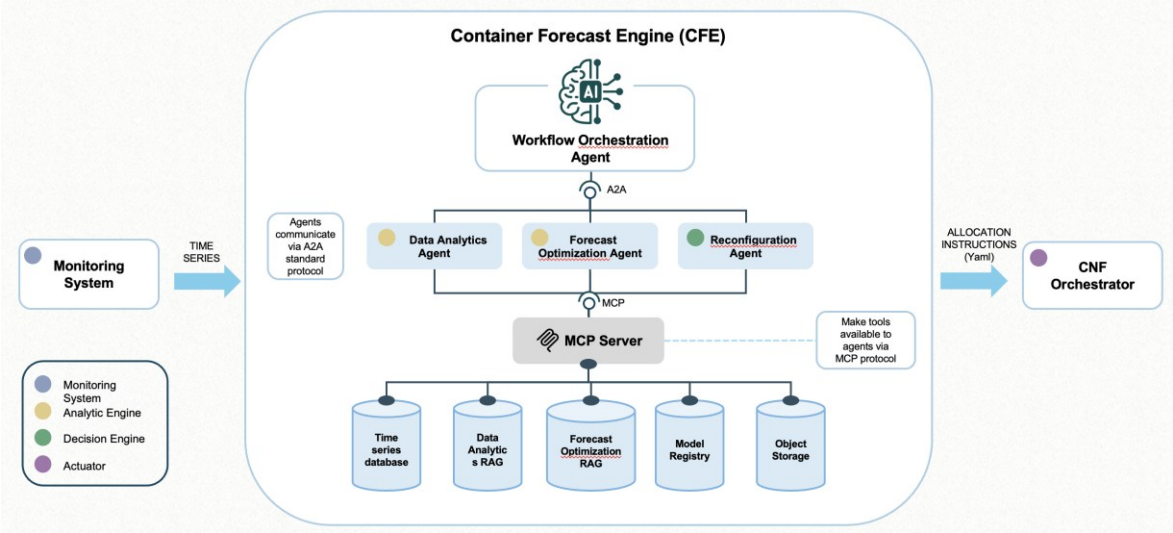


Figure 38: Architecture diagram of the Container Forecast Engine (CFE)

The CFE, shown in Figure 38, operates within the management and orchestration layer, positioned between two external components: an *Observability Subsystem* (typically Prometheus-based in Kubernetes-native environments) and a *CNF Orchestrator* (Kubernetes control plane). The framework comprises four agents:

- Data Analytics Agent: This agent ingests raw time series telemetry from the observability subsystem. Real-world telemetry is typically noisy, irregularly sampled, and may contain missing values. The Data Analytics Agent uses LLM-based reasoning in conjunction with analytical tools to preprocess and transform this raw data into a structured analytical summary. It detects anomalies such as missing values, outliers, and unexpected patterns; performs statistical analysis of the time series (trend, seasonality, autocorrelation, variance), and reasons over data quality and temporal characteristics. The output is an

analytical summary encoding both quantitative statistics and qualitative diagnostics consumed by downstream agents.

- Forecast Optimization Agent: This agent ingests the analytical summary produced by the Data Analytics Agent and is responsible for model selection and hyperparameter optimisation. Given the statistical profile of the time series, whether it is linear or non-linear, whether it exhibits strong seasonality, the agent reasons over a space of candidate forecasting model families (e.g., ARIMA, LSTM, Prophet) and ranks them based on expected suitability on current data. It then performs hyperparameter optimisation to identify the best configuration for the current data regime. This dynamic selection mechanism directly addresses concept drift: the agent continuously re-evaluates model suitability as statistical properties evolve, adapting the forecasting strategy without human intervention.
- Reconfiguration Agent: This agent takes the forecast produced by the selected model and translates it into actual K8s manifests. Rather than simply passing through forecast numbers, the Reconfiguration Agent aligns predicted demand with system-level objectives and scheduling policies. It retrieves relevant allocation templates from a knowledge base via an MCP server, structured documents encoding canonical resource allocation policies for different CNF profiles, and reasons over them to produce a context-appropriate scheduling recommendation that the CNF orchestrator can directly apply.
- Workflow Coordination Agent (orchestrator): A workflow orchestration agent oversees the execution of the entire workflow. It manages graph node sequencing, agents calls, ensures state consistency across agents, handles retries and evaluation of agents' outputs. Inter-agent communication is implemented through the A2A (Agent-to-Agent) protocol, with each agent exposing a standard interface callable by other agents and the orchestrator.

A key enabling technology across this agentic framework is the Model Context Protocol (MCP), used to interact with external tools and different data sources in a standardised, interoperable way. MCP-accessed resources include network topology information, historical records, a model registry, and allocation template libraries. In addition to MCP, Retrieval-Augmented Generation (RAG) is employed to ground agent reasoning in real-time, network-specific data. While pre-trained LLMs rely on general knowledge, RAG acts as a dynamic external knowledge base queried at inference time, injecting relevant context directly into the agent's reasoning prompt. This combination enables agents to make informed, live decisions that reflect current network state rather than static training data.

3.3.3.4.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The CFE framework will be integrated in the MS-AE-DE-ACT architecture, which enables closed-loop, AI-driven lifecycle management of CNFs.

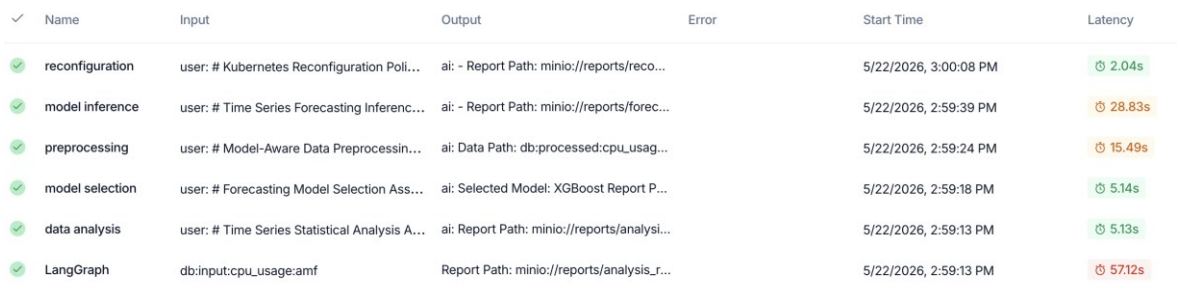
In this context, the **MS** collects fine-grained telemetry from CNFs (e.g., CPU, memory usage, container runtime metrics), which directly relates to the energy consumption components defined in KPI #15. These measurements are enriched and structured to provide the foundation for downstream analytics and forecasting. The **AE** is responsible for transforming monitored telemetry into predictive intelligence, including anomaly detection, workload forecasting, and energy consumption modelling. This layer directly supports the KPI objective by enabling the forecast of future resource demand, which is essential for reducing overprovisioning and improving energy efficiency. In particular, AE implements the **Data Analytics Agent** and **Forecast Optimization Agent** to adapt forecasting strategies to evolving network conditions, thereby improving prediction accuracy under heterogeneous workloads. This capability is essential to capture CNF behaviour across highly dynamic 6G environments. The **Reconfiguration Agent** within the **DE** operationalizes these predictions by translating them into proactive orchestration actions, aligning predicted demand with available computational and network resources. In the context of KPI #15, DE plays a central role in minimizing unnecessary CNF instantiations and reducing idle resource allocation, thus directly contributing to energy savings. Moreover, DE supports policy-driven decision-making, allowing tenant-specific constraints and energy-awareness objectives to be incorporated into the orchestration logic. Finally, the **ACT,** represented by the external **CNF Orchestrator,** execute these decisions by applying the generated reconfiguration actions, including scaling operations and resource allocation adjustments within Kubernetes-based deployments.

This architecture is validated within **PoC1**, which provides two complementary evaluation scenarios. In ES#1, the focus is on IoT energy management in an edge-cloud continuum under constrained and heterogeneous network conditions, where distributed AI and conflict resolution mechanisms are used to optimize shared resource usage across multiple tenants. This scenario directly evaluates KPI #15 by assessing how AI-driven orchestration reduces CNF energy consumption while maintaining fairness and QoS under scarcity conditions. In ES#2, a semantic-aware orchestration framework is deployed, where the AI-driven 6G orchestrator performs intelligent resource allocation across RAN, transport, and compute domains using semantic insights. Here, proactive workload prediction and AI-based allocation strategies are used to optimize resource utilization across domains, further contributing to energy efficiency by reducing unnecessary scaling and improving workload placement.

3.3.3.4.2 Preliminary results

At the current stage of development, the work has primarily focused on the design, architectural definition, and initial implementation of the CFE, a multi-agent framework for proactive orchestration of cloud-native network functions in 6G environments. This initial focus has been motivated by the highly emerging nature of multi-agent AI systems, whose enabling

frameworks, orchestration paradigms, and communication protocols have only become available very recently. Rather than concentrating on numerical forecasting benchmarks, the effort has therefore been directed toward establishing a robust and scalable agentic architecture capable of supporting production-oriented AI-driven orchestration workflows. In particular, the framework has been structured around specialized AI agents responsible for data analysis and preprocessing, forecasting model selection, inference execution, and Kubernetes reconfiguration generation, enabling a modular and extensible orchestration pipeline.



Name	Input	Output	Error	Start Time	Latency
reconfiguration	user: # Kubernetes Reconfiguration Poli...	ai: - Report Path: minio://reports/reco...		5/22/2026, 3:00:08 PM	2.04s
model inference	user: # Time Series Forecasting Inferenc...	ai: - Report Path: minio://reports/forec...		5/22/2026, 2:59:39 PM	28.83s
preprocessing	user: # Model-Aware Data Preprocessin...	ai: Data Path: db:processed:cpu_usag...		5/22/2026, 2:59:24 PM	15.49s
model selection	user: # Forecasting Model Selection Ass...	ai: Selected Model: XGBoost Report P...		5/22/2026, 2:59:18 PM	5.14s
data analysis	user: # Time Series Statistical Analysis A...	ai: Report Path: minio://reports/analysi...		5/22/2026, 2:59:13 PM	5.13s
LangGraph	db:input:cpu_usage:amf	Report Path: minio://reports/analysis_r...		5/22/2026, 2:59:13 PM	57.12s

Figure 39: LangGraph execution trace of the CFE multi-agent workflow.

Significant effort has also been dedicated to evaluating and integrating state-of-the-art open-source frameworks for agentic AI development, including LangChain [82] and LangGraph [83]. LangGraph was selected as the core orchestration framework due to its support for graph-based workflows, persistent state management, conditional execution logic, and production-oriented orchestration of multi-agent systems. In parallel, the project has adopted LLMOps principles from the early development phases, recognizing the importance of observability, traceability, and explainability in AI-driven infrastructure management systems. To this end, LangSmith [84] has been investigated and integrated to support runtime tracing, inspection of agent execution paths, debugging of LLM interactions, and monitoring of reasoning workflows, thereby improving transparency and interpretability of orchestration decisions.

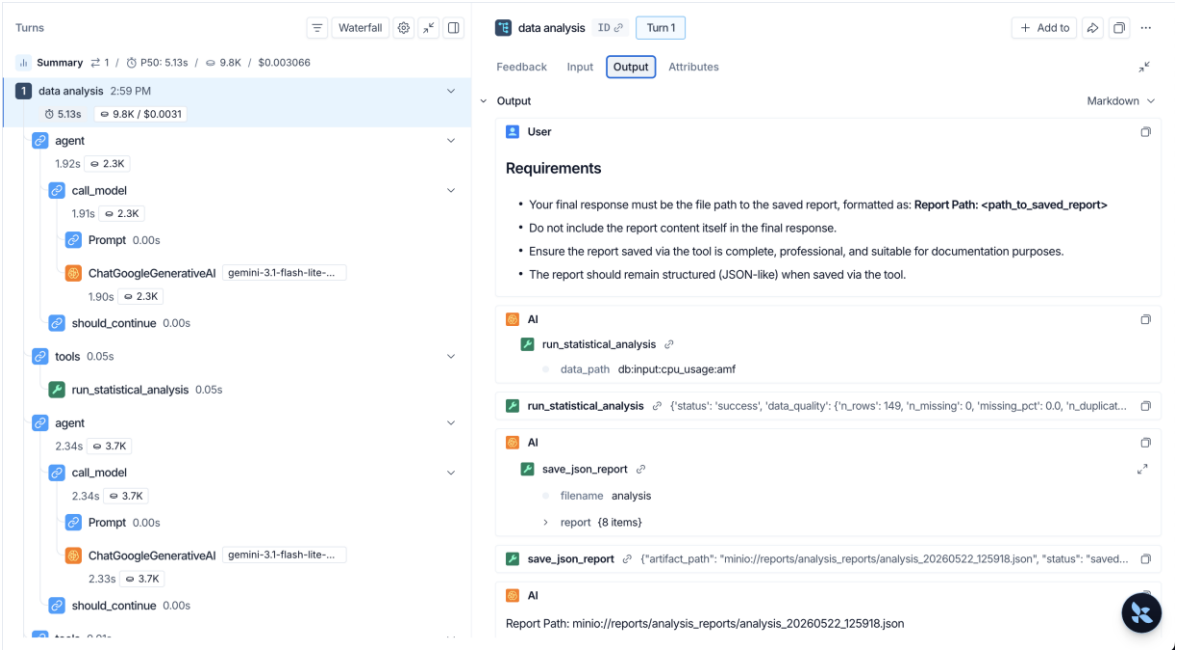
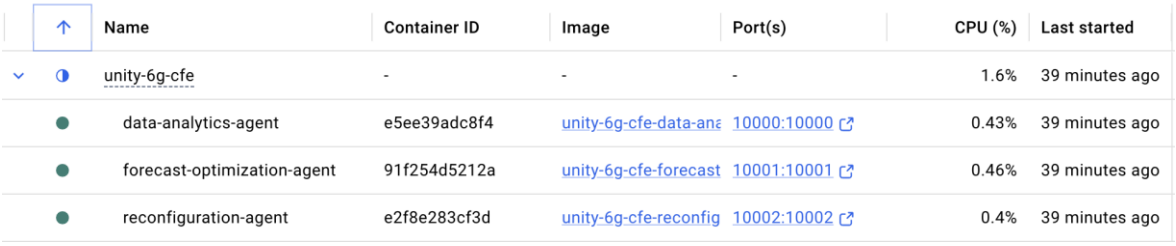


Figure 40: Example of an agent execution trace visualized through LangSmith.

Furthermore, the framework evolution has progressively moved beyond a purely local prototype toward a more scalable and production-oriented deployment model through containerized execution environments based on Docker technologies, paving the way for future integration into Kubernetes-native and distributed 6G orchestration infrastructures.



Name	Container ID	Image	Port(s)	CPU (%)	Last started
unity-6g-cfe	-	-	-	1.6%	39 minutes ago
data-analytics-agent	e5ee39adc8f4	unity-6g-cfe-data-ana	10000:10000	0.43%	39 minutes ago
forecast-optimization-agent	91f254d5212a	unity-6g-cfe-forecast	10001:10001	0.46%	39 minutes ago
reconfiguration-agent	e2f8e283cf3d	unity-6g-cfe-reconfig	10002:10002	0.4%	39 minutes ago

Figure 41: The figure shows the three specialized AI agents deployed as independent containers in Docker

To validate the initial implementation of the CFE agentic pipeline, the development has been grounded on a publicly available dataset of 5G Core network resource usage [81]. In particular, an open dataset describing AMF workload traces was adopted as a representative benchmark for control-plane signalling dynamics in 5G core environments. The dataset, published on Zenodo, provides realistic temporal patterns of network function resource consumption and has been used to emulate CPU and memory utilisation profiles for the initial stages of the multi-agent workflow. This choice enables the future evaluation of the complete pipeline, from data ingestion and preprocessing to forecasting and orchestration, under realistic and reproducible conditions, without requiring access to proprietary operator data. Leveraging this dataset, the early experiments have focused on validating the end-to-end behaviour of the agentic architecture rather than quantitative forecasting accuracy, ensuring that each agent in the

chain (analysis, model selection, inference, and reconfiguration generation) operates coherently over realistic 5G core workload dynamics.

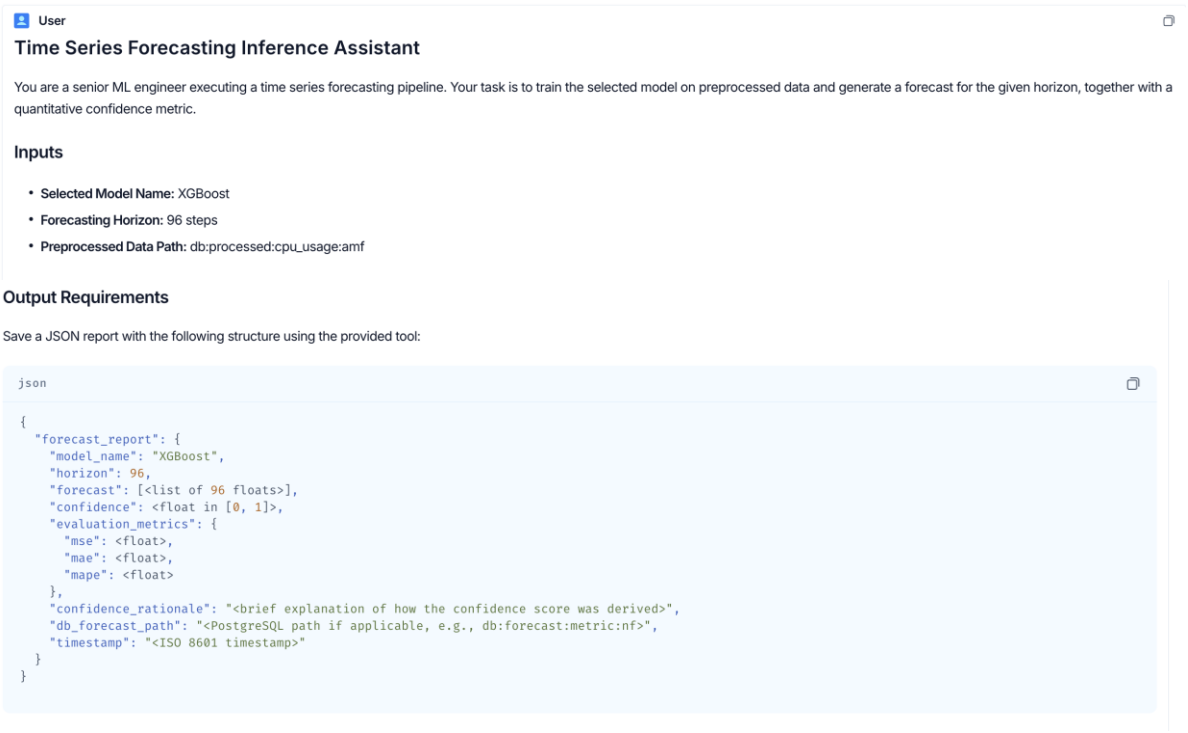


Figure 42: Examples of prompt engineering strategies used in the initial implementation of the multi-agent framework.

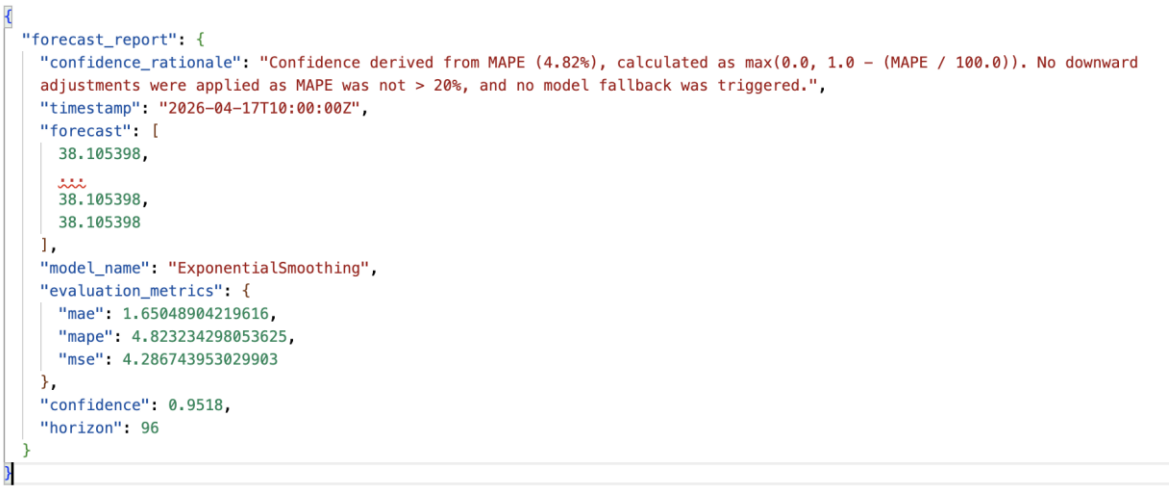


Figure 43: Example of an AI-generated forecast report produced by the Forecast Optimization Agent.

```yaml
apiVersion: v1
kind: ResourceQuota
metadata:
  name: cnf-default-quota
  namespace: cnf-default
spec:
  hard:
    requests.cpu: "1m"
    limits.cpu: "1m"
    requests.memory: "1Mi"
    limits.memory: "1Mi"
```

Figure 44: The figure shows a Kubernetes manifest snippet generated by the Reconfiguration Agent.

4 ALGORITHMS FOR SUSTAINABLE INTEGRATED NETWORKS AND ENERGY-EFFICIENT ENABLERS

4.1 INTRODUCTION

The algorithms presented in this section address a common challenge: how to make integrated 6G networks more sustainable, energy-efficient, and operationally robust while still meeting demanding service requirements. In a future network where access, transport, compute, and service orchestration are tightly coupled, performance can no longer be improved by optimizing each part in isolation. Instead, decisions must account for end-to-end effects across the radio access network, xHaul, edge-cloud resources, mobility behavior, traffic demand, and service constraints such as throughput, latency, reliability, and resilience. This section therefore gathers the algorithmic contributions developed in T4.2 around the network problems they target, rather than around organizational boundaries, in order to present a unified view of the technical directions pursued in the task.

A first family of problems concerns **network orchestration and planning**. In this context, the goal is to determine how network elements should be deployed, activated, dimensioned, and interconnected so that service requirements are satisfied at the lowest feasible cost and energy footprint. The need for such coordination is particularly strong in heterogeneous and partially meshed deployments, where a locally optimal choice at the radio or transport layer may create inefficiencies elsewhere in the system. Several algorithms in this section, therefore, focus on finding globally consistent configurations that improve utilization while avoiding over-provisioning and unnecessary operational expenditure.

A second family of problems concerns **dynamic resource management and planning**, which focuses on preserving QoS and QoE while minimizing avoidable energy use, signaling overhead, and reconfiguration cost. Indeed, even when the initial network design is efficient, the operating conditions of a 6G system may change rapidly due to traffic fluctuations, mobility, link degradation, weather effects, or failures. To cope with these variations, the algorithms in this section pursue predictive or closed-loop behavior, enabling the system to react before service quality is significantly degraded.

It is worth mentioning that some algorithms are not only intended to automate control, but also to help explain which variables drive cost, energy consumption, or reliability degradation in order to assess **sustainability and to support decision**.

This is particularly important in planning and optimization problems where operators need interpretable guidance on the trade-offs between performance, deployment cost, and energy

use. For that reason, the section includes methods that support prediction, evaluation, and optimization in a way that can inform network design choices and operational policies. Overall, the algorithms introduced in this section aim to provide a practical toolbox for sustainable integrated networking, combining efficiency, adaptability, and reliability in a unified framework.

4.2 STATE OF THE ART

In this section, we report the state of the art for the proposed algorithms divided according to the categories identified above.

4.2.1 Network orchestration and planning

4.2.1.1 Joint planning of access and xhaul resources

The literature on joint radio-access and transport-network optimization has moved well beyond isolated cell planning, but it still does not fully address the coupled cost-reliability-energy problem. Most existing approaches optimize a single layer, a single metric, or a reduced set of design variables, whereas the proposed approach (described in the following section) jointly considers site activation, user association, xHaul link activation, power allocation, bandwidth allocation, routing, and end-to-end reliability in one planning problem. Shehzad et al. [85] study backhaul-aware UAV positioning and terrestrial base-station association for fronthaul connectivity. Their main contribution is to show that access-side association decisions should be made jointly with transport-side feasibility, which is a useful precursor to cross-layer design. However, the work remains focused on a UAV-assisted access-support scenario and does not model infrastructure deployment costs, power budgeting, or end-to-end reliability as co-equal planning objectives.

[86] addresses joint user association and fronthaul/backhaul optimization in a multiple-UAV base-station setting. The work is important because it explicitly couples association, transport feasibility, and throughput performance, demonstrating that transport constraints can materially alter access decisions. Still, the optimization is centered on service performance rather than economic planning, and it does not jointly optimize CAPEX, OPEX, energy consumption, and reliability across a static terrestrial mesh. The proposed algorithm extends this idea by moving from a UAV-centric service model to a fixed candidate-site planning problem with explicit deployment and operating costs.

[87] analyzes 5G/6G optical fronthaul from a cost and energy perspective. It shows that cost-efficient fronthaul planning must account for both deployment and operational impacts, not only for technical feasibility. However, it is restricted to optical fronthaul and does not treat the full

access-xHaul interaction, wireless interference among candidate links, or the joint effect of user association and transport routing. The proposed algorithm builds on that cost-aware viewpoint but extends it to a wireless partial-mesh topology with reliability-aware access and transport decisions.

[88] provides a broad survey of wireless backhaul in 5G and beyond, and they clearly identify the main tension between deployment flexibility and the interference, capacity, and distance limits of wireless transport. It shows that backhaul cannot be designed independently from access in dense 6G systems.

[89] focuses on energy-efficient xHaul design guidelines and show that transport-layer choices strongly influence the energy footprint of a mobile network. Their contribution is important because it makes energy a first-class design criterion for 5G mobile xHaul. Yet the treatment is still largely transport-centric, and it does not jointly solve site deployment, user association, and reliability-constrained routing under a unified network-planning objective. The proposed algorithm generalizes this energy-aware perspective to the whole access-and-transport chain.

[90] studies optimized placement of DU/CU functions and routing in packet xHaul networks. This work is particularly relevant because it combines placement and flow routing, two elements that are also central to the proposed algorithm. However, the problem scope is still function-placement oriented, and it does not include the physical-layer type decisions that matter in the proposed algorithm, such as transmit power, bandwidth allocation, wireless interference, and end-to-end reliability over a partial mesh of candidate links. The proposed algorithm therefore broadens the design space from virtualized transport placement to full physical infrastructure planning.

Taken together, these works show clear progress toward cross-layer optimization, but three gaps remain. First, most studies optimize either access or transport in isolation, or treat one side as a fixed constraint rather than a decision space. Second, when energy or throughput is considered, CAPEX/OPEX and reliability are usually not modeled together with the same level of detail. Third, most methods do not jointly optimize the structural decisions that matter for a real operator: which sites to activate, which links to deploy, how to route traffic, how to allocate power and bandwidth, and how to guarantee service reliability end to end. The proposed algorithm, presented in Section 4.3.1.1, addresses these gaps by formulating the problem as a single multi-objective planning framework in which access and xHaul are optimized together. It explicitly captures the trade-off between sparse low-cost deployment and robust reliable connectivity, while allowing the planner to choose non-local associations when this improves the global solution. It also makes reliability a design constraint rather than an afterthought,

which is essential for 6G deployments where link availability, interference, and traffic heterogeneity can make a locally good solution globally infeasible.

4.2.1.2 Container Forecasting Engine

Energy efficiency is a central design objective of 6G networks, with infrastructure energy consumption expected to grow substantially as cloud-native deployments scale. In 6G architectures, network functions are fully containerised (CNFs) and dynamically scheduled across heterogeneous edge and cloud nodes. The energy footprint of this infrastructure is not fixed: it is directly shaped by how resources are allocated to running CNFs at any given time.

Two structural inefficiencies in conventional reactive orchestration contribute disproportionately to energy waste. First, over-provisioning: because a reactive orchestrator cannot anticipate demand, it allocates resources conservatively, keeping CPU and memory assigned well beyond the period they are actually utilised, so that capacity is available when a spike arrives. Allocated-but-idle compute translates directly into wasted energy. Second, cold-start operations: when a reactive orchestrator under-provisions and must scale up from scratch, initialising new container instances requires a burst of CPU and memory activity to restore service, incurring an energy-intensive transient before the CNF reaches steady state. Both inefficiencies are structural consequences of the reactive paradigm: the system reacts to demand rather than anticipating it, and the energy cost is unavoidable given that architectural constraint.

The state of the art for energy-efficient CNF orchestration in cloud-native networks reflects a transition from rule-based to learning-based approaches, but remains predominantly reactive. Threshold-triggered vertical scaling (e.g., Kubernetes Vertical Pod Autoscaler) adjusts resource limits based on observed utilisation, but provides no look-ahead and therefore cannot prevent the over-provisioning that builds up ahead of demand peaks. Horizontal Pod Autoscaler (HPA) with custom metrics partially addresses this by scaling replica counts, but again operates on current measurements. Recent work has explored RL-based autoscalers [68] that learn scaling policies from experience, achieving better adaptation over time, but these approaches still react to observed state rather than predicted future state. Predictive autoscalers such as LSTM-based forecasters [70] have demonstrated energy and latency gains in cloud workloads, but typically fix the forecasting model at deployment time and do not address the model-selection challenge arising from concept drift in heterogeneous 6G CNF workloads.

4.2.1.3 NTN-based task offloading

Recent research on 6G networks increasingly focuses on sustainability, efficiency, and adaptability, particularly in NTNs and O-RAN-based architectures. One major direction is the integration of Multi-access Edge Computing (MEC) to support latency-sensitive and energy-constrained services. Priority-aware MEC server placement strategies have demonstrated significant reductions in service loss for delay-critical applications while maintaining deployment and operational cost efficiency [91]. Moreover, co-locating MEC servers closer to O-RAN functional entities such as the O-DU and O-RU has been shown to substantially improve latency and energy efficiency under high traffic loads, making such placements especially suitable for dense and dynamic network environments.

Beyond server placement, the optimization of QoS and energy consumption has attracted considerable attention. Traditional methods often rely on static latency thresholds or global optimization objectives, which fail to capture user heterogeneity and time-varying network conditions. To address this limitation, AI and ML-based approaches have been proposed to enable per-user QoS adaptation within standardized 3GPP frameworks. AI-driven latency recommendation mechanisms allow joint optimization of energy efficiency and QoS, achieving improved performance in dense and heterogeneous scenarios [92].

Task offloading and resource scheduling represent another cornerstone of sustainable integrated networks. RL-based frameworks have been widely adopted to handle the stochastic nature of traffic and resources. For instance, Lyapunov-guided multi-objective RL approaches jointly optimize task scheduling, offloading decisions, energy consumption, and queue stability, achieving robust performance under dynamic conditions [93]. Similarly, in edge caching and content delivery, recommendation-enabled and AoI-aware caching schemes have been proposed to maximize long-term efficiency. Advanced DRL-based methods jointly consider user preferences, dynamic content popularity, cache replacement, and freshness, significantly improving system sustainability and performance [94].

In highly dynamic environments such as NTNs, task offloading becomes even more challenging due to mobility, fluctuating resources, and task dependencies. Recent DRL-based approaches incorporate time-space-varying resource models and dependency-aware task graphs to enable efficient task offloading in satellite-terrestrial integrated computing systems [95]. Moreover, AI-native O-RAN-based NTN architectures have demonstrated how open, virtualized, and disaggregated designs leveraging multi-tier RICs, dynamic functional splits, and cross-domain orchestration can effectively address challenges such as high latency, limited onboard resources, and dynamic topology [96]. These works collectively establish a strong foundation for sustainable and energy-efficient integrated networks leveraging NTNs and O-RAN principles.

4.2.1.4 A resource efficient cloud controller for large scale Wi-Fi networks

As data from real-world operational networks becomes increasingly accessible, there is a growing need for research focused on effectively leveraging this data for network optimization and management. Wi-Fi solution providers are particularly interested in advanced analytical tools to enhance their products, yet progress in this area has been limited. Addressing this gap, our work investigates the use of clustering and forecasting algorithms to unlock the potential of operational network data while addressing critical resource constraints inherent to large-scale environments that are managed by a central controller with limited memory and computational resources.

Time series clustering can be broadly categorized into three main approaches: distance-based [97], feature-based, and model-based methods [98]. Feature-based clustering extracts descriptive features from raw data to summarize key characteristics. Unlike distance-based methods, feature-based clustering is resilient to missing or noisy data and focuses on capturing overarching patterns rather than individual data points [99]. This robustness and interpretability make feature-based clustering an ideal choice for cloud-based network controllers operating under storage and computational constraints.

Recent research has explored various ways to combine clustering and deployed models. The approach proposed in [126] is to first divide the time series using clustering techniques (e.g. feature-based), and then fit available global models considering the series within each cluster. Alternatively, the work in [127] proposes grouping the series based on the minimum resulting forecast error of the assigned global model. The clustering algorithm is specifically designed to allocate the different time series so that the corresponding models best represent the existing prediction patterns.

4.2.1.5 Energy aware path computation algorithm for network slice provisioning

The accelerating adoption of AI-driven services, going from generative AI applications and real-time analytics to immersive extended reality (XR) and distributed inference, has triggered a structural shift in how traffic and computation are produced, placed, and transported across the edge–cloud continuum. Unlike traditional workloads, modern AI services are simultaneously compute-intensive and data-movement-intensive, frequently requiring repeated transfers of large datasets and intermediate representations between distributed accelerators, edge clusters, and centralized datacenters. As a result, energy demand is rising not only in compute facilities, but also across the transport and communication networks that interconnect them.

This evolution creates a dual requirement for network operators and infrastructure providers: sustaining high performance and service guarantees while simultaneously meeting sustainability and decarbonization targets. To date, network slicing mechanisms and slice-aware orchestration have been primarily designed around classical QoS metrics, e.g., latency, bandwidth, availability, and reliability. While these parameters remain essential, some others should be taken into account. In particular, energy consumption and carbon footprint are not direct outcomes of conventional transport path computation and resource allocation policies, which often default to minimal-hop or minimal-latency routing and do not internalize the environmental cost of infrastructure usage.

The proposed contribution advances beyond the current state-of-the-art by embedding sustainability as a first-class objective in transport slice provisioning. Specifically, it introduces an energy-aware path computation mechanism that provisions network slices while explicitly considering environmental constraints captured through Decarbonization Level Objectives (DLOs). The novelty lies in translating sustainability intents into a computable routing objective.

4.2.2 Dynamic resource management and planning

4.2.2.1 Conditional handover prediction

Heterogeneous networks are being massively deployed nowadays, due to the growing number of interconnected devices as well as different service demands and application scenarios. For this reason, future 6G networks integrate multiple RATs to ensure seamless connectivity under dynamic channel conditions. Within this framework, the execution and optimization of handover mechanisms between RATs have attracted a lot of attention. Current handover mechanisms typically react to instantaneous signal measurements leading to delayed decisions or ping-pong effects. Therefore, most recent studies leverage also ML aiming to overcome these limitations by proposing predictive mobility management. Authors of [100] propose a LSTM network to predict future user locations and trigger handover based on distance between the UE and the serving BS. In a similar context, [101] proposes a DNN to predict the probability of a cell to achieve the best signal quality and extend the 3GPP Conditional Handover (CHO). Both demonstrated high accuracy, while they also reduced latency and radio link failures. However, they remain intra-5G approaches, focusing only on horizontal handovers. Vertical handovers at multi-RAT environments, let alone cellular and WiFi coexistence, have not yet been so extensively studied. A worth mentioning work is presented in [102], where DRL is used to learn discrepancies between estimated and actual throughput, thereby guiding the selection between 5G and WiFi transmitters. This is a complex and not scalable solution, though, as DRL agent needs to be re-trained every time the network dynamics change.

4.2.2.2 Cell switching in integrated networks

Base stations (BSs) account for 60–80% of the total energy consumption in cellular networks, and this share is expected to grow as network densification accelerates toward the 6G of mobile communications [103]. Cell switching (CS), the selective deactivation of lightly loaded BSs whose traffic is offloaded to neighboring active nodes, is a way to reduce power consumption without additional hardware. The challenge lies in deciding *which* BSs to switch off, *when*, *where* to redirect users, and critically, *how long* to maintain the inactive state, all under dynamic traffic conditions and QoS constraints. Early CS approaches rely on deterministic heuristics. The *Sorting* strategy deactivates small base stations (SBSs) in ascending order of load until the macro base station (MBS) capacity is saturated [103]-[109]. While computationally lightweight (O(N log N)), it ignores heterogeneous power profiles across different BS types and provides no QoS guarantee for offloaded users. Rössler et al. [104] confirm with system-level simulations that switching off up to 50% of sectors yields a 40% power reduction at no throughput cost in low-load (nightly) conditions, but that further deactivations require accepting throughput degradation. Huang et al. [107] take a more dynamic approach in 5G heterogeneous networks (HetNets), combining Markov-based traffic prediction with a load-balancing cell association algorithm and dual-connectivity seamless handover. By anticipating traffic peaks and warming up sleeping femtocell gNodeBs (F-gNBs) proactively, their Dynamic F-gNB ON/OFF (DFOO) strategy partially addresses the toggling problem, a concern when frequent ON/OFF transitions induce signaling overhead and service interruptions.

The stochastic and time-varying nature of mobile traffic has motivated the adoption of RL. Tang [109] shows that Q-learning achieves 32–38% power savings in a beyond-5G (B5G) HetNet, outperforming heuristics by approximately 20% at large scale, with inference times under 0.1 s after offline training. Huang and Chen [105] couple a DQN agent with a convolutional neural network / recurrent neural network (CNN-RNN) traffic forecasting module: using predicted traffic statistics as the RL state significantly reduces MBS overload events and improves energy efficiency by 7–7.5% over DQN without forecasting, demonstrating the value of proactive, prediction-aware control. Rezaei and Ghahfarokhi [106] extend the problem to joint ON/OFF switching and power control and resource allocation (PCRA) in ultra-dense networks (UDNs), with each SBS acting as an independent multi-agent DQN agent; the approach outperforms all baselines in energy efficiency (EE), spectrum efficiency (SE), and QoS satisfaction. Rajule et al. [108] integrate LSTM-based traffic prediction with DQN control and an M/M/1 queuing model to bound blocking probability, achieving 31% average energy savings.

Despite these advances, purely terrestrial CS faces a fundamental bottleneck: once the MBS is saturated, no further SBSs can be deactivated regardless of their load. Moreover, users offloaded to a distant MBS may experience QoS degradation due to high terrestrial path loss under NLoS propagation. None of the reviewed works explicitly maximize the duration of the switch-off period to avoid repeated toggling.

HAPS, operating at approximately 20 km altitude as super macro base stations (SMBSs), offer unique advantages for CS: persistent line-of-sight (LoS) connectivity, coverage footprints up to 500 km in radius, and the ability to operate on solar energy [103]. By providing an additional high-capacity offloading tier, HAPS can relax the MBS capacity constraint that limits TN-only CS, while benefiting from favorable propagation conditions for users spread across the coverage area.

However, research on CS in integrated TN and NTN architectures remains scarce. Salamatmoghadasi et al. [103] propose the most comprehensive framework to date: a mixed-integer programming (MIP)-based joint optimization of BS states and user association in a vertical heterogeneous network (vHetNet), allowing offloaded users to be directed to either the MBS or the HAPS based on received power, under standardized 3GPP channel models for both terrestrial links (Urban Macro (UMa) model, TR 38.901) and HAPS links (LoS/NLoS model, TR 38.811). Their results demonstrate power reductions of up to 77% compared to the all-ON baseline at low load, far exceeding a terrestrial-only CS baseline. Prior HAPS-assisted CS works are more limited, relying on exhaustive search, load-sorting heuristics, or single-destination offloading to the HAPS only, and none of them use realistic 3GPP propagation models.

Table 7: SoA summary for traffic offloading for energy efficiency

Reference	**Network**	**Method**	**NTN offloading**	**Duration CS optimization**
[103]	TN + NTN	MIP optimization	Yes (HAPS)	No
[104]	TN	Load sorting	No	No
[105]	TN HetNet	DQN + DL forecast	No	No
[106]	TN	Multi-agent DQN	No	No
[107]	TN HetNet	Markov + heuristic algorithm	No	Partial
[108]	TN	LSTM + DQN	No	No
[109]	TN HetNet	Q-learning	No	No

Therefore, the SoA reveals a gap in addressing CS in integrated networks, as also highlighted in Table 7.

4.2.2.3 AI based energy efficient massive MIMO

The deployment of 5G networks and the design of future 6G architectures rely heavily on Massive multiple input multiple output (MIMO) technology to meet increasing demands for capacity and spectral throughput. However, this extreme densification of the number of antenna elements is accompanied by a dramatic increase in hardware energy consumption and the carbon footprint of the RAN infrastructure. Within the framework of sustainability objectives, it is imperative to rethink network sizing. The traditional approach of oversizing the network to guarantee QoS during peak hours is no longer energy efficient. The challenge, therefore, is to move towards a "Network Design-to-Impact" paradigm where the network topology and hardware configuration are co-optimized for energy efficiency.

The power consumption of a Massive MIMO base station depends not only on its radio frequency transmission power but is heavily influenced by the static power and consumption of the individual RF chains associated with each active antenna.

The central question addressed in the proposed work is: how can the overall energy consumption of an ultra-dense Massive MIMO network be drastically reduced without degrading the throughput experienced by users at the cell edge below a critical threshold? Conventional Sleep-Mode approaches simply put entire base stations into a temporary standby mode. Our approach, however, focuses on targeted and reconfigurable pruning of the least used MIMO base stations, based on spatial flow dynamics [110].

The rapid growth of mobile traffic, dense radio access deployments, and the use of massive MIMO have made energy efficiency a central design objective for 5G and future 6G networks. In 5G NR, high throughput is achieved partly through directional transmission and beamforming, but these mechanisms also create a dynamic control problem: the network must continuously decide which beams should serve which users, how resources should be shared, and when a user should be switched to another beam or base station. These decisions affect not only throughput, but also quality of service, signalling overhead, user fairness, and base station energy consumption.

4.2.2.4 Joint Beam-User allocation strategy

Conventional beam management and radio resource allocation methods often rely on heuristic rules based on instantaneous SINR, received power, or local throughput gain. Such methods are simple and computationally efficient, but they are usually myopic. A beam that looks optimal at a given moment may increase energy consumption, overload resources, trigger

unnecessary switching, or reduce service stability. This is particularly relevant in dense, mobile, and blockage-prone environments, where radio conditions change over time and short-term decisions can produce inefficient system-level behaviour.

Previous research has addressed parts of this problem from several directions. Load-dependent base station energy models show why radio decisions should be linked to realistic power consumption [111], while beam management studies highlight the complexity of directional communication and mobility in 5G/mmWave systems [112]. Multi-armed bandit methods have been successfully used for beam alignment, beam tracking, codebook selection, and rate adaptation [113]-[116]. Combinatorial bandits extend this idea to decisions where multiple actions must be selected jointly, which is closer to practical multi-user resource allocation [117]-[119]. Deep reinforcement learning has also been explored for energy-aware network control and beam/resource management [120]-[122], but such methods often require offline training and more complex policy design.

4.2.2.5 AI-empowered multivariate probabilistic forecasting a key enabler for sustainability in open RAN

Data-driven techniques have been recently considered for network traffic or resource prediction. Nevertheless, as pointed out in reference [124], there is currently a limited understanding of how traffic prediction errors, including forecasting inaccuracies or uncertainties, could affect power-saving strategies. Furthermore, it is often challenging to determine how improvements in the prediction of network traffic and/or resources could help in minimizing the power consumption in the RAN. Nevertheless, it has been recently shown that energy/power consumption is closely related to the requested radio resources. In particular, recent research in [125] has provided a realistic characterization of multi-carrier Base Stations, offering an analytical energy consumption model based on extensive data collection campaigns. In line with this study, references [125] and [129] have shown that, in modern BS, power consumption increases linearly with the PRB load, defined as the ratio of average used PRBs in a BS to the maximum number of PRBs available at the remote unit.

Accurate predictions of resource requirements, such as the utilization of radio (e.g., PRB) and/or computational resources, are crucial for efficient network operation. Over-utilization of resources can lead to unnecessary costs and energy consumption, while under-provisioning can impact performance and user experience. Previous studies have mainly focused on single-point forecasting models that provide deterministic predictions without taking into account the uncertainties inherent in the dynamics of the network. We propose a novel framework that integrates probabilistic forecasting techniques to enhance the power efficiency of O-RAN deployments. This framework utilizes advanced state-of-the-art AI-based forecasting models

to predict network conditions and resource requirements and enables dynamic and adaptive resource allocation. We conduct a thorough evaluation of several state-of-the-art multivariate forecasting techniques, including Multivariate LSTM, GPVAR, and TFT. These techniques can capture complex temporal dependencies and quantify prediction uncertainty.

4.3 ALGORITHMS DESCRIPTION

4.3.1 Network orchestration and planning

4.3.1.1 Joint planning of access and xhaul resources

This algorithm addresses the joint planning of access and xHaul resources in a wireless 6G deployment, with the objective of minimizing CAPEX, OPEX, and energy consumption while satisfying service constraints on capacity and reliability. The problem is formulated from the perspective of a mobile network operator that must decide which candidate sites to activate, how to associate users to access points, how to dimension the access and transport layers, and how to interconnect the network through a partial meshed wireless xHaul topology. This makes the algorithm directly relevant to sustainable network planning, where the design objective is not only to satisfy demand, but to do so with the lowest feasible cost and resource footprint.

The core novelty of the proposed approach is that it treats access and transport as a single coupled optimization problem rather than two independent design stages. In practice, the association of users to gNBs affects the traffic injected into the transport network, while the choice of xHaul links and their capacities constrain the feasible access configurations. This coupling is critical in 6G deployments, where the use of wireless xHaul, heterogeneous frequency bands, and partially meshed topologies creates nontrivial interdependencies between radio coverage, transport feasibility, interference, and energy cost. The algorithm therefore searches for configurations that balance these trade-offs in a globally consistent manner.

Accordingly, this algorithm advances the state of the art from isolated access or transport optimization toward a unified, reliability-aware, cost- and energy-efficient planning framework for integrated 6G network deployment.

The optimization problem uses a weighted objective function that minimizes the sum of CAPEX and OPEX. CAPEX covers the cost of activating gNB sites and deploying xHaul links, while OPEX accounts for operational energy-related costs at both the access and transport layers. The model includes variables for gNB installation, link activation, access transmit power, xHaul

transmit power, access beamforming, bandwidth allocation, and user association. It also includes constraints for user-to-gNB assignment, access capacity, xHaul capacity and multi-hop flow conservation, and link activation consistency. In the current formulation, xHaul reliability is derived from the link availability model and the end-to-end reliability is obtained from the selected path or set of feasible paths.

A key design choice is that user association is not restricted to the strongest local radio link. A UE may be intentionally associated with a gNB that is not the closest one, if this improves the overall end-to-end solution by reducing transport congestion, lowering system-wide power consumption, or increasing the reliability of the resulting path. This is an important distinction from local radio-centric policies, because a network that is optimal from the perspective of a single cell may be suboptimal at the system level. The HES-SO algorithm therefore implements a globally coordinated planning logic that can steer traffic in a way that is beneficial for the full network.

The algorithm is designed to operate on network snapshots, where each snapshot is represented by candidate gNB sites, core or aggregation nodes, feasible wireless links, link-interference relations, zone-level demand by SLA class, and link-reliability estimates. Rather than solving every instance through full combinatorial optimization and simulation, the proposed method leverages machine learning as a surrogate evaluator trained on simulated snapshot–configuration pairs. The ML model predicts the expected CAPEX, OPEX, energy consumption, and end-to-end reliability of each candidate design, thereby providing a fast feasibility and cost estimate that guides the search toward configurations that satisfy capacity and reliability constraints. In this way, the learning component does not replace the optimizer; instead, it accelerates it by filtering out poor candidates, ranking promising configurations, and enabling efficient exploration of the large joint design space created by site selection, user association, link activation, bandwidth allocation, and routing.

4.3.1.1.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

Within the UNITY-6G architecture, the proposed algorithm maps naturally to the AI-driven orchestration and planning capabilities of the management layer, specifically to the MS-AE-DE-ACT closed loop in D2.3 and the AIML subsystem described in D2.1. The algorithm converts an intent - “sustainable, reliable integrated network planning” - into concrete deployment and configuration decisions executed by HLO/LLO orchestrators, and it leverages the Data Continuum and Digital Twin to evaluate candidate designs offline before enactment.

From a KPI perspective, the algorithm is directly linked to CAPEX reduction, OPEX reduction, energy efficiency, resource utilization, served-demand ratio, and end-to-end reliability. It is also

relevant to load-balancing and traffic-steering indicators, because the user association and transport-routing decisions jointly determine how traffic is distributed across the network.

KPI mapping

- CAPEX reduction: target linked to KPI #17 - reduce CAPEX/OPEX via shared resources (expressed in the project as "Reduce CAPEX and OPEX with shared resources" KPI #17); expected planning-driven CAPEX savings are up to 15–25% for scenarios where infrastructure sharing and optimized link activation are applied.
- OPEX reduction: target KPI #38 - reduce OPEX by 30% through automation and improved link utilization; HES-SO contributes by producing plans that close the gap between static provisioning and demand-aware operation, enabling the 30% OPEX reduction target in PoC scenarios.
- Energy efficiency / integrated energy index: target KPI #43, increase network energy efficiency by a factor of 5 via distributed AI and energy-aware placement; HES-SO's energy-aware placement and workload steering are primary enablers for approaching KPI 43 in PoC evaluations.
- Link utilization: target KPI #20, increase link utilization from 40% to 80%; HES-SO directly optimizes user association and routing to hit the 40→80 utilization target by consolidating flows and avoiding over-provisioning.

In PoC-1 (Sustainable Networks for Disaster Handling / shared-resource scenarios) and PoC-2 (TN-NTN integration and Digital Twin validation), the proposed algorithm will be used both as an offline planning engine and as a pre-deployment validation module inside the AIML subsystem and the Data Continuum.

Offline planning PoC: the algorithm generates candidate network designs (active gNB set, x-haul topology, TX powers, bandwidth assignments, user association and multi-hop routing). Each candidate is evaluated in a Digital Twin (or ns-3 / equivalent simulator) producing KPI vectors (CAPEX, OPEX, energy consumption, link utilization, served demand ratio, E2E latency, packet error rate). The PoC compares HES-SO plans against three baselines: nearest gNB association, shortest-path transport selection, and fully-meshed overprovisioning. KPI differences are reported numerically (e.g., CAPEX %, OPEX %, energy kWh, link utilization %) for direct comparison.

Online/validation PoC: validated candidates are converted into deployment manifests used by the management layer (HLO/LLO/DMO) and, in a controlled PoC trial, activated in the testbed; runtime telemetry (SINR, utilization, queue occupancy, achieved rates, and RTT/jitter) is fed

back for closed-loop evaluation against KPI 6 (10 ms control loop), KPI#31 (service recovery < 180 s) and KPI#42 (DT-validated models not breaking time-sensitive operation in 99% of outputs).

4.3.1.1.2 Preliminary results

At this stage, the expected preliminary outcome is a planner capable of selecting a sparse but efficient partial mesh, activating only the sites and links needed to satisfy the traffic and reliability requirements. The most useful early results will show whether the algorithm reduces total cost and energy consumption compared with naive baselines while preserving service constraints. In particular, it should be possible to demonstrate that smarter user association and multi-hop transport planning can outperform simple local choices, especially when the network must trade off link length, interference, and power consumption.

For the first implementation phase, a practical workflow is to generate network snapshots, evaluate them with the optimization engine, and then validate the selected configurations in simulation. This workflow supports the extraction of cost, reliability, and energy KPIs while keeping the algorithm interpretable and adaptable. It also provides a natural bridge to future enhancements, such as surrogate modelling for fast feasibility estimation or heuristic search methods for large-scale deployment scenarios.

4.3.1.2 Container Forecasting Engine

The algorithm considered in this section, the CFE described in Section 3, addresses energy-efficient orchestration through an end-to-end pipeline that integrates energy-aware forecasting and allocation reasoning, thereby extending its role from proactive CNF resource management to sustainable resource allocation.

While Section 3 presents the CFE from the perspective of distributed intelligence and proactive CNF resource orchestration across the edge-cloud continuum, this section focuses on its energy-efficiency contribution. In particular, the CFE supports sustainable network operation by forecasting future CNF resource demand and by enabling the orchestrator to allocate CPU and memory only when needed, instead of relying on static over-provisioning or purely reactive threshold-based scaling.

The Forecast Optimization Agent continuously re-evaluates model suitability as the statistical properties of CNF workloads evolve, adapting from linear to non-linear models as traffic regimes change. More accurate forecasts directly reduce allocation error, which is the root cause of both over-provisioning (energy waste) and under-provisioning (cold-start energy cost).

Rather than mapping forecast values directly to resource limits through fixed scaling rules, the Reconfiguration Agent retrieves energy-efficient CNF allocation templates from its knowledge base and applies LLM-based reasoning to select and adapt the most appropriate template for the current forecast and system context. Templates encode energy-performance trade-off profiles for different CNF types, enabling context-sensitive decisions that would not be possible with static scaling policies.

By pre-allocating resources ahead of predicted demand peaks, rather than reacting to observed thresholds, the CFE eliminates the cold-start energy transient. The timing of pre-allocation is governed by a forecast horizon, which is set per CNF profile to match its known initialisation time, ensuring capacity is available when needed without holding idle resources longer than necessary.

This energy-oriented view does not introduce a separate algorithmic pipeline. Rather, it highlights how the same forecasting and reconfiguration workflow contributes to energy-efficient operation. By anticipating resource demand, the CFE can reduce unnecessary resource reservation during low-load periods, avoid computationally expensive cold-start operations during traffic peaks, and support timely scale-down decisions when predicted demand decreases. The telemetry processed by the Data Analytics Agent may include not only CPU, memory, and throughput metrics, but also energy-related indicators associated with CNF execution and node utilisation. These metrics can then be considered by the Forecast Optimization Agent and the Reconfiguration Agent when producing scheduling recommendations.

4.3.1.2.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

This algorithm will contribute achieving KPI#43 through the MS-AE-DE-ACT architecture, which enables coordinated, distributed, and AI-driven optimization across all layers of the integrated network. The architecture operationalizes a closed-loop control system that combines monitoring, predictive analytics, decision-making, and actuation to minimize total energy consumption while maintaining service quality and system performance under heterogeneous and dynamic 6G conditions.

The MS provides the foundation for KPI #43 by collecting multi-domain telemetry that extends beyond classical CNF metrics to include energy consumption indicators, traffic patterns, and available context-aware signals such as renewable energy availability and carbon intensity (“energy colour”). This enriched observability layer is essential to support energy-aware decision-making in distributed and resource-constrained environments. The Analytics Agent and the Forecast Optimization Agent, operating through the AE, transform monitored data into predictive intelligence. AE utilizes distributed AI agents, to reduce data movement overhead

and enable localized model adaptation across heterogeneous network segments. AE also performs forecasting of workload demand and renewable energy availability, enabling proactive resource optimization strategies aligned with future network conditions. The DE is responsible for translating analytical outputs into energy-aware orchestration policies. The Reconfiguration Agent within the DE optimizes end-to-end energy efficiency by jointly considering CNF placement, workload scheduling, and task offloading decisions across edge and cloud domains. It incorporates renewable energy awareness by dynamically aligning computational workloads with periods and locations of high renewable energy availability, while minimizing reliance on carbon-intensive energy sources. Additionally, DE integrates energy-aware scheduling of AI training workloads, reducing unnecessary execution during energy-critical periods. The ACT enforces these decisions by executing configuration updates across CNFs, Kubernetes clusters, and network functions, ensuring that energy-aware policies are effectively deployed in the operational environment.

4.3.1.3 NTN-based task offloading

This activity contributes to an energy-efficient resource optimization framework for O-RAN-enabled NTNs. The goal is to reduce energy consumption while maintaining low latency and reliable service delivery. The proposed approach focuses on intelligent resource management, including task offloading and computing resource allocation, under dynamic network conditions. By leveraging reinforcement learning, the system adapts its decisions based on traffic load, channel conditions, and user requirements. Unlike existing solutions that optimize energy or latency separately, the proposed approach jointly considers both aspects, resulting in improved efficiency and sustainability for NTN-assisted networks.

We consider a conventional NTN-based task offloading framework that focuses on energy-efficient and latency-aware computation without incorporating data-level processing mechanisms. In this system, UEs, a heterogeneous satellite layer, and a cloud MEC server collaboratively execute computational tasks under dynamic and resource-constrained network conditions. Each task is characterized by parameters such as data size, computational requirement, latency constraints, and priority levels and when offloading is selected, the complete task data is transmitted through the network. The satellite layer consists of regenerative satellites capable of limited onboard processing and transparent satellites that forward data directly to the MEC server, enabling flexible and adaptive execution paths.

At each time slot, UEs make intelligent decisions on whether to execute tasks locally or offload them based on channel conditions, task priority, queue states, and system load. High-priority tasks are prioritized for offloading to ensure timely completion, while lower-priority tasks may be processed locally to reduce communication overhead and conserve energy. This decision-

making mechanism allows the system to dynamically balance computation and communication resources, achieving efficient utilization of both UE and network capabilities.

Each UE maintains multiple queues for task arrivals, local processing, and offloading, and these queues evolve under stochastic task arrivals and service rates. The system is designed to ensure queue stability while minimizing energy consumption and delay. To achieve this, the problem is formulated as a long-term stochastic optimization problem that captures the trade-off between energy efficiency and performance. Since multiple UEs share limited satellite and MEC resources, the system inherently operates in a multi-agent environment, where users interact and compete for shared resources.

To address the complexity of long-term constraints, Lyapunov optimization is employed to stabilize the system queues and transform the problem into a sequence of per-slot decision problems. Virtual queues are introduced to enforce delay and backlog constraints, and the drift-plus-penalty framework enables a systematic trade-off between performance optimization and system stability. Furthermore, due to the high-dimensional and stochastic nature of the system, DRL is adopted to learn efficient control policies. The DRL agent observes system states such as channel conditions, queue lengths, and energy levels, and determines optimal actions including offloading decisions and resource allocation. The reward function is carefully designed to penalize energy consumption, delay, and congestion, thereby guiding the system toward energy-efficient and stable operation.

Overall, this framework provides an effective and scalable solution for energy-efficient task offloading in NTN environments, particularly in scenarios where reliable communication, low latency, and efficient resource utilization are critical. By jointly optimizing computation and communication decisions under stochastic dynamics, the system achieves strong performance without requiring additional processing complexity at the data level.

4.3.1.3.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed PoC demonstrates an AI-enabled semantic-aware NTN/O-RAN architecture aligned with the UNITY-6G D2.1 and D2.3 vision of AI-native, energy-efficient, and resilient 6G infrastructures integrating terrestrial and non-terrestrial networks. The considered system includes UE devices, a heterogeneous NTN layer with regenerative and transparent satellites, and O-RAN integrated MEC/cloud resources. The PoC focuses on intelligent semantic communication, adaptive task offloading, and AI-driven orchestration to jointly optimize latency, and energy efficiency. At the UE side, semantic compression is applied before transmission to reduce redundant communication overhead while preserving application-level information quality. AI-based decision making dynamically selects between local processing, satellite offloading, or MEC execution depending on queue conditions, available resources,

and SLA requirements. Lyapunov-based queue optimization is further employed to maintain queue stability and minimize AoI and delay.

The proposed algorithm will target:

- KPI #23: Reduction of communication overhead by a factor of 10. Achieved through semantic-aware encoding/compression that minimizes transmitted data while preserving required service quality.
- KPI #35: Improve NTN reliability ratio from *10^{-3}* to *10^{-4}*. Supported through intelligent satellite selection, adaptive routing, AI-assisted orchestration, and resilient multi-constellation NTN connectivity.
- KPI #45: Maintain SLA guarantees under semantic-aware communication. Ensured through adaptive semantic control, queue-aware optimization, and DRL-based resource management maintaining latency, reliability, and QoS constraints.

The PoC contributes to the development of intelligent NTN-integrated 6G architectures by combining semantic communication, AI-native orchestration, O-RAN integration, and energy-efficient edge/cloud-assisted service delivery within the UNITY-6G framework

4.3.1.3.2 Preliminary results

The complete design of the proposed adaptive compression and intelligent task scheduling framework is still in progress. Hence, there is still no full performance evaluation comparing the baseline and enhanced systems. At this stage, the aim of the preliminary analysis is to identify two important building blocks for efficient edge/cloud-assisted service delivery in the UNITY-6G framework: (i) characterization of the communication latency introduced by multi-hop NTN connectivity, (ii) evaluation of the energy feasibility of executing intelligent decision-making functions onboard regenerative satellites.

A first step in the design of efficient NTN assisted service delivery mechanisms is to understand the latency characteristics of the underlying satellite network. Figure 45 shows the end-to-end latency under different number of satellite hops including one-way latency and round-trip-time (RTT). As expected, latency increases with the number of satellites hops due to the cumulative propagation, forwarding and processing delays introduced at each satellite node. The one-way latency increases from about 40 ms for a single hop connection to almost 95 ms for a five-hop connection. Similarly, RTT increases from approximately 79 ms to 190 ms over the same range of hop counts. This result characterizes the fundamental behavior of multi-hop NTN communications and serves as a key metric reference for the proposed framework. The observed latency growth highlights the challenges of long-distance service delivery over satellite networks and motivates the need for latency-aware task placement, adaptive

compression, and intelligent orchestration mechanisms. Future evaluations will assess the potential of the proposed framework to reduce some of this latency overhead through optimized task execution and reduced communication payloads.

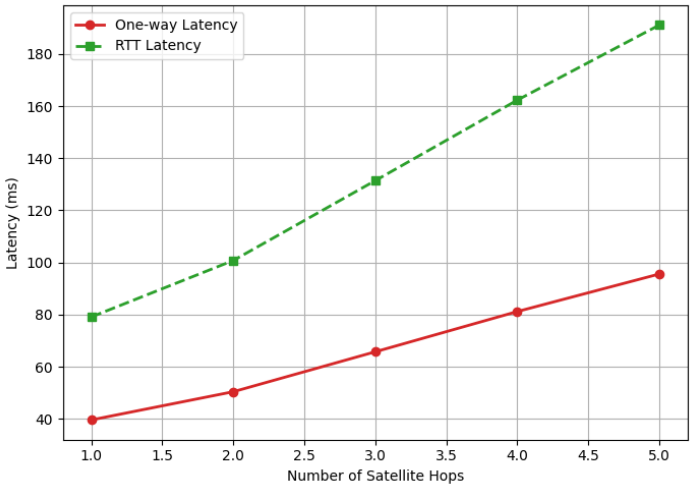


Figure 45: Latency Performance with Varying Satellite Hops

Although latency is one of the main challenges for the NTN-assisted delivering of services, the use of efficient decision-making functions onboard satellites brings another concern on energy sustainability. Regenerative satellites can assist in the task of scheduling, resource management, and service orchestration decisions in the presented UNITY-6G architecture. It is thus important to assess whether such computational functions can be sustained within realistic satellite energy budgets.LEO satellites operate in a cycle of sunlight and eclipse. Solar panels collect electrical energy during daylight and store it in onboard batteries, while during eclipse periods all satellite operations depend solely on stored battery energy. Thus, the inclusion of any further computational load shall be balanced with the consumption of the existing mission critical subsystem. We model the energy dynamics of a regenerative LEO satellite to evaluate the feasibility of onboard intelligence in terms of energy harvesting by means of solar energy, energy consumption by mission-critical subsystems, and the extra energy for onboard computation. Representative energy values for these components, taken from the literature, are summarized in Table 8.

Table 8: : Energy Consumption Model for LEO Satellite Operation

Component	Energy (Wh per cycle)
Harvested Solar Energy (H_j)	115 - 125 [18], [19]
Subsystem Energy ($E_{s,j} + E_{c,j}$)	85 - 95 [20], [21]
RL Computation Energy ($P_c\tau_j$)	3 - 5 [22]

An important point is that the mission-critical subsystems dominate the overall energy consumption, consuming around 85-95 Wh per orbital cycle, while the energy consumed by the RL-based scheduling and decision-making functions is relatively small, typically 3-5 Wh per cycle. This means that the impact on the aggregate satellite power budget of intelligent orchestration mechanisms is minimal. To study further the long-term operational feasibility, Figure 46 shows energy harvesting and consumption behavior over successive orbital cycles. These results are an important indication that learning-based orchestration functions can be executed onboard regenerative satellites while being compatible with realistic energy constraints. This is especially important for future NTN-integrated edge computing systems in which intelligent resource management is expected to play a key role in achieving efficient service delivery.

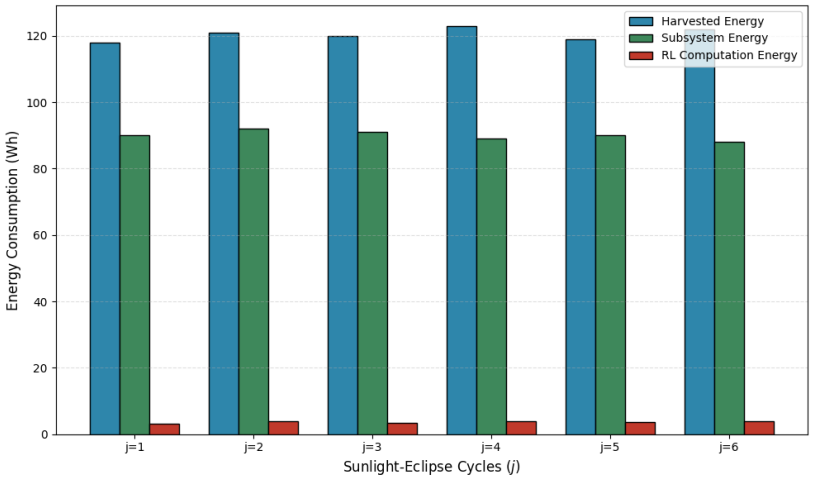


Figure 46: Energy harvesting and consumption in LEO satellite cycles

In general, the preliminary results establish communication and energy baselines on which the proposed framework will operate. The latency characterization highlights the communication challenge imposed by multi-hop NTN connectivity, while the energy analysis demonstrates the feasibility of deploying intelligent processing functions onboard satellites. Based on these observations, future work will study the integration of adaptive compression, energy-aware task offloading and intelligent service orchestration to jointly optimize latency, energy consumption and resource utilization in the UNITY-6G ecosystem.

4.3.1.4 A resource efficient cloud controller for large scale Wi-Fi networks

We consider a Wi-Fi cloud-based network controller that manages a large number of APs under storage and computational constraints.

As the number of APs increases, the scalability challenge becomes more pronounced, with the potential number of clusters and predictive models growing significantly.

Therefore, it is desirable to strike a balance between the deployed models (in terms of numbers, complexity and training) and resource utilization (e.g., memory) in the cloud.

[126][127]Both deploying a separate predictive model for each cluster or using highly complex models across all clusters may be infeasible for constrained storage and computational resources centralized controllers.

Exploiting clustering techniques, we propose (see Figure 47) an energy-efficiency-oriented deployment for constrained Wi-Fi network controllers.

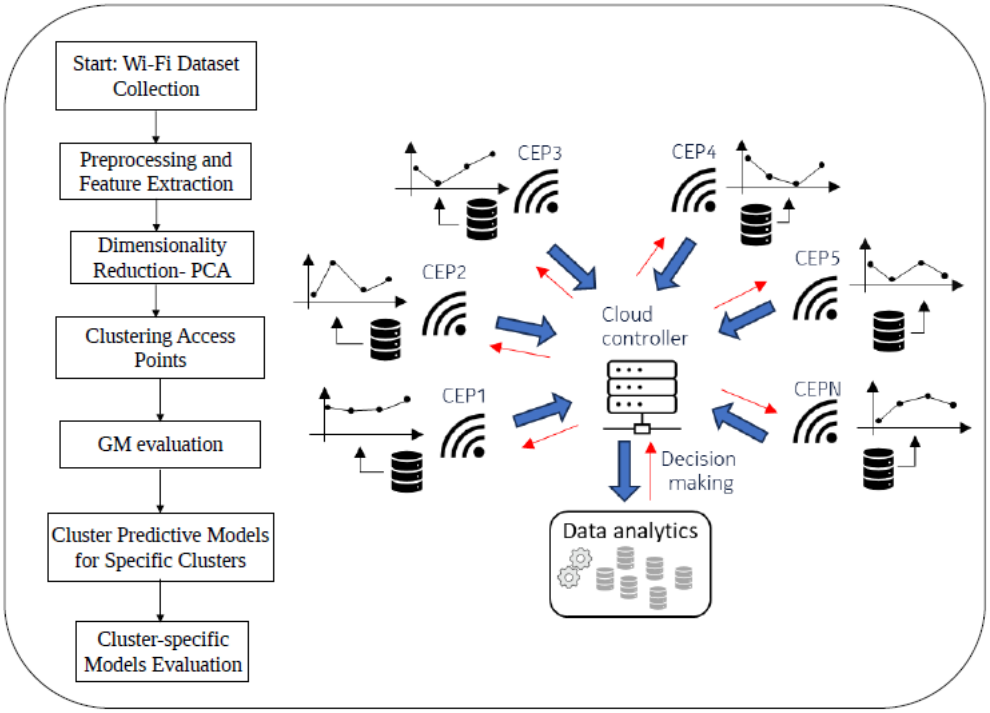


Figure 47: Proposed clustering and prediction analysis framework.

The clustering process consists of feature extraction and clustering. A set of features f such as trends, autocorrelation, and seasonality are derived to represent the time series S, emphasizing energy-related dynamics like low activity patterns. In the clustering phase, standard algorithms like k-means are applied to group the time series based on these features. To reduce the high dimensionality produced by this feature extraction process, Principal Component Analysis (PCA) is applied. This dimensionality reduction technique addresses feature redundancy by identifying the most informative relationships within the data, enabling meaningful clustering while reducing computational demands.

This framework achieves scalability by enabling selective model deployment: lightweight models serve simpler clusters, while specialized predictive models are deployed only for high activity clusters where their complexity is justified. By identifying clusters with similar behaviors, the algorithm minimizes the need for maintaining multiple redundant models, thereby optimizing memory usage.

Additionally, the clustering framework ensures that computational resources are allocated efficiently, avoiding the unnecessary execution of high-complexity models in low-activity or less critical clusters.

4.3.1.4.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed CFL algorithm maps naturally onto the Unity architecture across two domains of the Unification Layer. On the access side, Wi-Fi APs reside in the Non-3GPP RAN Domain, which is specifically designed to manage and orchestrate IEEE 802.11 APs in multi-RAT deployments. This domain provides the infrastructure through which APs can expose traffic load telemetry and receive control instructions from the Infrastructure Domain Manager. Each AP locally trains a model on its own traffic observations and participates in the distributed learning process without transferring raw data to the cloud controller.

The cloud-side intelligence instead can be naturally hosted in the AI/ML Domain. The cloud controller logic -aggregating locally trained model updates, applying the clustering to identify groups of APs sharing similar traffic distributions, and distributing cluster-specific models back to the APs for local inference - maps precisely onto the training, inference, and evaluation pipelines of this domain.

From a KPI perspective, direct contribution is done to KPI#43, that aims to increase the energy efficiency of the network through various AI techniques, energy-aware CNF placement and task offloading. The proposed framework finds a tradeoff between model complexity (and accuracy) and resource utilization.

4.3.1.5 Energy aware path computation algorithm for network slice provisioning

The proposed algorithm adopts the concept of Decarbonization Level Objectives (DLOs) for network slices, aligned with the framework introduced in [130], DLOs extend the conventional slice specification by capturing sustainability intents and measurable indicators that can be enforced or optimized during provisioning and lifecycle management.

The proposed DLOs are expressed through the following sustainability metrics:

- Energy Consumption (EC): total energy attributable to provisioning and carrying slice traffic, expressed in Joules or kWh. EC captures the absolute energy cost of supporting a slice, including baseline and load-dependent components where applicable.
- EE: energy efficiency of slice delivery, expressed in W/(bit/s). EE represents the relationship between delivered capacity (or performance) and power required to deliver it.
- Carbon Emissions (CE): carbon intensity associated with consumed electricity, expressed in gCO_2/kWh. CE captures the environmental impact of energy usage by accounting for the origin of electricity and the generation mix at the facility level.
- Usage of Renewable Energy (URE): renewable share of electricity used by the facility, expressed as a percentage or rate. URE provides a complementary indicator to CE by representing how much of the electricity is sourced from renewables.

Once these requirements are collected, the algorithm will calculate the optimal green paths from the energy metrics from the network nodes, which are:

- Power consumed by each node in idle state, measured in Watts, called P_{idle}
- Power consumed by components in nodes (e.g. transceivers, boards), measured in Watts, called $\mathrm{P}_{components}$
- Energy efficiency of each node, measured in Watts per bit per second [ee]
- Usage of renewable energy, measured as a rate. This data is specific to the plant where the node is located [ure]
- Carbon emissions, measured in grams of CO2 per kWh. This data is specific to the plant where the node is located [ce]

Taking into account the topology in the network, this algorithm builds a weighted graph of the topology. The weight of each node is named as Green Index [GI], measured in grams of CO2, and it defines how “Green” is each node in the topology. Then, it computes the shortest path by applying a Dijkstra Algorithm. The formula that the planner uses to compute the nodes’ GI is depicted below:

$$\boldsymbol{GI} = (\mathrm{P}_{idle} + \mathrm{P}_{components} + ee \times traffic) \times \frac{time}{1000} \times (1 - ure) \times ce$$

Where P_{idle}, $\mathrm{P}_{components}$ and ee*traffic are calculated based on the studies [131].

After assigning Green Index values, the algorithm computes an end-to-end route that minimizes cumulative environmental impact while respecting slice constraints. The green path computation is formulated as an additive shortest-path problem in which the path cost corresponds to the sum of GI contributions along the route. The green optimal path is computed as follows:

$$P_{opt} = \min_{P \in \mathcal{P}(A,B)} \sum_{v \in P} GI(v)$$

being *v* a node of the set

$$P = \{A, B, C, D, E, F, G\}$$

with the following restrictions:

$$\sum_{v \in P} ee(v) \leq EE$$

$$\forall v \in P,\ ure(v) \geq URE$$

$$\sum_{v \in P} ce(v) \leq CE$$

$$\sum_{v \in P} ec(v) \leq EC$$

By construction, this approach selects routes that preferentially traverse nodes with lower carbon impact and/or higher renewable usage, instead of defaulting to latency-only or hop-count-only optimizations.

In a slice orchestration context, this green path computation can be invoked as part of the slice provisioning workflow, either as a primary objective for “green slices” (DLO-constrained slices) or as a secondary optimization dimension under performance constraints.

While the current algorithm provides a practical and deployable baseline for sustainability-aware slice path computation, it adopts simplifying assumptions that will be addressed in subsequent iterations.

A primary limitation is the assumption of static values for CE and URE. In operational environments, both metrics can be highly time-dependent due to changes in electricity generation mix, demand response strategies, and regional grid conditions. Consequently, static CE/URE values may lead to suboptimal decisions when the carbon intensity of electricity exhibits strong temporal variability.

Future developments will therefore incorporate dynamic environmental metrics obtained from external monitoring systems, facility management platforms, and energy providers. This evolution will enable time-aware and context-aware computation of low-carbon network paths, supporting strategies such as carbon-aware routing aligned with renewable availability windows.

4.3.1.5.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

This algorithm operates in the Transport Management Domain as a submodule of the Domain Management Orchestrator, the Network Slice Controller.

It is mapped with the following KPIs:

- KPI#16. Reduce radio link power consumption up to 30% per link on average across a transport path (X-haul) due to full network coordinated load balancing, improved link utilization and distributed AI algorithms (between components O-RU, O-DU, O-CU links in Open RAN)
- KPI#43. Increase integrated network energy efficiency by a factor of 5 through distributed AI techniques, semantic communications, renewable energy usage, proactive strategies for resource allocation, energy aware CNF placement and task offloading, and energy aware scheduling of AI training and inference

4.3.1.5.2 Preliminary results

As an illustrative example, a network slice request incorporating sustainability-aware requirements through DLOs is considered. From this request, the following key parameters are derived:

- EC: 18000 Joules
- EE: 5 GigaWatt/bps
- CE: 650 gCO_2/kWh
- URE: 50%.

Based on these DLO specifications, the next step consists of extracting the corresponding energy-related metrics from the underlying transport network infrastructure (Figure 48). These metrics include node-level power consumption characteristics, energy efficiency profiles, and facility-level sustainability indicators. Using this information, the network topology is transformed into a weighted graph representation, where each node is assigned a Green Index reflecting its environmental impact. The resulting weighted topology is summarized in Table 9.

Finally, the proposed green path computation algorithm is executed over the weighted graph to determine the optimal route that minimizes the overall environmental impact while satisfying the slice requirements. The resulting green-optimal path is depicted in Figure 49.

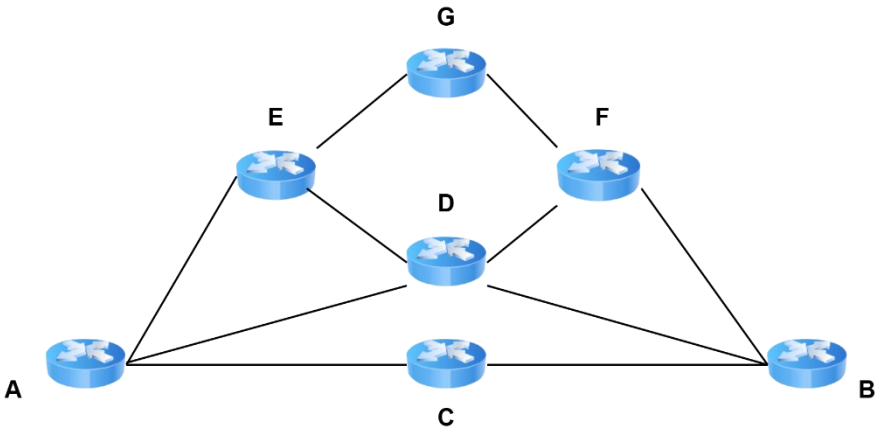


Figure 48: Network Topology

Table 9: Topology nodes Green Index

	A	B	C	D	E	F	G
GI	145	450	282	380	355	242	226

```
INFO - Planning optimal path from A to B with DLOS: {'EC': 18000, 'CE': 650, 'EE': 5, 'URE': 0.5}
INFO - Optimal path: ['A', 'C', 'B']
```

Figure 49: Optimal path computed

4.3.2 Dynamic resource management and planning

4.3.2.1 Conditional handover prediction

The proposed solution is a Predictive CHO (P-CHO) framework orchestrated by a RAT Steering Controller. In this way, it advances beyond the state-of-the-art as it switches from conventional reactive to proactive/predictive handover in multi-RAT environments. Instead of relying on instantaneous signal measurements and threshold-triggered events, the framework anticipates short-term signal evolution and enables early mobility management, thus improving not only responsiveness but also reliability. In addition, it uses RAT-aware predictors to capture heterogeneous channel characteristics and forecast signal quality. These predictions are incorporated in a complete control loop that prepares both horizontal (6G) and vertical (6G/Wi-Fi) handover events. Learning-driven prediction along with control logic improves over previous methodologies where ML aspects are often isolated from the handover process. Moreover, it introduces a hysterisis-enabled mechanism which requires the predicted performance gains

to hold for consecutive time steps before triggering a handover. This helps improving network stability and reducing unnecessary handovers.

The system model under consideration is a multi-RAT environment that consists of two wide-area cellular BSs and two overlapping short coverage Wi-Fi APs in each BS's cell area, as depicted in Figure 50. The system simulates the transmission of downlink signals that follow a generic distance-based attenuation model for the 6G and a short-range indoor hotspot communication with multipath and wall penetration for Wi-Fi. There are also UEs that traverse predefined trajectories within the topology with constant speed. The trajectories are chosen in a way that captures different areas, directions, and RAT overlaps.

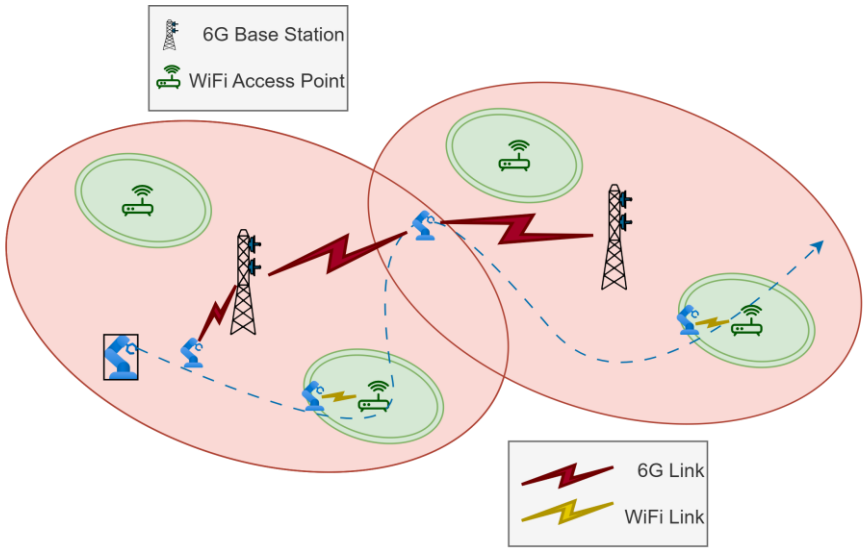


Figure 50: Random UE trajectory in a Multi-RAT HetNet [128]

As far as the algorithm is concerned, its core lies in predicting future signal indicators based on historical measurements. Let suppose SINR for the rest of the description. For each RAT type, a different model variant is used:

- Cellular 6G networks: Bidirectional LSTM (BiLSTM) to capture complex temporal dependencies (e.g., interference, fading).
- IEEE 802.11 WiFi networks: Lightweight LSTM, reflecting simpler channel dynamics.

Assuming a trajectory t and a specific time-step k, then x_k is the feature vector. The models try to learn the next-step SINR: $\hat{s}_{k+\tau}^{(r)} = f^{(r)}(x_k)$, for every RAT r where $f^{(r)}$ is the learned function for RAT r and τ is the forecasting horizon. The RAT Steering Controller gathers all these predictions and utilizes a Decision Module to select the target serving RAT as:

$$r_k^{\cdot} = \arg\max \hat{s}_{k+\tau}^{(r)}$$

A handover is triggered when the predicted gain of a target RAT over the current one exceeds a threshold:

$$\hat{s}_{tar}^{(r)}[k+\tau] - \hat{s}_{curr}^{(r)}[k+\tau] \geq \delta_{QoS}$$

This transforms traditional static thresholds into time-aware decision criteria which can be more or less sensitive to handover by choosing the appropriate δ_{QoS} for each objective. To improve stability, the framework introduces also a hysteresis condition. In other words, the predictive gain must hold for multiple consecutive time steps before triggering a handover.

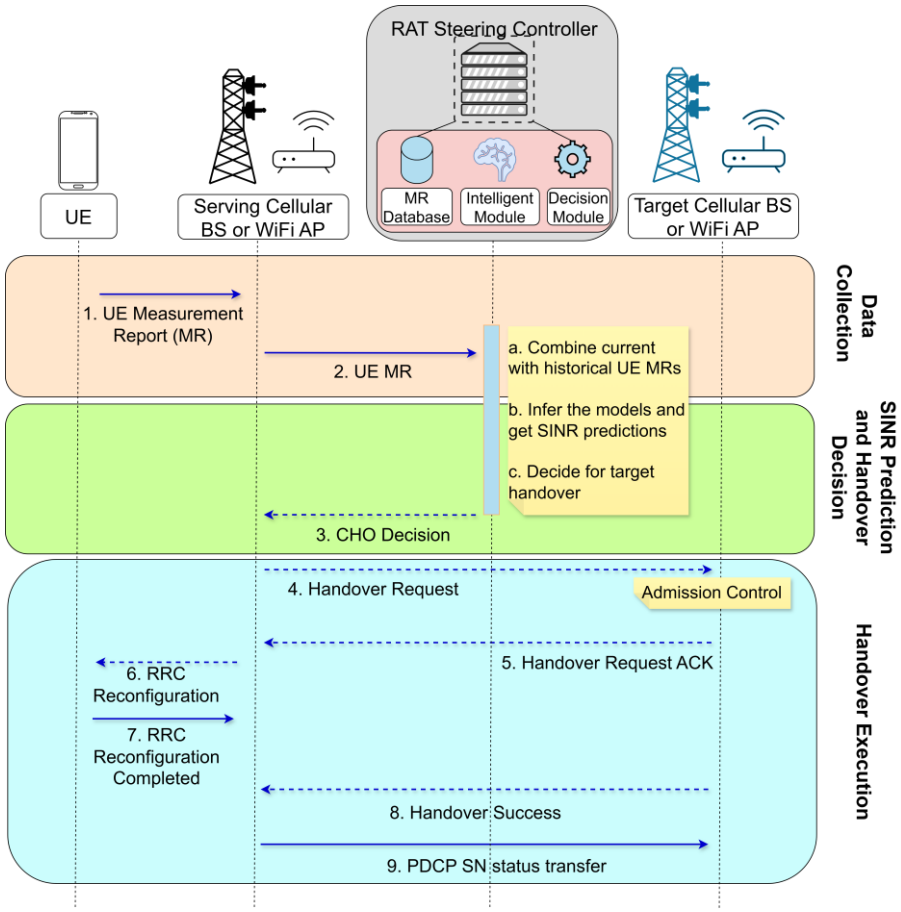


Figure 51: Proposed P-CHO workflow [128]

Overall, the system operates in three stages, according to Figure 51:

1. Data Collection: The UE periodically reports measurements (e.g., RSSI, SINR, throughput) from all candidate RATs and store them in a timeseries database.

2. Prediction and Decision: LSTM pre-trained models predict short-term signal quality for each RAT. These predictions are evaluated to determine whether a condition with predefined threshold is satisfied and for how many consecutive time-steps.

3. Handover Execution: If yes, the handover is executed following standard 3GPP procedures.

4.3.2.1.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

Intelligent P-CHO algorithm contributes to Use Case 4 focusing on Multi-RAT O-RAN enabled industrial networks for time sensitive applications. In this context, the solution is going to be also integrated into PoC 2 ES 2, which aims to demonstrate the resource optimization of a multi-RAT time sensitive deployment by utilizing AI/ML techniques. More specifically, the ML pipeline of intelligent handover algorithm presented above, will be deployed as an xApp or dApp within the PoC 2 testbed to predict the steering decisions between the 6G cellular network and IEEE 802.11 technologies. Based on real-time and historical measurements, it will recommend network reconfiguration actions in order to optimize connectivity, reliability, and latency performance. To evaluate the effectiveness of the algorithm as part of the whole multi-RAT system, it is mapped with the following KPIs:

- KPI #2: Maintain end-to-end latency (smaller than 20 ms) and reliability guarantees (99% of PDR) of time sensitive applications in a dynamic NPN context. The P-CHO mechanism minimizes service interruption and packet losses during RAT transitions by anticipating signal degradation before it occurs. The KPI will be evaluated separately for 6G and WiFi during mobility scenarios and dynamic network conditions.
- KPI #4: Real-time reconfiguration of IEEE 802.11 network domain in range smaller than 10 ms from taking the decision.

  The algorithm enables fast decision-making for UE-WiFi association by continuously forecasting signal quality. The measurement focuses on the latency between xApp/dApp decision until the actor that configures the AP.
- KPI #5: Real-time reconfiguration of. cellular network domain in the range smaller than 10 ms from taking the decision.

  Similarly, proactive prediction of cellular channel conditions enables rapid reconfiguration of the 6G domain, reducing handover delays. The measurement focuses on the latency between xApp/dApp decision until the actor that configures the O-RU.
- KPI #7: Cross-domain (cellular – IEEE 802.11) end-to-end latency smaller than 20 ms.

  Since the solution operates across heterogeneous RATs, the intelligent steering mechanism aims to maintain seamless service continuity between cellular and IEEE 802.11 domains. The KPI is assessed by monitoring latency during data traffic transmission from

an end-device that is connected to WiFi to another device that is connected to 6G-RAN network.

- KPI #8: Cross-domain jitter of max of 2 ms, with less than 10% of the packets missing deadline.

  Same concept applies also here. Jitter refers to differences of experienced latencies of consecutive packets, that are anticipated to be reduced due to the predictive nature of the solution.

4.3.2.1.2 Preliminary results

At this point, the initial evaluation of the algorithm relies on a custom Python simulator that uses realistic channel models and UE movement. Within this framework, we have produced some preliminary results, focusing on the behavior and optimization of the prediction models, before the final trajectory-based steering evaluation. For all the plots the yellow lines on the left correspond to the BS's model, while the blue lines on the right correspond to the AP's model.

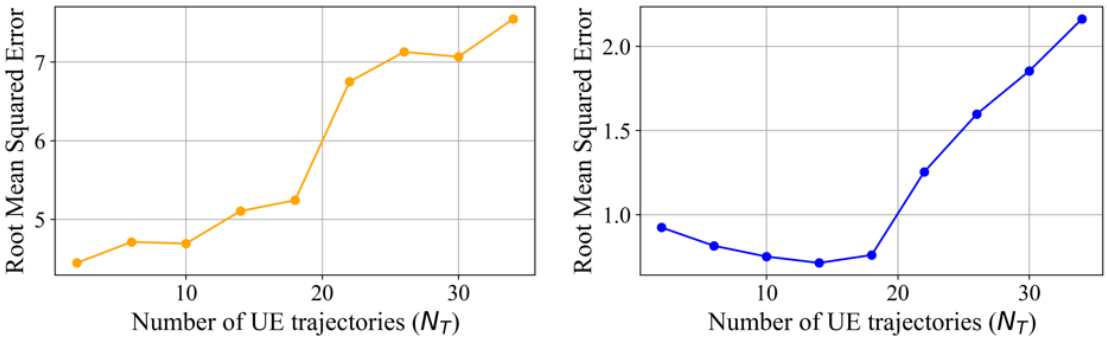


Figure 52: Prediction RMSE versus number or UE paths

One interesting finding concerns the impact of the training trajectory diversity on prediction accuracy, depicted in Figure 52. Experimental analysis showed that by increasing the number of training trajectories, the dataset becomes more diverse, and the model can learn more realistic mobility patterns. However, this results in increased Root Mean Squared Error (RMSE), as by keeping the topology stable, the more the trajectories, the more the points and areas in common, a fact that confuses both models. Consequently, it is important to use a balanced and representative mobility dataset.

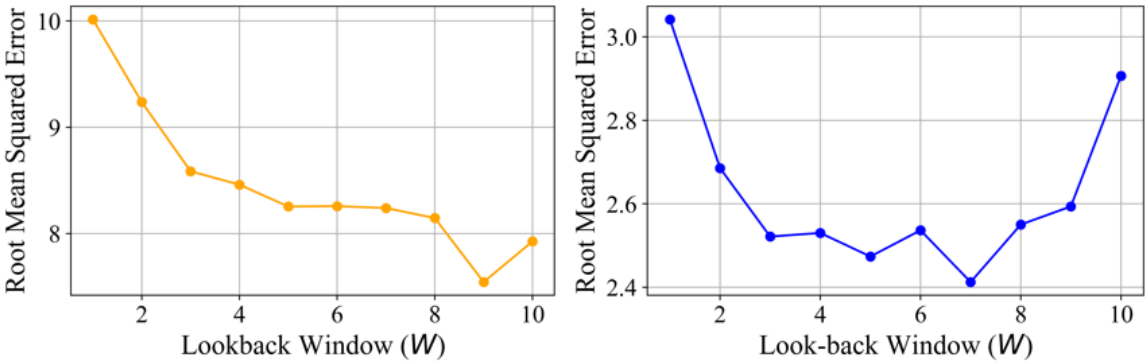


Figure 53: Prediction RMSE versus look-back window size

Additional evaluations investigated the impact of look-back observation window on the forecasting performance. As can be shown from Figure 53 longer look-back window provides more historical information for the model and as a result better performance in general. Increasing the window size a lot, though, increases the computational complexity and may introduce unnecessary information. So, here we have found the window sizes that balance prediction accuracy and model efficiency and these are 9 past samples for the BSs and 7 for the APs.

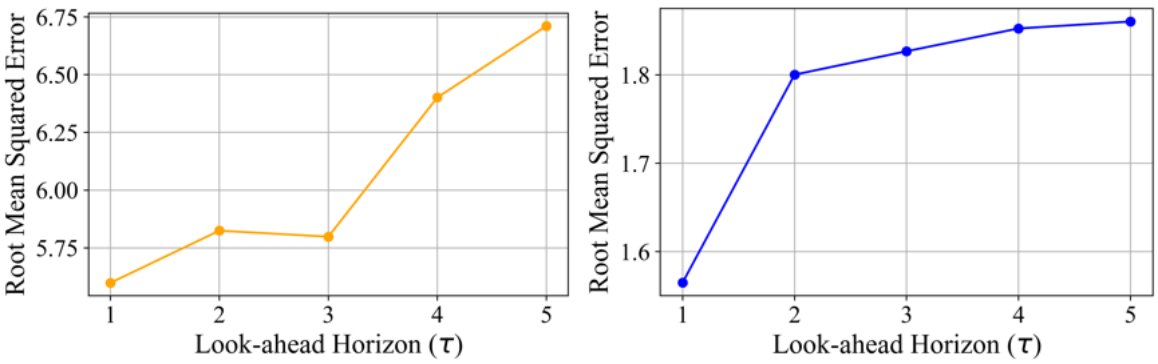


Figure 54: Prediction RMSE versus look-ahead horizon size

We have also evaluated multi-step prediction models. For this purpose, we extend the dataset with more target columns for more future steps. So, the models are capable of forecasting multiple future SINR values at once. Figure 54 reveals that predicting multiple steps ahead introduces more uncertainty compared to single-step prediction, but allows the system to anticipate future conditions earlier, therefore it is interesting to understand the trade-off between prediction accuracy and decision latency. These insights will be critical in the testbed evaluation phase, because earlier handover supports directly the KPIs related to low latency and fast reconfiguration in time-sensitive industrial applications.

4.3.2.2 Cell switching in integrated networks

We consider the downlink transmission of an integrated terrestrial and non-terrestrial network composed of a set of terrestrial gNBs and one NTN satellite node. The set of terrestrial gNBs

is denoted by $\mathcal{B} = \{1,2,\dots,B\}$, where B is the total number of terrestrial gNBs. The set of UEs is denoted by $\mathcal{U} = \{1,2,\dots,B\}$, where the number of UEs is assumed to be equal to the number of gNBs. The NTN satellite node is denoted by s.

Each UE is initially associated with the terrestrial gNB that provides the highest RSRP. Therefore, for each UE u, the initial serving gNB is given by:

$$b_u^0 = \underset{b\in\mathcal{B}}{\operatorname{argmax}} RSRP_{u,b} \tag{1}$$

where $RSRP_{u,b}$ denotes the RSRP measured by UE u from gNB b.

In this framework, the UEs are assumed to be distributed in such a way that, after highest-RSRP association, there exists a one-to-one mapping between the set of gNBs $\mathcal{B}$ and the set of UEs $\mathcal{U}$ with identical indices. Hence, UE $u = b$ is associated with gNB b, and this UE is denoted by u_b.

Therefore, each gNB has exactly one associated UE. If gNB b is active, it serves UE u_b through the terrestrial network. If gNB b is switched off, UE u_b is offloaded to the NTN satellite.

This simplified assumption allows the joint cell-switching and traffic-offloading problem to be represented using a single binary decision variable per gNB.

Each terrestrial gNB is assumed to have bandwidth B_{TN}, and the NTN satellite has bandwidth B_{NTN}. In this framework, we assume

$$B_{TN} = B_{NTN} = B. \tag{2}$$

All terrestrial gNBs and the NTN satellite reuse the same bandwidth B. Therefore, the NTN bandwidth is not orthogonal to the TN bandwidth, and satellite and terrestrial transmissions may interfere with each other.

Each UE requests a bandwidth denoted by B_u^{req}. In this simplified framework, all UEs request the same bandwidth, equal to the bandwidth of the terrestrial gNB and the NTN satellite:

$$B_u^{req} = B_{TN} = B_{NTN} = B, \quad \forall u \in \mathcal{U}. \tag{3}$$

Since each active gNB serves only one UE, the terrestrial bandwidth allocated to UE u_b is

$$B_{u_b,TN} = B_{TN}. \tag{4}$$

Only one UE can be offloaded to NTN in this simplified framework. Equivalently, only one gNB can be switched off. Therefore, if UE u_b is offloaded to NTN, the bandwidth allocated to that UE by the satellite is

$$B_{u_b,NTN} = B_{NTN}. \tag{5}$$

Cell-Switching and Offloading Decision Variable

The binary decision variable is defined as:

$$z_b \in \{0,1\}, \quad \forall b \in \mathcal{B} \tag{6}$$

where $z_b = 1$ if gNB b is ON and serves UE u_b, and $z_b = 0$ if gNB b is OFF and UE u_b is offloaded to NTN.

Thus, the terrestrial and NTN association indicators can be written as:

$$x_{u_b,TN} = z_b, \quad x_{u_b,NTN} = 1 - z_b. \tag{7}$$

Therefore, in this framework, the cell-switching decision and the traffic-offloading decision are coupled and represented by the same binary variable z_b.

Terrestrial SINR Model

When UE u_b is served by gNB b, its terrestrial signal-to-interference-plus-noise ratio is given by:

$$\mathrm{SINR}_{u_b,TN} = \frac{P_b h_{b,u_b}}{\sum_{\substack{j\in\mathcal{B} \\ j\neq b}} z_j P_j h_{j,u_b} + (1 - z_k) P_{s,k} h_{s,u_b} + N_{u_b,TN}} \tag{8}$$

where $k \in \mathcal{B}$ denotes the index of the gNB that is switched off, if any. Here, P_b is the transmit power of serving gNB b, h_{b,u_b} is the channel gain between gNB b and UE u_b, P_j is the transmit power of interfering gNB j, h_{j,u_b} is the channel gain between interfering gNB j and UE u_b, $P_{s,k}$ is the satellite transmit power allocated to the offloaded UE associated with switched-off gNB k, h_{s,u_b} is the channel gain between the satellite and UE u_b, and $N_{u_b,TN}$ is the noise power at UE u_b on the terrestrial link.

The term z_j indicates whether interfering gNB j is active. If $z_j = 0$, gNB j is switched off and does not contribute interference. Since the NTN satellite reuses the same bandwidth as the

terrestrial gNBs, the satellite transmission also contributes interference to terrestrial UEs when one UE is offloaded to NTN.

In this simplified framework, the terrestrial transmit powers, satellite transmit powers, and noise powers are treated as fixed environment parameters rather than optimization variables.

NTN SINR Model

When UE u_b is offloaded to NTN, its satellite signal-to-interference-plus-noise ratio is given by:

$$\mathrm{SINR}_{u_b,NTN} = \frac{P_{s,u_b} h_{s,u_b}}{\sum_{j\in\mathcal{B}} z_j P_j h_{j,u_b} + N_{u_b,NTN}}. \tag{9}$$

Here, P_{s,u_b} is the satellite transmit power allocated to UE u_b, h_{s,u_b} is the channel gain between satellite node s and UE u_b, P_j is the transmit power of active terrestrial gNB j, h_{j,u_b} is the channel gain between gNB j and UE u_b, and $N_{u_b,NTN}$ is the noise power at UE u_b on the NTN link.

Since the NTN bandwidth is reused with the TN bandwidth, interference is considered between terrestrial and satellite links. Therefore, the satellite link quality is modeled using SINR rather than SNR.

The achievable terrestrial rate of UE u_b is given by:

$$R_{u_b,TN} = B_{TN}\log_2\left(1 + \mathrm{SINR}_{u_b,TN}\right). \tag{10}$$

The achievable NTN rate of UE u_b, when it is offloaded to the satellite, is given by:

$$R_{u_b,NTN} = B_{NTN}\log_2\left(1 + \mathrm{SINR}_{u_b,NTN}\right). \tag{11}$$

Since only one UE can be offloaded to NTN, the offloaded UE receives the full NTN bandwidth B_{NTN}.

The total system throughput is defined as the sum of the rates achieved by all UEs, considering whether each UE is served by its terrestrial gNB or offloaded to NTN:

$$T_{total} = \sum_{b\in\mathcal{B}} \left[z_b R_{u_b,TN} + (1 - z_b) R_{u_b,NTN}\right]. \tag{12}$$

If $z_b = 1$, UE u_b contributes through its terrestrial rate $R_{u_b,TN}$. If $z_b = 0$, UE u_b contributes through its NTN rate $R_{u_b,NTN}$.

All gNBs are assumed to have the same constant ON-state power consumption, denoted by P_{ON}, so that $P_{b,ON} = P_{ON}$ for all $b \in \mathcal{B}$.

Therefore, the total terrestrial gNB power consumption is:

$$P_{consumed} = P_{ON} \sum_{b \in \mathcal{B}} z_b. \tag{13}$$

Since P_{ON} is assumed to be constant and identical for all gNBs, but its exact numerical value is not required in this framework, the number of active gNBs, $\sum_{b \in \mathcal{B}} z_b$, is used to represent terrestrial power consumption in the objective function. The unknown constant P_{ON} is absorbed into the weighting coefficient of the objective.

The power saving due to switching off gNBs can be written as:

$$P_{saving} = \sum_{b \in \mathcal{B}} (1 - z_b)\, P_{ON}. \tag{14}$$

The satellite energy consumption is not included in the optimization objective because the satellite is assumed to use solar energy. Therefore, only terrestrial gNB power consumption is considered in the energy-related part of the formulation.

Switching a gNB ON or OFF may introduce additional transient energy consumption. In this framework, switching a gNB ON is assumed to be significantly more costly than switching it OFF. Therefore, instead of penalizing every switching operation, the switching-cost term penalizes only OFF-to-ON transitions.

The switching cost is defined as:

$$C_{switch} = \sum_{b \in \mathcal{B}} \sum_{t=1}^{T} \big(1 - z_b(t-1)\big)\, z_b(t). \tag{15}$$

Here, $t \in \{1,2,\dots,T\}$ is the time-step index, T is the total number of time steps in one episode, $z_b(t)$ is the ON/OFF state of gNB b at time step t, and $z_b(t-1)$ is its state at the previous time step.

The term $\big(1 - z_b(t-1)\big)z_b(t)$ is equal to 1 only if gNB b is OFF at time step $t-1$ and becomes ON at time step t. Otherwise, it is equal to 0.

Thus, for each gNB:

- an ON-to-OFF transition is not penalized;
- an OFF-to-ON transition results in a switching cost;
- no state change results in zero switching cost.

This creates a soft switching inertia that allows switching while discouraging frequent reactivation of switched-off gNBs.

The coefficient μ, used later in the objective functions, represents the penalty associated with switching a gNB back ON. This penalty can be related to the transient power or energy consumed by a gNB during an OFF-to-ON transition. Since the exact switching energy is not explicitly modeled, it is absorbed into μ. Therefore, the term μC_{switch} discourages unnecessary reactivation of gNBs while keeping the framework simple.

Objective Function Options

Two objective-function options are considered for Framework 1.

The first objective maximizes the total system throughput while penalizing the number of active gNBs and repeated switching.

Starting from the power-consumption expression in (13), the original weighted power penalty can be written as $\beta P_{consumed} = \beta P_{ON} \sum_{b\in\mathcal{B}} z_b$.

Since P_{ON} is constant and unknown, it is absorbed into the coefficient by defining $\lambda = \beta P_{ON}$.

The resulting objective is:

$$\max_{\{z_b\}} T_{total} - \lambda \sum_{b\in\mathcal{B}} z_b - \mu C_{switch}. \tag{16}$$

where λ controls the penalty for keeping gNBs active, and μ controls the penalty for switching gNBs back ON.

Substituting the expression of T_{total}, the objective becomes:

$$\max_{\{z_b\}} \sum_{b\in\mathcal{B}} \left[z_b R_{u_b,TN} + (1 - z_b) R_{u_b,NTN}\right] - \lambda \sum_{b\in\mathcal{B}} z_b - \mu C_{switch}. \tag{17}$$

Pros

- Simple and practical.
- Easy to use later as an RL reward.

- Directly controls the throughput-active-gNB trade-off.
- Does not require the exact value of P_{ON}.
- Less risky than pure energy-efficiency maximization because it still rewards absolute throughput.
- The switching-cost term discourages unnecessary OFF-to-ON transitions without forcing a hard switching limit.

Cons

- λ must be tuned carefully.
- μ must also be tuned carefully.
- Throughput, active-gNB count, and switching cost have different units/scales.
- Interpretation of coefficients is less direct.

The second option is a Scaled Energy-Efficiency with Switching Cost

The energy efficiency is defined as:

$$EE = \frac{T_{total}}{P_{consumed}}. \tag{18}$$

Using (13), we have:

$$EE = \frac{T_{total}}{P_{ON} \sum_{b \in \mathcal{B}} z_b}. \tag{19}$$

Since P_{ON} is constant and identical for all gNBs, maximizing EE is equivalent to maximizing the scaled energy-efficiency metric:

$$\widetilde{EE} = \frac{T_{total}}{\sum_{b \in \mathcal{B}} z_b}. \tag{20}$$

This metric represents the throughput achieved per active gNB. It is not the exact physical bits/Joule value unless P_{ON} is known, but it preserves the same optimization preference when P_{ON} is constant.

The corresponding objective with switching cost is:

$$\max_{\{z_b\}} \widetilde{EE} - \mu C_{switch}. \tag{21}$$

Substituting the expression of T_{total}, this becomes:

$$\max_{\{z_b\}} \frac{\sum_{b\in\mathcal{B}} \left[z_b R_{u_b,TN} + (1-z_b) R_{u_b,NTN}\right]}{\sum_{b\in\mathcal{B}} z_b} - \mu C_{switch}. \tag{22}$$

Pros

- Does not require the exact value of P_{ON}.
- Physically motivated by energy efficiency because active gNB count represents terrestrial power consumption.
- No need to choose a unit-conversion weight β in the original power-consumption term.
- Naturally encourages switching off unnecessary gNBs.
- The switching-cost term discourages unnecessary OFF-to-ON transitions.

Cons

- $\widetilde{EE}$ is a scaled energy-efficiency metric, not the exact physical bits/Joule value unless P_{ON} is known.
- It may prefer low-active-gNB configurations even if total throughput becomes small.
- The ratio can behave badly when the denominator becomes very small.
- It is less straightforward for RL compared with an additive reward.
- It may require strong QoS constraints to prevent poor service.
- It is mathematically more difficult because it is a fractional objective.
- μ must be tuned carefully.

The optimization problem is subject to the following constraints.

C1: Binary Cell-Switching Constraint

$$z_b \in \{0,1\}, \quad \forall b \in \mathcal{B}. \tag{23}$$

This constraint ensures that each gNB is either active or switched off.

C2: Maximum Number of Switched-Off gNBs

$$\sum_{b\in\mathcal{B}}(1-z_b)\leq 1. \tag{24}$$

This constraint ensures that at most one gNB can be switched off. Equivalently, at most one UE can be offloaded to NTN.

C3: QoS Constraint

$$z_b R_{u_b,TN}+(1-z_b)R_{u_b,NTN}\geq R_{u_b}^{min},\quad \forall b\in\mathcal{B}. \tag{25}$$

Here, $R_{u_b}^{min}$ is the minimum acceptable service rate required by UE u_b. In this simplified framework, the minimum acceptable service rate is assumed to be equal for all UEs, i.e., $R_{u_b}^{min}=R^{min}$ for all $b\in\mathcal{B}$. This constraint ensures that each UE satisfies its minimum QoS requirement, regardless of whether it is served by the TN or offloaded to NTN.

Final Optimization Forms

Option 1 Final Form

$$\max_{\{z_b\}}\sum_{b\in\mathcal{B}}\left[z_b R_{u_b,TN}+(1-z_b)R_{u_b,NTN}\right]-\lambda\sum_{b\in\mathcal{B}}z_b-\mu C_{switch} \tag{26}$$

subject to C1-C3.

Option 2 Final Form

$$\max_{\{z_b\}}\frac{\sum_{b\in\mathcal{B}}\left[z_b R_{u_b,TN}+(1-z_b)R_{u_b,NTN}\right]}{\sum_{b\in\mathcal{B}}z_b}-\mu C_{switch} \tag{27}$$

subject to C1-C3.

The optimization problem formulated in Framework 1 is a discrete cell-switching and traffic-offloading problem. The decision variable z_b determines whether gNB b is active or switched off, and the same variable also determines whether UE u_b is served by the TN or offloaded to the NTN satellite.

Since at most one gNB can be switched off, the controller selects one decision from a finite set of possible switching/offloading actions. Moreover, the switching-cost term in (15) depends

on the transition between consecutive ON/OFF states, which introduces temporal dependency into the decision-making process.

Therefore, instead of solving the optimization problem independently at each time step, reinforcement learning can be used to learn a policy that maps the current network state to a switching/offloading action.

Since the action space is finite and discrete, a Deep Q-Network is used. The proposed solution is referred to as DQN 1.

The interaction between the learning agent and the TN-NTN environment is modeled as a Markov Decision Process, defined as:

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma) \tag{28}$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ represents the state-transition dynamics, $\mathcal{R}$ is the reward function, and $\gamma \in [0,1]$ is the discount factor.

At each time step t, the agent observes the state s_t, selects an action a_t, receives a reward r_t, and moves to the next state s_{t+1}.

A policy is defined as:

$$\pi(s_t) = a_t. \tag{29}$$

The objective of the reinforcement learning agent is to maximize the expected discounted return:

$$G_t = \sum_{\tau=t}^{T-1} \gamma^{\tau-t} r_\tau. \tag{30}$$

Therefore, the reinforcement learning objective is:

$$\max_{\pi} \mathbb{E}_\pi[G_t]. \tag{31}$$

In DQN 1, the state represents the current condition of the TN-NTN system, the action represents the cell-switching/offloading decision, and the reward is designed according to the scaled energy-efficiency objective in (22).

Q-learning is a value-based reinforcement learning method that learns an action-value function:

$$Q(s_t, a_t). \tag{32}$$

The action-value function represents the expected discounted return obtained by selecting action a_t in state s_t and following the learned policy afterward.

The optimal action-value function satisfies the Bellman optimality equation:

$$Q^*(s_t, a_t) = r_t + \gamma \max_{a' \in \mathcal{A}} Q^*(s_{t+1}, a'). \tag{33}$$

The optimal policy is then obtained as:

$$\pi^*(s_t) = \operatorname{argmax}_{a \in \mathcal{A}} Q^*(s_t, a). \tag{34}$$

In DQN, the action-value function is approximated by a neural network:

$$Q(s_t, a_t; \theta). \tag{35}$$

where θ denotes the trainable parameters of the online Q-network.

A target network is also used and is denoted by:

$$Q(s_t, a_t; \theta^-). \tag{36}$$

where θ^- denotes the target-network parameters.

For a transition (s_t, a_t, r_t, s_{t+1}), the target value is defined as:

$$y_t = r_t + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a'; \theta^-). \tag{37}$$

The loss function is defined as:

$$L(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[\left(y_t - Q(s_t, a_t; \theta) \right)^2 \right], \tag{38}$$

where $\mathcal{D}$ is the replay buffer.

The online network is trained by minimizing (38), and the target network is periodically updated as:

$$\theta^- \leftarrow \theta. \tag{39}$$

Experience replay is used to store previous transitions and sample mini-batches during training.

The DQN agent is a centralized controller responsible for selecting the cell-switching and traffic-offloading decision.

Because Framework 1 assumes a one-to-one mapping between gNBs and UEs, switching off gNB b automatically means that UE u_b is offloaded to the NTN satellite.

Thus, the DQN agent learns a policy that selects the ON/OFF state transition of the terrestrial gNBs while accounting for scaled energy efficiency, switching cost, and QoS satisfaction.

The state should contain compact information about the current network configuration and service quality.

According to the assumptions of Framework 1, the common QoS threshold, transmit powers, and noise powers are treated as fixed environment parameters and are therefore not included in the DQN state.

At time step t, the state is defined as:

$$s_t = \left[\mathbf{z}(t), \widetilde{EE}(t), \mathbf{\Delta R}(t)\right]. \tag{40}$$

The current ON/OFF vector is:

$$\mathbf{z}(t) = [z_1(t), z_2(t), \dots, z_B(t)]. \tag{41}$$

Using the rate formulation in (10) and (11), the achieved rate of UE u_b at time step t is:

$$R_{u_b}(t) = z_b(t)R_{u_b,TN}(t) + \big(1 - z_b(t)\big)R_{u_b,NTN}(t). \tag{42}$$

Using the throughput definition in (12), the total throughput at time step t is:

$$T_{total}(t) = \sum_{b\in\mathcal{B}} R_{u_b}(t). \tag{43}$$

Using the scaled energy-efficiency definition in (20), the current scaled energy efficiency is:

$$\widetilde{EE}(t) = \frac{T_{total}(t)}{\sum_{b\in\mathcal{B}} z_b(t)}. \tag{44}$$

The QoS margin of UE u_b is defined as:

$$\Delta R_{u_b}(t) = R_{u_b}(t) - R^{min}. \tag{45}$$

Therefore, the QoS-margin vector is:

$$\mathbf{\Delta R}(t) = \left[\Delta R_{u_1}(t), \Delta R_{u_2}(t), \dots, \Delta R_{u_B}(t)\right]. \tag{46}$$

A positive value of $\Delta R_{u_b}(t)$ means that UE u_b satisfies the minimum rate requirement, while a negative value indicates a QoS violation.

Since at most one gNB can be switched off, the action space is defined as:

$$\mathcal{A} = \{0,1,2, \dots, B\}. \tag{47}$$

The action $a_t = 0$ corresponds to keeping all terrestrial gNBs ON in the next state:

$$z_b(t+1) = 1, \quad \forall b \in \mathcal{B}. \tag{48}$$

The action $a_t = k$, where $k \in \mathcal{B}$, corresponds to switching off gNB k:

$$z_k(t+1) = 0, \tag{49}$$

$$z_b(t+1) = 1, \quad \forall b \in \mathcal{B},\ b \neq k. \tag{50}$$

Equivalently,

$$z_b(t+1) = \begin{cases} 0, & a_t = b,\ a_t \neq 0, \\ 1, & \text{otherwise.} \end{cases} \tag{51}$$

This action definition automatically satisfies the maximum switched-off gNB constraint in (24).

Therefore, DQN 1 selects one action from $B+1$ possible switching/offloading decisions instead of learning a full binary vector with 2^B possible combinations.

DQN 1 uses Objective Function Option 2 in (22) as the basis of the reward.

After action a_t is selected, the next ON/OFF vector $\mathbf{z}(t+1)$ is obtained. Using (42), the achieved rate of UE u_b after applying the action is given by:

$$R_{u_b}(t+1) = z_b(t+1)R_{u_b,TN}(t+1) + \big(1 - z_b(t+1)\big)R_{u_b,NTN}(t+1). \tag{52}$$

Using (43), the total throughput after applying the action is:

$$T_{total}(t+1) = \sum_{b\in\mathcal{B}} R_{u_b}(t+1). \tag{53}$$

Using (44), the scaled energy efficiency after applying the action is:

$$\widetilde{EE}(t+1) = \frac{T_{total}(t+1)}{\sum_{b\in\mathcal{B}} z_b(t+1)}. \tag{54}$$

The OFF-to-ON switching cost after applying the action is consistent with (15) and is written as:

$$c_{switch}(t+1) = \sum_{b\in\mathcal{B}} \big(1 - z_b(t)\big)\, z_b(t+1). \tag{55}$$

The QoS violation penalty after applying the action is:

$$c_{QoS}(t+1) = \sum_{b\in\mathcal{B}} \max\Big(0, R^{min} - R_{u_b}(t+1)\Big). \tag{56}$$

The immediate reward is defined as:

$$r_t = \widetilde{EE}(t+1) - \mu c_{switch}(t+1) - \eta c_{QoS}(t+1), \tag{57}$$

where μ is the OFF-to-ON switching-cost coefficient and η is the QoS violation penalty coefficient.

Substituting (54), (55), and (56) into (57) gives:

$$\begin{aligned} r_t = {} & \frac{\sum_{b\in\mathcal{B}} \big[z_b(t+1) R_{u_b,TN}(t+1) + \big(1 - z_b(t+1)\big) R_{u_b,NTN}(t+1)\big]}{\sum_{b\in\mathcal{B}} z_b(t+1)} \\ & - \mu \sum_{b\in\mathcal{B}} \big(1 - z_b(t)\big)\, z_b(t+1) \\ & - \eta \sum_{b\in\mathcal{B}} \max\Big(0, R^{min} - R_{u_b}(t+1)\Big). \end{aligned} \tag{58}$$

The first term rewards scaled energy efficiency, the second term penalizes OFF-to-ON transitions, and the third term penalizes QoS violations. No additional active-gNB penalty is added because the number of active gNBs is already included in the denominator of $\widetilde{EE}(t+1)$.

During training, DQN 1 uses an ϵ-greedy policy:

$$a_t = \begin{cases} \text{random action from } \mathcal{A}, & \text{with probability } \epsilon, \\ \underset{a \in \mathcal{A}}{\text{argmax}} Q(s_t, a; \theta), & \text{with probability } 1 - \epsilon. \end{cases} \quad (59)$$

The exploration probability ϵ is gradually reduced during training. After training, the learned policy is:

$$\pi^*(s_t) = \underset{a \in \mathcal{A}}{\text{argmax}} Q(s_t, a; \theta). \quad (60)$$

The final DQN 1 learning problem is:

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right], \quad (63)$$

where r_t is defined in (57), the state is defined in (40), the action space is defined in (47), and the learned policy is defined in (60).

Thus, DQN 1 learns a switching and offloading policy that maximizes the scaled energy-efficiency objective of Framework 1 while accounting for OFF-to-ON switching cost and QoS satisfaction.

4.3.2.2.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

Within the UNITY-6G architecture, the proposed algorithm runs as an xApp through the DE, physically placed on ground and connected to both the NTN and TN gNBs through the CN. The cross-domain coordination between the O-RAN domain and the NTN domain achieved through the UNITY-6G architecture ensures that 1) analytics from both domains can be used at the xApp, and 2) decisions can affect both domains concerned.

It targets KPI#16 cutting radio link power consumption by up to 30% per link, KPI #43 which aims at improving the overall energy performance by a factor of 5.

4.3.2.3 AI based energy efficient massive MIMO

The objective of the activity is to develop an algorithm to reduce the overall energy consumption of an ultra-dense Massive MIMO network without degrading the throughput experienced by users at the cell edge below a critical threshold.

To solve this nonlinear and spatially correlated optimization problem, we propose a greedy iterative algorithm guided by the network's spatial load.

Step 1: Initial Network Load Evaluation (Traffic Modeling)

The algorithm uses stochastic geometry to model the positions of base stations and users. Traffic is modeled at the flow level (not the packet level) via M/G/1/PS (Processor Sharing) queues. The algorithm calculates the ergodic load $rho_b(\rho b)$ of each base station b using fixed-point iterations. The SINR calculation incorporates pathloss, large-scale fading (shadowing), and interference suppression specific to Massive MIMO (MRT spatial filtering). Interference generated by the rest of the infinite network is approximated using Campbell's theorem.

Step 2: Energy Quantification

The power consumption of each base station is evaluated using a detailed power model:

$$P_{BS} = P_{fixe} + M.P_{ant} + \eta^{Ptx}.\rho + P_{DSP\,(M,K)}$$

Where P_{fixe} is the base power consumption, M is the number of active antennas, P_{ant} is the power consumption per RF chain, η is the amplifier efficiency, Ptx is the transmitted radio frequency power, M is the number of active antennas, K is the number of active users, $P_{DSP\,(M,K)}$ is the baseband signal processing power, and ρ is the station load.

Step 3: Identification and Pruning

At each iteration, the algorithm follows the following heuristic logic:

- Calculation of the Cell-Edge Throughput for the 95th percentile of users. If this throughput falls below the QoS constraint, the algorithm stops.
- Identification of the target base station with the lowest spatial load ρb among those still operating at their maximum configuration.
- Reduction of the MIMO configuration of this station (e.g., dividing the number of antennas by 4, from 1024 to 256).
- Update of the inter-cell interference matrices and return to Step 1.

This loop allows us to approach the highest EE and SE of the network.

4.3.2.3.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed algorithm for 5G energy efficiency will be implemented to meet the objectives outlined in KPI#43. This KPI aims to significantly improve the overall energy performance of the network, with a specific target to increase the integrated network energy efficiency by a factor of 5. Achieving such a substantial enhancement requires innovative strategies and advanced optimization techniques.

To this end, we propose a proactive resource allocation strategy within the 5G RAN, which leverages antenna traffic prediction models. By accurately forecasting traffic patterns, the system can dynamically adjust network resources, deactivate unnecessary components during low traffic periods, and optimize power consumption without compromising service quality. This approach not only enhances energy efficiency but also contributes to the sustainability goals of the network.

The results presented in our initial tests show a reduction in energy consumption from 52 kW to 48 kW. While this represents a positive step forward, it does not yet achieve the targeted factor of 5 in energy efficiency. Nevertheless, these preliminary findings validate the potential of our approach. We plan to further refine our algorithms and conduct simulations in more densely populated and complex scenarios, where the benefits of proactive resource management are expected to be even more pronounced.

In addition to optimizing the RAN, we are also focusing on reducing energy consumption within the 5G core network, which accounts for a significant portion of overall energy use. By applying similar predictive and adaptive techniques, we aim to create a holistic energy-efficient architecture across the entire network.

This algorithm will be integrated into the AI/ML subsystem models, enabling automated and intelligent energy optimization. The integration will facilitate continuous learning and adaptation, ensuring that the network maintains optimal energy performance as traffic patterns and network conditions evolve over time. Ultimately, our goal is to contribute to a greener, more sustainable 5G infrastructure that balances high performance with minimal environmental impact.

4.3.2.3.2 Preliminary results

To validate the algorithm, a complete system simulator was developed using MATLAB. The scenario models a macro-cell area with an inter-site distance (ISD) of 450 m. The initial MIMO configuration of each base station is extreme, set at 1024 antennas (32×32). The channel model incorporates the channel hardening characteristic of Massive MIMO, validated by Monte Carlo simulations, and out-of-area interference noise is simulated using the Campbell model.

The strict acceptability constraint is maintaining a Cell-Edge Throughput greater than 5 Mbps for 95% of users as shown in Figure 55. The power dissipated per RF chain is set at 1 W, making the 1024 antennas highly power-intensive (approximately 1100 W to 1300 W per BS depending on its load).

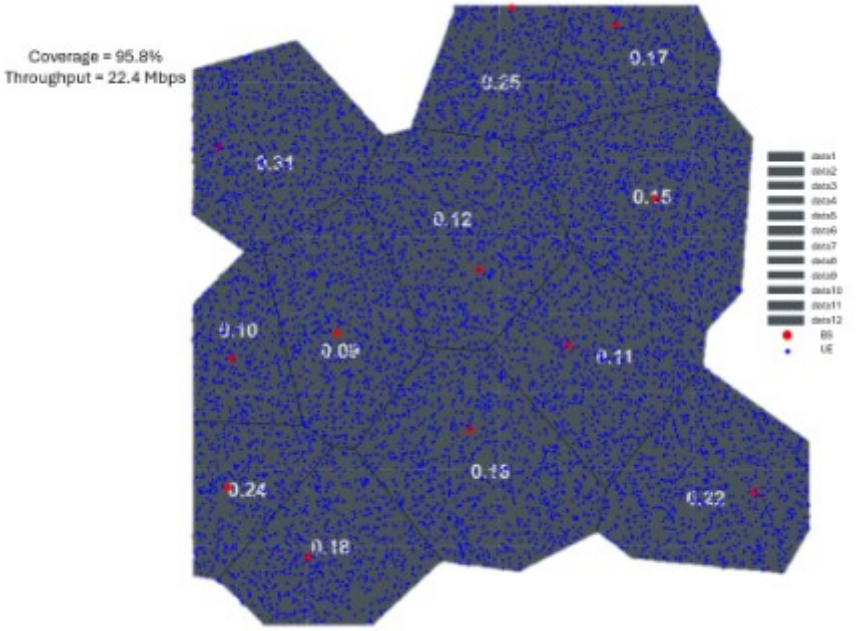


Figure 55: The system simulator for Massive MIMO.

We evaluated an iterative pruning sequence, where at each step, the least loaded base station has its antennas reduced from 1024 to 256.

Energy savings

The initial network, composed of approximately fifty base stations, has a total power consumption of 52.34 kW. In just 5 iterations of targeted pruning, the overall network power consumption drops almost linearly to 48.26 kW. This reduction of more than 4 kW demonstrates the significant impact of disabling redundant RF chains on underloaded cells.

The goal of this contribution is to move beyond simulation and validate the algorithm on a real cluster of existing base stations. Using an LLM model designed for time series prediction (Chronos), it is possible to input a predicted throughput value into the simulator and adjust the MIMO configuration to achieve the desired throughput.

4.3.2.4 Joint beam-user allocation strategy

The objective of this activity is to formulate a joint beam-user allocation as an online contextual combinatorial learning problem. The aim is to preserve the practicality of heuristic allocation while adding a learning mechanism that explicitly considers throughput, energy efficiency, QoS satisfaction, fairness, and beam switching cost. The resulting approach is intended to reduce unnecessary energy consumption and unstable beam switching without treating throughput as the only optimisation target.

The project proposes a Combinatorial Multi-Armed Bandit framework based on Linear Thompson Sampling, referred to as CMAB-LinTS in cell free network. The method treats each

possible beam-user pair as an arm. At each simulation step, the algorithm selects a super-arm, meaning a feasible set of beam-user assignments that jointly determines how users are served by the available beams.

The key idea of the proposed approach is to replace purely myopic beam selection with an adaptive online decision mechanism. Instead of selecting beam-user assignments only according to instantaneous SINR or immediate throughput gain, CMAB-LinTS estimates the expected value of each candidate assignment using contextual information and feedback collected during network operation. The model learns online and therefore does not require offline pre-training or a fixed traffic model.

In the proposed formulation, each possible beam-user pair is treated as an arm. At each time step t, the algorithm selects a feasible set of assignments, called a super-arm:

$$S(t) \subseteq M \times U(t)$$

where M is the set of available beams and U(t) is the set of active users. This formulation is suitable for radio resource allocation because the network does not choose only one action, but a combination of beam-user assignments that must jointly satisfy capacity, bandwidth, SINR, and MIMO-layer constraints.

Each candidate beam-user pair (m,u) is described by a contextual feature vector:

$$x_{(m,u)}(t) \in R^d$$

The context includes radio conditions, geometry, load, mobility-related information, MIMO state, persistence flags, previous allocation state, QoS deficit, and recent performance statistics. This allows the algorithm to distinguish between assignments that may have similar instantaneous SINR but different energy cost, stability, or QoS impact.

Linear Thompson Sampling is used as the learning mechanism. The expected reward of a candidate assignment is modelled as a linear function of its context:

$$E\left[r_{(m,u)}(t)\middle|x_{(m,u)}(t)\right] = x_{(m,u)}^{T}(t)\theta$$

Where θ is an unknown parameter vector learned during operation. At each decision step, the algorithm samples a plausible parameter vector from the posterior distribution:

$$\theta\widetilde{(t)} \sim N\left(\theta\widehat{(t)}, \alpha^2 A^{(-1)}(t)\right)$$

The predicted value of a beam-user pair is then computed with an additional pessimism correction:

$$\widehat{r_{m,u}}(t) = x_{m,u}^T(t)\tilde{\theta}(t) - k\sqrt{x_{m,u}^T(t)A^{-1}(t)x_{m,u}(t)}, k = 0.3$$

This provides a natural balance between exploitation and exploration. Assignments that are currently estimated as beneficial are likely to be selected, while uncertain assignments can still be explored when the available evidence is limited.

The proposed method is designed for energy-aware beam management, so the reward cannot be based on throughput alone. A throughput-only objective may activate additional resources even when the rate gain is marginal, while an energy-only objective may reduce power consumption at the cost of unacceptable QoS degradation. Therefore, the project uses a multi-objective allocation score:

$$s_{(m,u)}(t) = 0.35\Delta_p red(t) + 0.65\Delta U_t hr(t) - \beta\Delta E(t) - \gamma \ln(1 + c_s w(t)/s) - \delta\Delta_Q oS(t) + b_u nserved(t)$$

The score combines the predicted utility from the online learning model, incremental throughput utility, incremental energy cost, beam switching or handover cost, QoS shortfall penalty, and support for previously unserved users. The throughput component is based on the achievable user rate,

$$R_u(t) = W_u(t)\log_2\left(1 + SINR_u(t)\right)$$

but the final decision also considers whether the additional throughput justifies the energy and stability cost.

This formulation allows the algorithm to search for a balanced operating point. CMAB-LinTS can prefer an assignment with slightly lower instantaneous throughput if it reduces energy consumption, avoids unstable beam switching, or improves QoS satisfaction for users that would otherwise remain underserved.

The energy component uses an incremental activation-cost model:

$$\Delta E(t) = 1[beamnew] \cdot P_{(beam,static)} + P_{(beam,dyn)} + 1[BSnew] \cdot P_{(BS,static)}$$

Where $P_{(beam,static)}$, $P_{(beam,dyn)}$, and $P_{(BS,static)}$ are configurable per-activation cost coefficients. In the current setup, the default values are 2 W, 0.2 W, and 0.5 W. The model captures the incremental cost of activating a new beam or waking a sleeping base station, without requiring full knowledge of the transceiver-chain parameters.

As a result, the algorithm is encouraged not only to increase throughput, but also to improve the ratio between delivered data rate and consumed network power. This makes the proposed

CMAB-LinTS approach suitable for proactive, AI-driven, energy-aware beam-user allocation in 5G and future 6G networks.

4.3.2.4.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

The proposed algorithm fulfils the objective of KPI #43 in the area of AI-driven energy efficiency. Although the current results do not yet demonstrate a 5× improvement, CMAB-LinTS already shows a clear positive impact by reducing energy consumption and improving energy efficiency through proactive, energy-aware beam-user allocation. The method learns from network conditions and balances throughput, QoS, handovers, and energy cost, which directly supports the KPI direction. With further optimisation, such as improved reward tuning, sleep-mode mechanisms, traffic consolidation, and broader network-level control, the achieved energy-efficiency gain could likely be increased further.

4.3.2.4.2 Preliminary results

The proposed CMAB-LinTS allocator was evaluated in a downlink 5G NR simulation environment and compared with a greedy heuristic baseline. The simulation considers three deployment scenarios: urban, suburban, and rural. These scenarios differ in propagation conditions, blockage, user distribution, and base station layout. The purpose of this evaluation was to verify whether the proposed learning-based allocator can improve energy-aware beam-user allocation without causing a major loss in throughput or QoS.

The main simulation parameters are summarised in Table 10.

Table 10: Parameters of the simulation

Parameter	Value
Base stations B	3
URA size	8×8 elements
Carrier frequency f_c	3.5GHz
Element spacing	0.5λ
Sector width	120°
Beams/sector	30
Max users/beam	15
System bandwidth	120MHz
Min. BW/user	5MHz
MIMO layers L	6 (max 3/BS)
TX power	35dBm
BS/UE height	35m / 1.5m
Blockage loss/wall	5dB
$n_{\text{trx}}, P_0, \Delta, P_{\text{out,max}}$	2, 130W, 4.7, 20W
P_{sleep}	20W
Simulation horizon	500 steps
Number of users N	20, 30, 40, 50, 60
Runs/configuration	30

The number of users was varied from 20 to 60 in order to test the algorithms under different load conditions. For each environment and user density, both the greedy heuristic and CMAB-LinTS were evaluated using identical trajectories. The comparison included mean throughput,

Jain fairness index, QoS satisfaction, total energy used, energy efficiency, and the average number of handovers per user.

The preliminary results (**Error! Reference source not found.**) show that CMAB-LinTS improves the mean throughput in almost all tested configurations. The improvement is visible across rural, suburban, and urban environments. In the rural scenario, the gain is particularly strong for low and medium user densities, especially for 20, 30, and 40 users. In suburban and urban scenarios, the improvement is more moderate but remains consistent across most user densities. This suggests that the contextual learning model is able to identify beam-user assignments that provide better system-level utility than the purely myopic heuristic baseline.

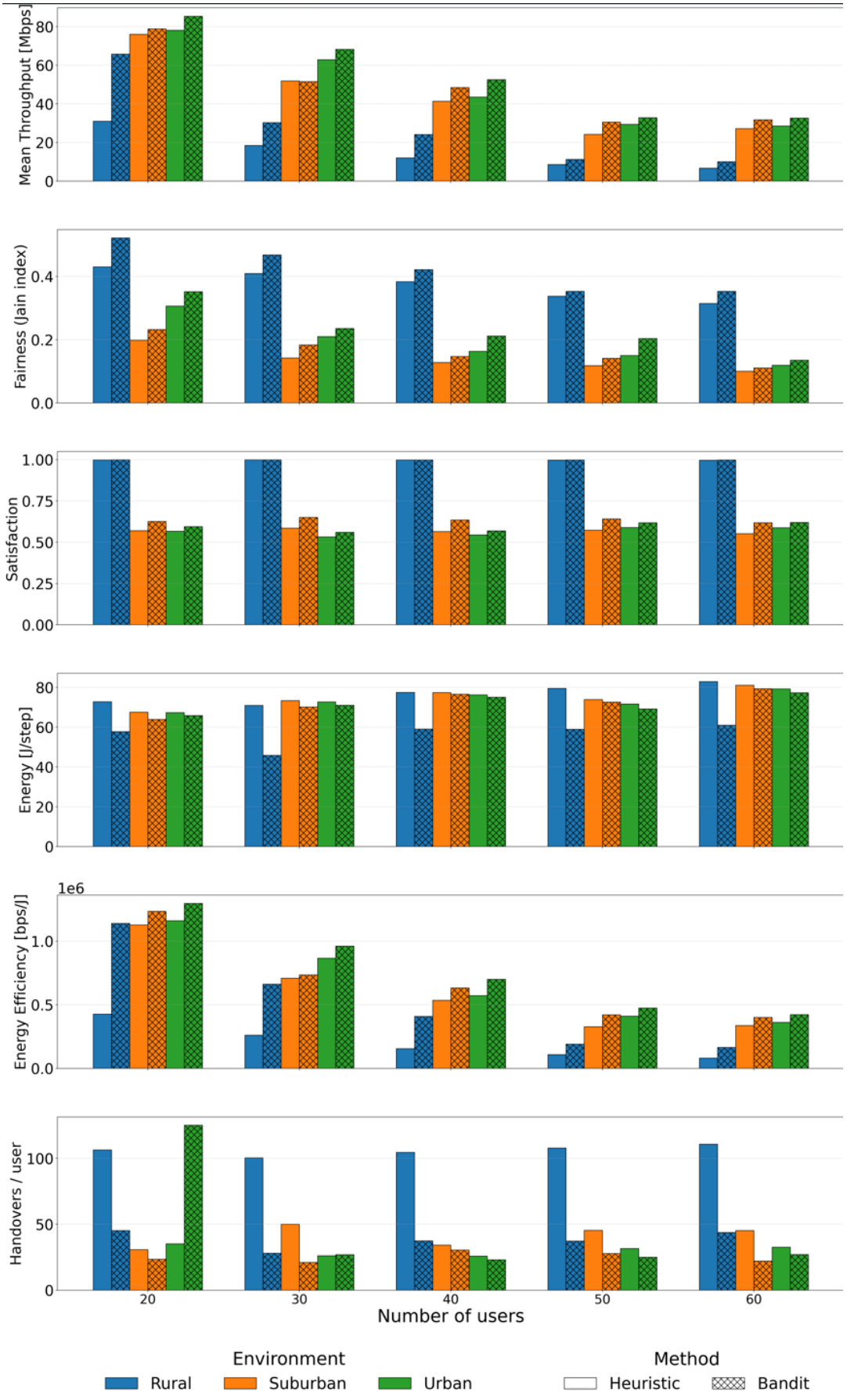


Figure 56: Results of the simulation

Fairness results are also generally favourable for the proposed method. The Jain fairness index achieved by CMAB-LinTS is usually higher than the value obtained by the heuristic allocator. This indicates that the learning-based approach does not improve throughput only by favouring already well-served users. Instead, the QoS repair mechanism and the bonus for previously unserved users help distribute service evenly among users.

QoS satisfaction follows a similar trend. In the rural scenario, both methods already achieve very high satisfaction, close to the maximum value, which limits the p ossible improvement. In suburban and urban scenarios, CMAB-LinTS provides a small but consistent increase in satisfaction. This shows that the proposed reward formulation helps the algorithm maintain service quality while simultaneously optimising other objectives such as energy consumption and allocation stability.

The energy results confirm the energy-aware nature of the proposed approach. CMAB-LinTS reduces total energy used in all scenarios, with the largest reduction observed in the rural environment. In suburban and urban cases, the reduction is smaller but still visible. This demonstrates that the load-dependent energy term in the reward function successfully influences the allocation decisions and prevents the algorithm from activating resources when the additional throughput gain is not justified by the energy cost.

The improvement in energy efficiency is one of the clearest results. Across all environments and user densities, CMAB-LinTS achieves higher energy efficiency than the heuristic baseline. This means that the proposed method is able to deliver more useful throughput per unit of consumed energy. The result is especially important because it shows that the algorithm does not simply reduce energy by lowering service quality, but improves the trade-off between throughput and power consumption.

The handover results are more mixed. In most rural configurations, CMAB-LinTS significantly reduces the number of handovers per user compared with the heuristic method. This confirms the effect of the switching-cost term included in the reward function. However, in some urban low-load configurations, especially for 20 users, the bandit-based method produces more handovers than the heuristic baseline. This suggests that the exploration mechanism can sometimes increase allocation changes when the model is still learning or when several beams provide similar utility. Therefore, the switching penalty should be further tuned in future work to avoid unstable behaviour in selected scenarios.

Overall, the preliminary results demonstrate that CMAB-LinTS is a promising method for energy-aware beam-user allocation. Compared with the greedy heuristic baseline, the

proposed algorithm achieves higher throughput, better or comparable fairness, improved QoS satisfaction, lower energy consumption, and substantially higher energy efficiency.

4.3.2.5 AI-empowered multivariate probabilistic forecasting a key enabler for sustainability in open RAN

We consider a scenario where different O-RUs serve a geographical area through two types of base stations: *capacity* and *coverage* base stations. These base stations are responsible for serving users by allocating PRBs to handle the aggregated data traffic in the O-RAN network. The capacity base station is designed to manage high data traffic in areas with high user density. It usually has fewer PRBs than a coverage base station and is mainly used to offload traffic from the coverage base station during peak demand periods. These demand peaks often exhibit seasonal patterns depending on the time of day or day of the week. In contrast, the coverage base station is designed to provide wider network coverage and has a larger PRB capacity, enabling it to serve a higher volume of traffic. This ensures continuous service availability when the capacity base station is switched off. Figure 57 below illustrates the considered scenario. The objective is to predict the future PRB demand at both the capacity and coverage base stations, as well as the total demand when all PRBs (i.e., capacity + coverage traffic) are handled solely by the coverage base station. These predictions help determine the optimal moments to switch off the capacity base station while ensuring that the coverage base station can accommodate the entire traffic load without exceeding its available PRBs. Such a strategy is essential for improving energy efficiency and optimizing resource utilization by exploiting seasonal traffic patterns over time.

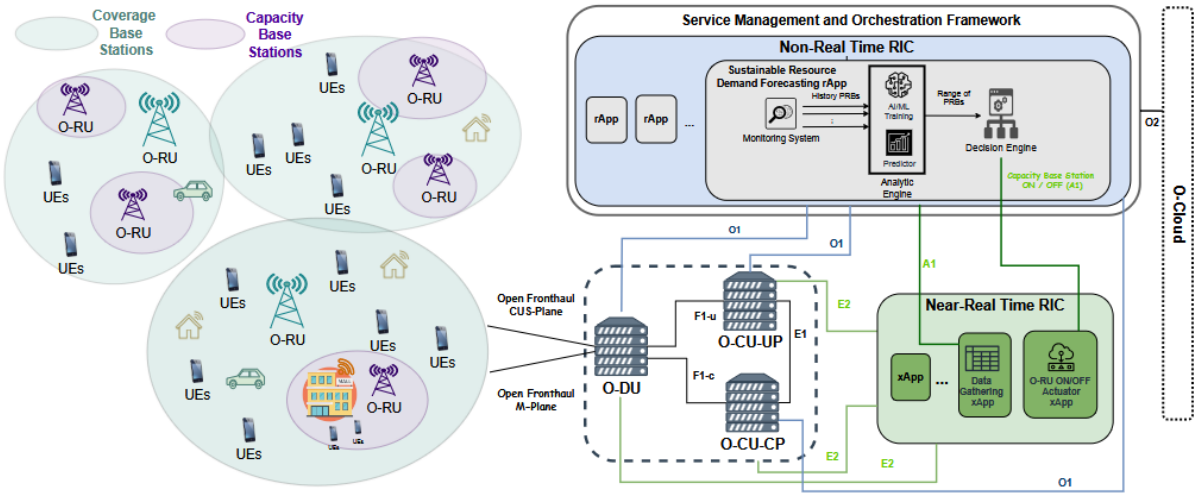


Figure 57: Scenario description for the probabilistic-aware energy saving algorithm.

The role of multivariate probabilistic forecasting in improving the energy efficiency and sustainability of O-RAN operations is explored next. It presents an overview of the O-RAN architecture and power consumption models, emphasizing the limitations of traditional

deterministic approaches for resource allocation and energy management. The deliverable further investigates and compares several multivariate forecasting techniques, presented in Section 3.3.3.3, including GPVAR [132], TFT [133], and LSTM [134], for predicting PRB utilization and enabling more sustainable and energy-aware resource optimization in O-RAN networks.

Formally, the multivariate PRB allocation forecasting problem is represented as a time series $\boldsymbol{y}_t \in \mathbb{R}^N$at time t. In our specific use case, without loss of generality and for the sake of simplicity, we considered one capacity, one coverage O-RUs collocated under the umbrella of the same RIC. Thus, we set *N*, which is total number of time series to N=*3* for each of the series, i.e., coverage, capacity, and total summation of requested radio resources, i.e. coverage + capacity. The objective is to model the joint conditional distribution of future PRB allocation values based on previously observed data, denoted by $P(\boldsymbol{y}_{t_0:T}|\boldsymbol{y}_{1:t_0-1})$, where $\boldsymbol{y}_{t_0:T} = [\boldsymbol{y}_{t_0:T}, \cdots, \boldsymbol{y}_{t_0+T}]$ denotes the future PRB allocation, given its past values $[\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{t_0-1}] := \boldsymbol{y}_{1:t_0-1}$. Here t_0 denotes the time point from which we assume y_t to be unknown. Once we learn to forecast and quantify the uncertainty in the prediction range, we provide a sustainability analysis using the power consumption model presented below.

Power consumption model

Consider the power consumption model for 5G BS recently introduced in [129]. The total power consumption, denoted by P_T , can be mathematically characterized as follows:

$$P_T = P_0 + P_{BB} + P_{Tran} + P_{PA} + P_{out},$$

where P0 represents the baseline power consumption in sleep mode, P_{BB} denotes the baseband processing power consumption, P_{Tran} represents the power consumption by the RF chains, P_{PA} indicates the static consumption of the power amplifier and Pout corresponds to the power required for data transmission. The initial terms, i.e., $P_{0,}P_{BB}, P_{Tran}$ are dependent on the number of available and active RF chains. For the sake of simplicity and without a loss of generality, herein we assume one transceiver. Similarly, the term P_{PA} is given by the product of the number of RF chains and the static power consumed by each multi-carrier power amplifier. The last term is proportional to the transmitted traffic volume:

$$P_{out} = \frac{1}{\eta} P_{TX} \frac{R_{BS}}{C_{BS}},$$

where η denote the efficiency of the power amplifiers, and P_{TX} represent the maximum transmit power. The symbols R_{BS} and C_{BS} denote the actual rate and the capacity of the base station, respectively. A surrogate for $\frac{R_{BS}}{C_{BS}}$ quotient is the DL PRB load,defined as the ratio between the

average number of DL PRBs and the maximum number of DL PRBs available. Recent studies in [135][136] have demonstrated that the DL PRB load is the most significant feature in modeling radio unit power consumption. Additionally, it is important to note that the transmit power increases linearly with the number of used PRBs [129]. Consequently, there is a trade-off between meeting the DL PRB resource demands and minimizing energy consumption.

Proposed power saving algorithm

This section focuses on achieving power savings in O-RAN using advanced probabilistic forecasting models such as GPVAR and TFT as well as multivariate LSTM, which serve as a baseline model for benchmarking. It is assumed that the estimators are integrated into the O-RAN RIC controller. Effective and dynamic resource allocation is important for efficient power consumption without wasting resources. The accurately predicted PRBs in turn help in designing an efficient strategy to switch off the capacity base station depending on the resource utilization. **The two significant conditions** to be satisfied for switching off the capacity base station are:

1. Capacity base stations forecast PRB demand at a particular time should be less than a predefined threshold.

2. The total PRB demand (PRB required for capacity + coverage base station) predicted at a specific time instance should not be above the maximum PRBs that a coverage station can offer without any bottlenecks. In another words, the coverage cell must be able to serve the requested PRBs with a high probability in the next hours if the capacity BS is switched off.

Effective power usage is ensured through condition one by switching off the underutilized capacity base station while the service quality degradation is prevented by condition 2 through ensuring that the coverage base station can handle the combined load requirements.

The proposed solution in this section starts with the gathering of historical data on allocated PRBs from different base stations in xApp via O-DU. The gathered historical data is shared with resource forecasting rApp for further processing. Next, the analytical engine in rApp trains the forecasting estimators using the input PRB allocation information, followed by the prediction of future PRB demands using the trained models. The decision engine uses these predictions to check whether conditions 1 and 2 are met. The fulfilment of both conditions guarantees that the capacity base station can be switched off and that the coverage station can handle all service requests from users in that particular area without overhead. If this is not the case, the capacity station is active and the total power consumption is higher, which results in no or low power savings. To calculate the power consumption and the corresponding

power savings, the parameters of the power consumption model explained above are initialized with values as explained in the reference [129]. Finally, the decision is implemented by the actuator xApp within the O-RAN framework. The process includes continuous monitoring and decision adjustment for efficient network operation and increased energy savings. This approach helps to achieve a better balance between resource utilisation and power efficiency, which is important for a sustainable O-RAN

4.3.2.5.1 Mapping of the algorithm onto the UNITY-6G architecture and addressed KPIs

Aligned with the UNITY-6G architecture, the proposed AI-enabled Multivariate Probabilistic Forecasting is a Key Enabler for Sustainable Open RAN. By replacing traditional deterministic single-point models with multivariate probabilistic forecasting architectures (e.g. GPVAR and TFTs), we presented solid foundations to advance towards energy-efficient networks.

Next, we show how the proposed solution contributes to each KPI:

1) KPI #43: Increase integrated network energy efficiency**.** The proposed technique integrates multivariate probabilistic models directly into the Open RAN ecosystem (specifically, via the Non-Real time RIC). Because these architectures forecast a range of possible demand percentiles for the multi-step ahead timesteps (e.g., PRB utilization) rather than a single-point guess, the RIC can proactively scale resources up or down without risking service degradation. Traditional scheduling over-provisions compute resources to deal with peaks. By accounting for spatial and temporal multivariate interdependencies, the proposed framework allows the network to plan energy-aware scheduling for AI workloads (training/inference) and offload tasks to periods or edge nodes with high renewable energy availability.

2) KPI #15: Save up to 25% of energy costs by reducing CNF energy consumption. CNFs consume considerable computational energy when idle or over-allocated. The paper's simulation results explicitly highlight that energy savings of around 20–30% are achieved by tuning resource allocation to specific confidence percentiles of the probabilistic forecasts. By dynamically adjusting CPU/memory provisioning for CNFs based on exact probabilistic traffic distributions rather than worst-case peak estimates, the network drops its baseline idle energy footprint.

3) KPI #16: Reduce radio link power consumption up to 30% per link on average across a transport path (X-haul). Dynamic Active Cell Management: Because the RAN accounts for roughly 73% of overall mobile network energy, the paper focuses heavily on optimizing PRB utilization. Armed with high-accuracy, uncertainty-aware forecasts, the system can dynamically put capacity-serving layers/links into deep sleep modes during low-traffic percentiles and

balance traffic across remaining X-haul links. This optimization yields the exact 20–30% reduction in per-link power consumption specified in this KPI.

4) KPI #17: Reduce CAPEX and OPEX with shared resources**.** By shifting network management from reactive troubleshooting to zero-touch proactive optimization (driven by the Near-RT and Non-RT RIC), operational costs (OPEX) tied to energy bills and manual network tuning drop drastically.

When infrastructure capacity is managed using deterministic margins, operators must buy and deploy extra hardware (over-provisioning) to guarantee reliability during unpredicted spikes. The proposed solution shows that understanding uncertainty structures allows operators to safely maximize the utilization of shared cloud-native pools. This extends the lifespan and capacity of existing infrastructure, delaying the need for capital-intensive expansion (CAPEX).

4.3.2.5.2 Preliminary results

This section presents an analysis of the performance of deterministic and probabilistic multivariate forecast models. Python is the primary programming language used to develop the estimators within the rApp framework. The DL requested PRBs for the next 48 hours are predicted using various estimators, namely multivariate LSTM, GPVAR, and TFT. The data is split into a 70:30 ratio, where training and validation are performed using 70 percent of available PRB data, and the remaining 30 percent is used for model evaluation. In this study, approximately one month of PRB data was used for model training, which in turn aided in capturing realistic temporal variations in network load. The DL PRB history data can be obtained from O-DU via the O1 interface in O-RAN. The dataset used in this work was generated by simulating traffic and mobility patterns for different numbers of end users in a realistic urban environment. The authors considered a city-wide RAN scenario for traffic simulation and recorded user behaviour. For this work, PRB data from only two base stations was considered: a coverage base station and a capacity base station.

The multivariate LSTM model is designed using the Keras library. The LSTM network is trained for 41 epochs with 3 dense layers, 10 hidden layers, 0.5 dropout rate, Adam optimizer, and L2 regularization. The learning rate is adjusted based on validation loss.

The probabilistic estimators GPVAR and TFT are modeled using the PyTorch library. The parameters used for these estimators include batch size, maximum epochs, prediction length, and hidden size, which are set to 128, 50, 48 hours, and 100, respectively. The context length is 120 hours (5 days). To improve the understanding of temporal patterns, additional time-related variables such as hour of day, weekday, and weekend indicators are included as time-varying known inputs when creating data loaders. In the case of GPVAR, two RNN layers are

used with a multivariate normal distribution loss function, Adam optimizer, and learning rate of 0.01. TFT models are trained with a learning rate of 0.03 using quantile loss and the Ranger optimizer. The optimal learning rate for both models is determined using a tuner module. Overfitting is mitigated by tuning the learning rate and applying early stopping.

In this study, it is assumed that two types of base stations are available, namely coverage and capacity base stations, which serve as the first two input time series for the analysis. Total PRB demand, defined as the sum of actual PRB demand from coverage and capacity cells, forms the third time series. Accurately predicting and analyzing required PRBs ensures that user quality-of-service requirements are met through efficient network operation. The maximum number of PRBs available at the coverage base station is 140 PRBs.

Sustainability Analysis Assessment

This section focuses on the sustainability aspects of probabilistic forecasting using the O-RAN framework. The main objective of this analysis is to calculate the achievable power savings with different probabilistic models based on the estimated PRB requirements for a single-carrier use case with 64 RF chains. Performing such an in-depth analysis is crucial for an energy-efficient and sustainable network. As described in [129], different normalized parameters are used to calculate power consumption. The baseline power consumption in sleep mode, denoted as P_0, is set to 0.22, and the maximum transmit power P_{TX} is set to 0.299. The power consumption of the baseband processing P_{BB} , the RF chains P_{Tran}, the power amplifiers P_{PA} and the efficiency of the power amplifiers η are set to 0.16, 0.09408, 0.24384, and 0.4, respectively. All these values are normalized. The reduction in power consumption and the resulting power savings can be efficiently quantified based on these parameters, contributing to an accurate assessment of sustainability.

A crucial aspect of the sustainability analysis is the calculation of power savings resulting from switching off the capacity base station depending on the predicted PRB requirements. The previous section, *Multivariate PRB Forecasting Assessment*, demonstrated the benefits of using probabilistic forecasting models to estimate PRB requirements accurately through their effective understanding of historical traffic data and current network conditions. These capabilities are critical for optimizing power consumption, as they help determine when a low-traffic base station can be deactivated to achieve greater power savings without compromising quality of service.

The capacity base station is switched off when the following two conditions are met:

- The predicted PRB demand of the capacity base station falls below a selected threshold value ranging from 10 PRBs to 55 PRBs.

- The predicted total PRB demand remains below the maximum capacity of the coverage base station, which is 140 PRBs.

If both conditions are satisfied, the coverage base station alone can handle the traffic requirements without bottlenecks or excessive overhead when the capacity base station is switched off. Furthermore, selecting the highest threshold of 55 PRBs (approximately 80% of the maximum capacity of the capacity base station) provides a safety margin that accounts for dynamic traffic conditions and helps prevent service degradation caused by sudden traffic spikes.

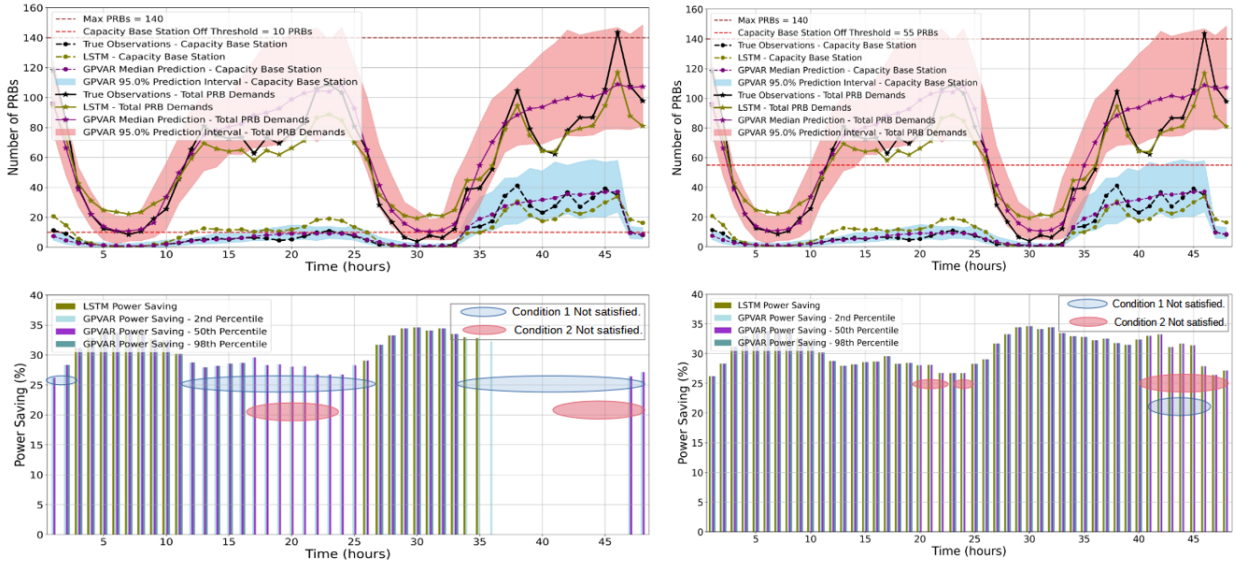


Figure 58: Power Saving with GPVAR Estimator. Capacity Base Station OFF Threshold = 10 PRBs (left) and 55 PRBs (right).

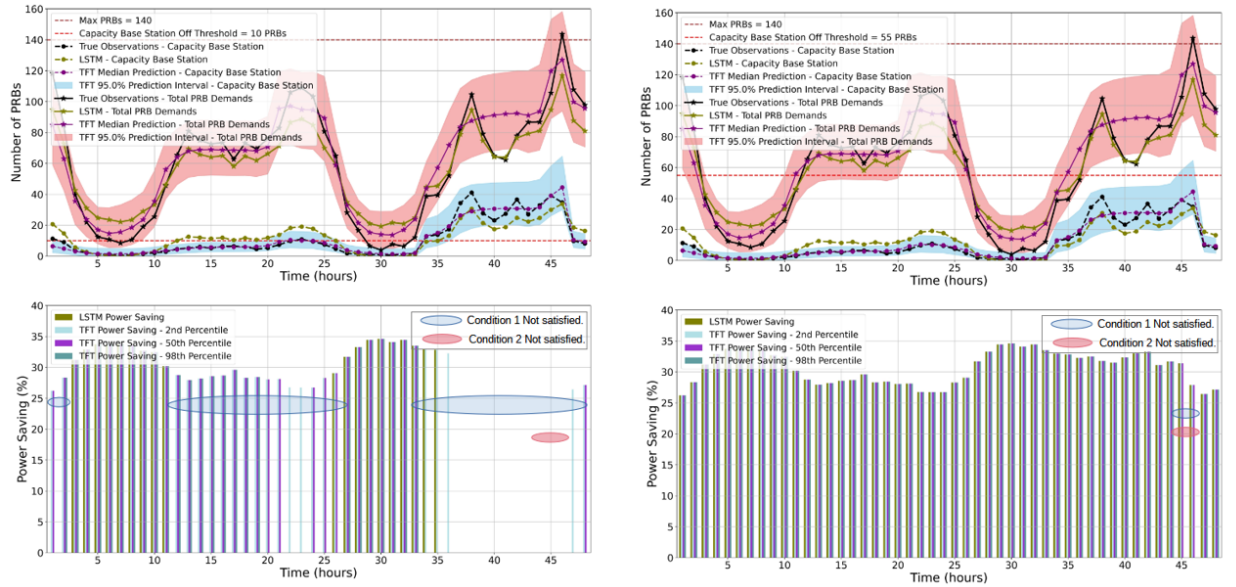


Figure 59: Power Saving with TFT Estimator. Capacity Base Station OFF Threshold = 10 PRBs (left) and 55 PRBs (right).

Figure 58 and Figure 59 present the PRB demands and corresponding power-saving percentages over a 48-hour prediction period for different forecasting models (GPVAR and

TFT). The top subplot illustrates the actual and predicted values of total PRBs and the capacity base station PRB requirements. The maximum capacity of the coverage base station is indicated by a brown dashed line at 140 PRBs. The red dashed line represents the capacity base station OFF threshold, set at either 10 PRBs (left subfigure) or 55 PRBs (right subfigure) depending on the figure. The actual total PRB demand, which combines the demands of both the capacity and coverage base stations, is shown as a black line with star markers. LSTM predictions are shown as an olive-green line with star markers. The 95% prediction interval produced by probabilistic estimators is represented by a coral-red shaded region, while the median prediction is shown as a purple star-marked line.

Similarly, the actual PRB demand of the capacity base station is represented by a black line with circle markers. LSTM predictions are displayed as an olive-green line with circle markers, while probabilistic model predictions are represented by a blue-shaded 95% prediction interval and a purple circle-marked median prediction. The lower subplot in each figure shows the power savings achieved using different forecasting models. LSTM results are shown using olive-green bars, while probabilistic model results at the 2nd, 50th, and 98th percentiles are represented by sky-blue, purple, and steel-blue bars, respectively.

Power savings are zero whenever the predicted PRB requirement exceeds the capacity base station OFF threshold or when the predicted total PRB demand exceeds 140 PRBs, making it impossible to switch off the capacity base station. Blue and magenta elliptical markers indicate violations of Conditions 1 and 2, respectively.

Figure 58 (left) presents the results for the GPVAR estimator with a 10-PRB OFF threshold. The capacity base station remains active most of the time because the required conditions are frequently not met. Between hours 17–24 and 41–48, the predicted total PRB demand at the 98th percentile exceeds 140 PRBs, causing Condition 2 to fail. As a result, the capacity station must remain active because the coverage station alone cannot handle the traffic demand. During periods highlighted by blue ellipses, such as hours 1–2, 12–25, and 34–48, Condition 1 is not satisfied, resulting in zero power savings.

Figure 58 (right) the OFF threshold is increased from 10 PRBs to 55 PRBs. A higher threshold allows the capacity base station to be switched off more frequently. Consequently, the blue ellipse regions are smaller because Condition 1 is satisfied more often. However, GPVAR still struggles to accurately capture data calibration and trends, causing Condition 2 to be violated at the 98th percentile. Selecting an appropriate OFF threshold requires balancing quality of service during high-traffic periods with maximizing energy savings without compromising network performance.

The left subfigure in Figure 59 shows the power savings and PRB demand predictions using the TFT estimator with an OFF threshold of 10 PRBs. The lower threshold means that the capacity base station is switched off only when the predicted demand remains below the assigned threshold.

The right subfigure in Figure 59 presents the TFT results with a higher OFF threshold of 55 PRBs. Compared with GPVAR and TFT using the lower threshold, the capacity base station is switched off much more frequently, resulting in the highest energy savings. Condition 2 is satisfied most of the time because TFT predictions more closely follow the actual traffic trends than GPVAR predictions. Except for hours 45 and 46, Condition 2 is satisfied throughout the prediction horizon regardless of the selected OFF threshold.

The influence of both percentile selection and OFF threshold values can be further understood through the results presented in Table 11.

Table 11: Comparison of the forecasting models in terms of Energy Saving

Forecast Models		**Percentiles**					
		2-nd	**25-th**	**50-th**	**75-th**	**90-th**	**98-th**
		Total PRB Demands Overprovisioning %					
LSTM		41.66					
GPVAR		2.08	37.5	64.58	91.66	91.66	100
TFT		10.41	37.5	60.57	81.25	89.58	97.91
		Total PRB Demands Underprovisioning %					
LSTM		58.33					
GPVAR		97.91	62.5	35.41	8.33	8.33	0
TFT		89.58	62.5	39.43	18.75	10.41	2.08
Capacity Base Station OFF Threshold	**Forecast Models**	**Energy Saving %**					
10 PRBs	**LSTM**	11.255					
	GPVAR	23.01	21.70	21.05	15.48	14.26	13.02
	TFT	23.01	21.70	19.16	16.68	16.68	14.83
15 PRBs	**LSTM**	19.23					
	GPVAR	26.05	22.34	21.70	21.05	21.05	17.33
	TFT	23.77	22.34	21.70	21.05	21.05	18.54
20 PRBs	**LSTM**	23.79					
	GPVAR	30.50	23.01	22.34	21.70	21.70	17.33
	TFT	27.49	23.01	23.01	22.34	21.05	21.05
35 PRBs	**LSTM**	30.58					
	GPVAR	30.58	30.58	27.44	24.57	23.01	18.62
	TFT	30.58	29.75	28.96	23.77	23.01	23.01
55 PRBs	**LSTM**	30.58					
	GPVAR	30.58	30.58	30.58	30.58	30.58	22.27
	TFT	30.58	30.58	30.58	30.58	29.75	28.96

Table 11 presents the over-provisioning and under-provisioning percentages (i.e., operational network metrics) of different forecasting models for predicted total PRB demand, along with the corresponding energy savings achieved at various OFF thresholds and prediction percentiles.

The over- and under-estimation percentages are independent of the selected capacity station OFF threshold. Lower percentiles generally achieve higher average hourly energy savings

across all forecasting models, but this comes at the expense of significantly higher under-provisioning rates, which can negatively affect quality of service. Increasing the base station OFF threshold generally leads to greater energy savings. However, the threshold must be selected carefully to ensure that the coverage base station does not become overloaded.

Overall, probabilistic forecasting models such as TFT and GPVAR provide a more effective balance between under-provisioning and energy savings than deterministic approaches. This makes them more suitable for real-world deployments where maintaining both network performance and resource efficiency is essential. Multivariate LSTM exhibits a high under-provisioning ratio (around 58%). The main rationale for this behavior is that is a single-point forecaster that optimizes the MSE, i.e., the mean of the PRB demand.

Among the evaluated methods, TFT demonstrates the best overall performance in terms of key forecasting metrics such as Continuous Ranked Probability Score (CRPS) , prediction deviation, and quantile loss. Its stronger confidence calibration and more accurate representation of traffic trends contribute to lower over-provisioning rates at higher percentiles and improved energy savings.

4.3.3 Energy-Aware enabler with Power Grid Index

The detailed description of the algorithm, the TSO–VPP (Transmission System Operator interface/ Virtual Power Plant), and the benchmark grid model are provided in Section 4.2.8 (Power Grid Representation Model) of D3.1 and it's out of scope of this deliverable's purpose. The present section focuses on the contribution context and the planned direction of the work.

Beyond the state of the art, the green index is reconstructed within a co-simulation environment, providing reproducible grid data for the project's training activities. The VPP / 6G-domain, ancillary-service-oriented context, coupled with the TSO–VPP communication architecture, positions the "energy-colour" signal as an enabler for the project's distributed-AI services.

It should be stated explicitly that IPTO acts here as an *enabler* rather than as a consumer of the signal, the algorithm itself is not operated by IPTO towards any orchestration or scheduling decision. Its purpose is to supply grid-aware, reproducible data that assists the training phase of the distributed-AI algorithms developed in the project. The decisions that act on the signal, and the outcomes that realize the project's sustainability targets, reside within other components of the architecture.

The system will be extended along several directions, reported here as planned work rather than implemented capability. Day-ahead forecasting is foreseen for both load and the

renewable ratio, using national data with weather as an input. The same implementation can also be instantiated as two independent simulators, each driven by the data of a distinct geographical location, producing two distinct indexes. The aforementioned signals are intended to support environmentally aware operation across space and time. Within the broader European energy-market context, in which aggregators participate through day-ahead and intraday mechanisms, the present work targets the grid-aware signal informing such participation and does not itself implement market clearing or bidding.

5 SEMANTIC COMMUNICATION

5.1 INTRODUCTION

Semantic Communication (SemCom) enables communication systems to concentrate on what information is crucial rather than how many bits are sent by integrating semantic extraction, knowledge bases, and intelligent encoding/decoding mechanisms. This method improves energy efficiency, boosts responsiveness for time-sensitive applications, and drastically lowers resource consumption. Semantic awareness also enables priority-driven communication, where critical information such as emergency alerts, disaster warnings, and rescue coordination messages can be transmitted with higher priority than less important data, ensuring timely and reliable communication in disaster scenarios.

Despite its potential, several obstacles must be overcome when integrating semantic communication into real-world 6G systems, especially in TN-NTN integrated O-RAN architectures. These systems involve communication, computation, and control among user devices, edge nodes, and cloud infrastructure. To operate efficiently, radio, computing, and network resources must be jointly optimized while adapting to changing channel conditions, user mobility, and diverse service needs. Additionally, the presence of mixed semantic and bit-oriented traffic, dynamic traffic patterns (both periodic and event-driven), and varying application requirements further complicate resource management. From a system perspective, the goal is to maximize semantic throughput while minimizing latency and energy consumption across the entire network. From a user perspective, the focus is on ensuring task success, maintaining semantic fidelity, and reducing Age of Information (AoI) for time-sensitive services through context-aware transmission strategies. However, the dynamic, distributed, and multi-layered nature of O-RAN-based TN-NTN systems renders traditional static optimization approaches inadequate, necessitating AI-driven, closed-loop control mechanisms for real-time and adaptive network management. Leveraging the O-RAN framework, including non-real-time and near-real-time RICs, the network can continuously monitor system conditions, learn optimal policies, and dynamically adjust semantic compression, scheduling, traffic steering, and resource allocation decisions.

In the following, we design and assess three different algorithms: one for task offloading, the second one for resource allocation, and the last one for users scheduling. Finally, a semantic-aware O-RAN-enabled NTN framework that leverages AI-driven feature extraction, adaptive compression, and dynamic resource allocation is introduced. All the proposed solutions are based on the scenario described in Section 5.2.

Before delving into the description of the architecture and designed solutions, we provide a brief state of the art of the major application of semantic communication for wireless systems.

5.1.1 State of the art

Semantic communication has emerged as a powerful paradigm for improving efficiency in future 6G systems by transmitting task-relevant information rather than raw data. A closely related concept is the AoI, which captures data freshness and is critical for time-sensitive applications. Early studies predominantly treated AoI and semantic-aware communication independently: AoI-based designs focused on minimizing freshness-related metrics, whereas semantic communication prioritized data importance without explicitly accounting for temporal relevance. More recent research bridges this gap by jointly modeling data freshness and semantic relevance. Compression-based forecasting frameworks consider both data relevance and AoI, enabling the system to discard stale information and transmit only timely and task-relevant data, thereby improving prediction accuracy and communication efficiency [138]. A notable contribution further derives structured optimal policies that jointly account for AoI and semantic value, demonstrating substantial performance gains over designs that treat these aspects separately [139].

Semantic awareness has also been incorporated into mobile edge computing and task offloading. Semantic-aware multi-task offloading frameworks leverage compact semantic representations instead of raw data transmission, significantly reducing network congestion and energy consumption. By combining semantic compression with reinforcement learning based decision-making, these systems jointly optimize communication, computation, and QoE, yielding improved latency and energy efficiency [140]. Building on this principle, Puligheddu et al. [141] propose SEM-O-RAN, a semantic O-RAN slicing framework for computer vision task offloading. The objective is to optimize data compression and network usage in a semantic-adaptive manner, rather than treating all offloaded data uniformly. The main functional blocks are the deep learning analyzer and the semantic edge slicing module, which are mapped to the non-real-time RIC (non-RT RIC) and near-real-time RIC (near-RT RIC), respectively. However, many existing solutions assume static network conditions or independent tasks, limiting their applicability in realistic and dynamic environments.

The integration of semantic communication with O-RAN architectures offers a promising path toward semantic-aware network control. A semantic-aware O-RAN architecture is introduced in [142] comprising three main modules: the central unit semantic plane (CU-SP), the semantic RIC (S-RIC), and the semantic engine. The semantic engine is responsible for orchestrating semantic information, managing the life cycle of semantic models, and supporting user experience management. The S-RIC provides a programmable platform for deploying

semantic-oriented applications, referred to as semantic xApps, and is designed to interact with the control plane (CP) and user plane (UP) of the central unit (CU) and distributed unit (DU), either through O-RAN-specified interfaces or through dedicated interfaces to be defined. The CU-SP receives instructions from the S-RIC, enhances the interaction with the CU-CP and CU-UP for task coordination, and receives, processes, and forwards semantic information from the enhanced O-DU and O-RU. At the medium access control (MAC) layer, [143] designs a semantic-aware MAC scheduling framework for XR applications. In contrast to conventional schedulers based on proportional fairness (PF) or network-level metrics, the proposed scheduler incorporates QoE-related information and employs a three-dimensional convolutional neural network (3D-CNN) for frame prediction and semantic relevance estimation, demonstrating QoE improvements over PF-based and frame-level integrated schedulers. More broadly, a semantic scheduling framework can augment network pipelines by introducing semantic processing units at the user side, within the network control plane, and close to the rendering engine. Such units may extract compact descriptors related to predicted user motion, expected scene variation, update relevance, and prediction confidence. At a higher level, a semantic orchestrator can combine these descriptors with network-side information to support QoE-aware decisions among users belonging to the same semantic class, while a semantic unit near the rendering and compression functions can assist in selecting rendering quality and compression level over a decision horizon. These considerations motivate further investigation into the role of semantic units within O-RAN, particularly regarding how they should be tailored to support the expected QoS and QoE.

Extending semantic communication to NTNs has been identified as a key enabler for global 6G connectivity. Semantic-aware NTN frameworks integrate AI-driven compression and semantic control into O-RAN architectures, achieving substantial delay reductions while preserving inference accuracy [144]. A more comprehensive semantic-empowered NTN framework extends semantic principles across the entire communication stack, including sampling, joint semantic-channel coding, routing, and resource allocation, leading to improved robustness and QoE in IoT and emergency communication scenarios [145].

5.2 ARCHITECTURE

Figure 60 shows a semantic-aware O-RAN UNITY-6G architecture for integrated TN and NTN. Semantic intelligence is built into the whole network architecture. One important part of the design is the hierarchical semantic logic in the RAN, transport, and core areas. At the RAN level, semantic-aware xApps within the Near-RT RIC function as dynamic, plug-and-play control agents that handle real-time RAN telemetry and semantic context. It enables the E2

interface to make intelligent decisions about scheduling, resource allocation, and adaptive control.

In the transport domain, the semantic transport logic works by having the O-DU send information to the transport network through the cooperative transport interface (CTI). This allows the O-DU and the transport infrastructure to share resources in a way that takes into account the current situation. This helps the transport network to understand the state of the network and the needs of services in a manner that makes reasonable sense. It also lets the transport network manage shared resources across many endpoints by adapting to meet changing needs while keeping communication between the O-DU and O-RU quick and simple.

As the core network becomes more complex, semantic intelligence is extended into the core/cloud domain through the DMO Core Semantic function. The DMO Core Semantic manages semantic information exchanged between the core network, applications, and cloud services, enabling semantic-aware service orchestration and coordination across the integrated TN-NTN infrastructure. Together with the DMO Application Semantic function, it provides semantic context to support intelligent service delivery and cross-domain coordination.

Another important design choice is to improve the semantics of the SMO framework. The non-RT RIC, which is part of the SMO, is important for long-term RAN intelligence, policy creation, and semantic model management. The non-RT RIC's rApps ecosystem uses services that are exposed by the SMO through the R1 interface to do AI/ML-driven tasks like training models, making predictions, and updating cycles. This enables the network to change all the time to meet changing service needs. It affects near-real-time operations by making policies and control guidance that are sent to the Near-RT RIC through the A1 interface. This connects real-time RAN optimization with high-level semantic intelligence.

The architecture can be broadened to include Y1 consumers, defined as external or internal entities that securely access RAN analytics and information services provided by the Near-RT RIC via authenticated and authorized interfaces, facilitating regulated semantic information exchange beyond the operator domain. Distributed semantic management is realized through the DMO Core Semantic, DMO Application Semantic, and IDMO functions. The DMO Core Semantic coordinates semantic information within the network infrastructure, whereas the DMO Application Semantic manages semantic interactions with application services such as XR applications. The IDMO provides hierarchical semantic coordination across multiple network domains, enabling consistent semantic management throughout the integrated TN-NTN system. Smart gateway mechanisms also make sure that TN-NTN segments can work together without any problems, and that operations are driven by intent. Semantic encoding

and decoding functions used in both cloud and user domains make end-to-end semantic communication possible.

At the service layer, the HOLO client represents the user-side extended reality (XR) application that generates semantic content. A semantic decoder reconstructs the received semantic information into application-level content before delivery to the end user. The decoded information is exchanged through a Smart Gateway Client, which provides interoperability between the user application and the semantic communication framework. On the network side, the HOLO Cloud Server hosts cloud-based semantic processing services, while the Semantic Encoder converts application data into compact semantic representations before transmission. A corresponding Smart Gateway Server enables secure and seamless interaction between cloud services and the semantic-aware network infrastructure. Together, these components enable end-to-end semantic communication between user devices and cloud applications across the integrated terrestrial and non-terrestrial network.

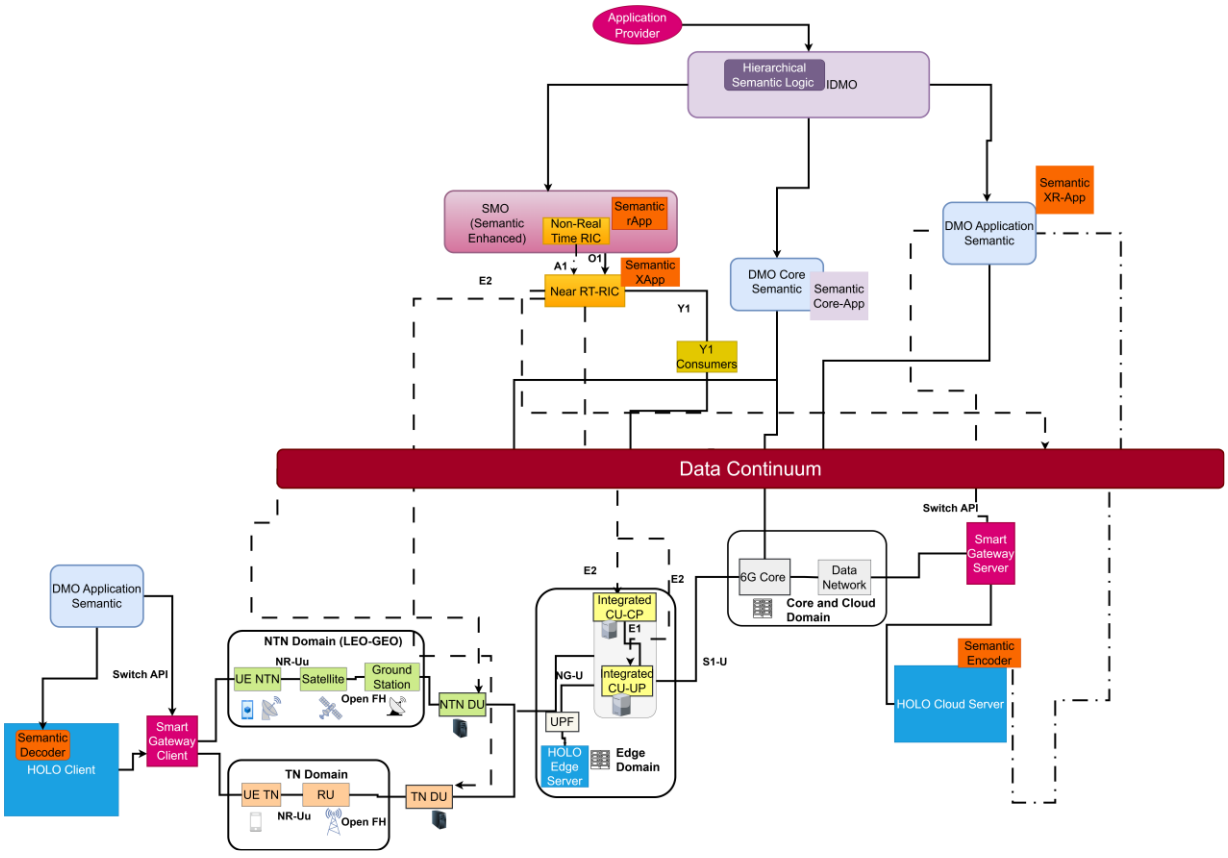


Figure 60: semantic-aware O-RAN UNITY-6G architecture for integrated TN and NTN

5.2.1 Targeted KPIs

The proposed concepts contribute to several project KPIs by enhancing network performance, efficiency, and adaptability in dynamic multi-domain environments. The targeted KPIs are summarized below:

Latency, Reliability, and QoS Guarantees

Semantic solutions help faster and more accurate decision-making to meet strict latency, PDR, and reliability requirements.

- KPI #2: Ensures end-to-end latency below 20 ms and reliability guarantees achieving 99% Packet Delivery Ratio (PDR) for time-sensitive applications in dynamic Non-Public Network (NPN) scenarios.
- KPI #7: Maintains cross-domain (cellular – IEEE 802.11) end-to-end latency below 20 ms.
- KPI #8: Limits cross-domain jitter to a maximum of 2 ms, with less than 10% of packets missing the deadline.
- KPI #37: Supports ultra-high service reliability, targeting 99.99999% service continuity through proactive AI-driven management.

Real-Time Adaptation and Control

Semantic abstraction reduces decision complexity, enabling faster network reconfiguration within strict time limits.

- KPI #5: Enables real-time reconfiguration of cellular network domains within less than 10 ms after decision-making.
- KPI #34: Contributes to reducing Open RAN real-time control loop timescales below 10 ms.

Efficiency, Resource Utilization, and Scalability

Semantic-based control reduces overhead and improves resource usage across large-scale multi-domain networks.

- KPI #22: Increases the number of network service lifecycle management (LCM) tasks by 20%.
- KPI #23: Reduces communication overhead by up to a factor of 10 while preserving SLA compliance.
- KPI #33: Supports scalability by enabling orders of magnitude more tenants through distributed AI and resource management.
- KPI #40: Enhances edge computational resource utilization in the cloud by up to 40%.

Energy Efficiency and Sustainability

Semantic awareness helps optimize energy use by making more efficient scheduling and transmission decisions.

- KPI #15: Reduces CNF energy consumption, contributing to up to 25% energy savings.
- KPI #16: Lowers radio link power consumption by up to 30% per link through coordinated load balancing and distributed AI.
- KPI #43: Improves overall network energy efficiency by up to 5× via energy-aware scheduling, resource allocation, and AI-driven optimization.

Cost Reduction and Operational Efficiency

Semantic automation reduces manual effort and operational overhead, lowering CAPEX and OPEX.

- KPI #17: Contributes to CAPEX and OPEX reduction through shared resource usage.
- KPI #19: Decreases network management overhead and reaction time via proactive, integrated resource management.
- KPI #38: Reduces operational expenditure by up to 30% through automation and improved resource utilization.

AI and Learning Optimization

Semantic solutions reduce training time by using structured data instead of raw data, improving learning speed.

- KPI #27: Reduces AI/DRL training and inference time by up to 80% through AI-assisted data generation.

Reliability in Advanced Network Scenarios

Semantic understanding improves prediction and proactive handling of failures to maintain reliability.

- KPI #35: Improves reliability in NTN, enhancing reliability ratios from 10^{-3} to 10^{-4}.
- KPI #45: Ensures that semantic-aware communication techniques do not degrade SLA guarantees or KPI compliance.

5.3 AOI-AWARE SEMANTIC COMMUNICATION

An AoI-aware semantic communication framework is introduced to improve communication efficiency in NTN-based systems. Instead of transmitting raw data, the proposed approach extracts and transmits task-relevant semantic information, while explicitly considering data freshness through the AoI. A learning-based offloading strategy is used to dynamically decide whether data should be processed locally or offloaded over NTN links, based on semantic importance, latency constraints, and energy cost. This design enables the system to prioritize fresh and meaningful information, reduce unnecessary transmissions, and improve overall latency and energy efficiency. To mathematically characterize data freshness, the AoI at UE u is defined as

$$\Delta_u(t) = t - \tau_u(t),$$

where $\tau_u(t)$ denotes the generation time of the most recently successfully delivered semantic update at time t.

The AoI evolves a stochastic process depending on whether a successful semantic update occurs. Let $\chi_u(t)$ denote the update success indicator, modeled as a Bernoulli random variable:

$$\chi_u(t) \sim \text{Bernoulli}\big(p_u(t)\big),$$

where $p_u(t)$ is the probability of successful semantic delivery, which depends on the execution mode, semantic compression ratio, channel state information, and routing decision.

The AoI evolution is then given by

$$\Delta_u(t+1) = \{1, if \chi_u(t) = 1; \Delta_u(t) + 1, if \chi_u(t) = 0\}$$

A semantic update is considered successful if the task is executed locally, at a regenerative satellite, or at the MEC via transparent forwarding, and satisfies the semantic accuracy constraint:

$$A_u(t) \geq A_u^{min}$$

The success probability can be written as a function of the control decision vector:

$$p_u(t) = f\big(y_u(t), \beta_u(t), r_u(t), CSI_u(t)\big)$$

where $y_u(t)$denotes the execution mode, $\beta_u(t)$ is the semantic compression ratio, and $r_u(t)$ is the routing decision. The expected AoI evolution is then expressed as

$$E[\Delta_u(t+1)] = \big(1 - p_u(t)\big)(\Delta_u(t) + 1) + p_u(t) \cdot \overline{T_u^{\text{upd}}}(t)$$

where $\overline{T_u^{\mathrm{upd}}}(t)$ denotes the effective semantic update delay, i.e., the time required to successfully generate, transmit, and decode a semantic update. This formulation explicitly shows that AoI is a decision-dependent stochastic state variable, tightly coupled with semantic compression, routing, and execution mode selection in the proposed NTN framework.

5.3.1 System model

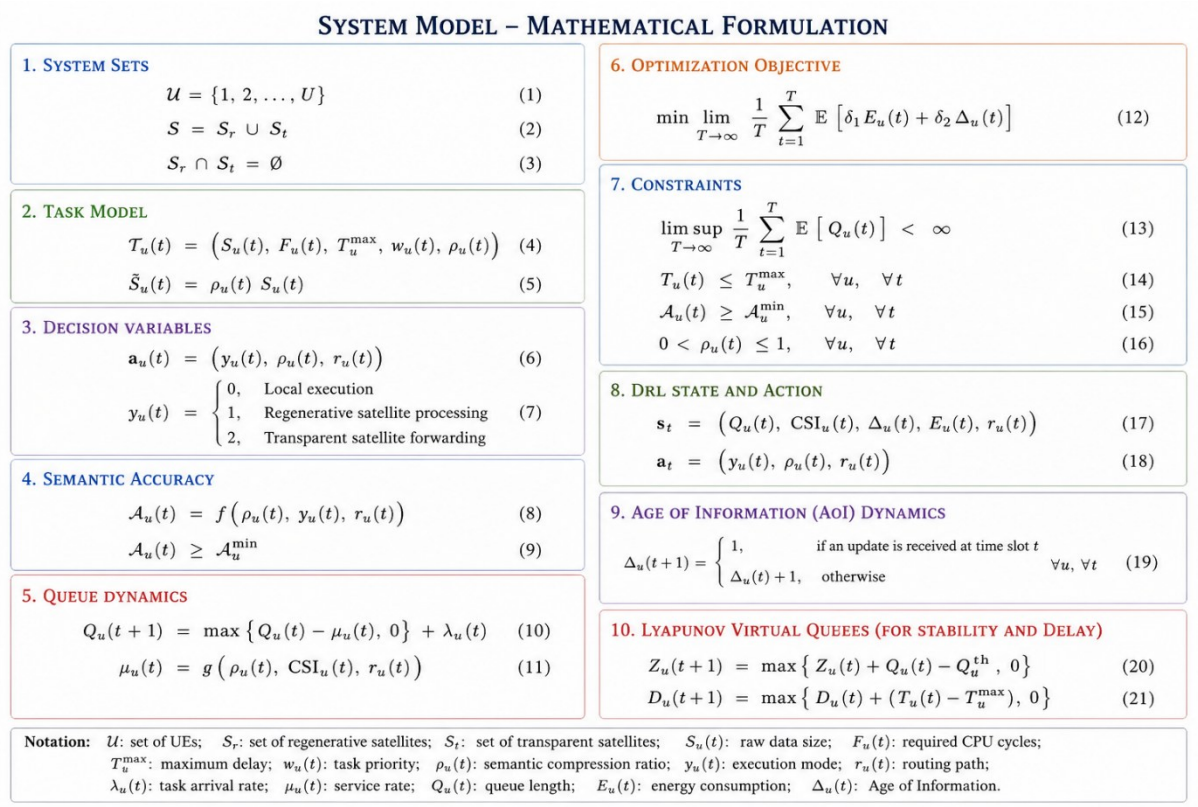
The proposed system introduces a semantic-aware task offloading framework for NTNs, where UEs, heterogeneous satellites, and a cloud-based MEC server collaboratively process delay-sensitive and computation-intensive tasks. The system is particularly designed for dynamic and mission-critical scenarios such as disaster recovery, where terrestrial infrastructure may be unreliable or unavailable. In this framework, the satellite layer consists of both regenerative satellites, which are capable of onboard processing, and transparent satellites, which act as relay nodes forwarding data to the MEC server. Unlike conventional systems, the key distinguishing feature here is the incorporation of semantic communication, where the transmitted information is compressed based on its meaning rather than raw data size. Each UE generates tasks characterized by data size, computational requirement, latency constraint, and priority level, and applies a controllable semantic compression ratio to reduce transmission overhead while maintaining acceptable semantic accuracy.

At each time slot, UEs make intelligent decisions regarding task execution mode, semantic compression level, and routing path across the multi-constellation satellite network. These decisions depend on multiple factors, including channel conditions, task priority, energy availability, and required semantic accuracy. High-priority tasks are given preferential treatment and are more likely to be processed either locally or through regenerative satellites to minimize delay and ensure timely updates. In contrast, less critical tasks may be aggressively compressed and offloaded to the MEC server via transparent satellites to conserve energy and communication resources. The system therefore dynamically balances the trade-off between energy consumption, latency, and semantic fidelity. The notion of semantic accuracy is explicitly modeled as a function of the compression ratio, execution mode, and routing decisions, ensuring that tasks meet minimum quality requirements regardless of where they are processed.

To manage system dynamics, each UE maintains multiple queues, including task arrival, local processing, and semantic offloading queues. These queues evolve based on stochastic task arrivals, scheduling decisions, and service rates influenced by compression and resource availability. A priority-aware scheduling mechanism ensures that critical tasks are processed immediately, while non-critical tasks are delayed in a controlled manner to avoid congestion. The overall system objective is formulated as a long-term stochastic optimization problem that

minimizes the weighted sum of energy consumption and AoI, while satisfying constraints on delay, semantic accuracy, and queue stability. Due to the coupling among users and shared satellite resources, the problem is further modeled as a multi-agent stochastic game, where each UE acts as a rational agent seeking to optimize its own performance while interacting with others.

To ensure long-term stability, Lyapunov optimization is employed by introducing virtual queues that capture delay and backlog constraints. This allows the transformation of the original time-averaged problem into a sequence of per-slot deterministic optimization problems. However, due to the high-dimensional and non-convex nature of the problem, deriving an analytical solution is intractable. Therefore, a deep reinforcement learning (DRL) framework is adopted to learn optimal decision policies in a model-free manner. The DRL agent observes the global system state, including queue lengths, channel conditions, AoI, energy levels, and satellite availability, and outputs optimal actions such as execution mode, compression ratio, and routing decisions. The reward function is carefully designed to penalize energy consumption, information staleness, and queue congestion, aligning with the Lyapunov drift-plus-penalty framework. Overall, the proposed semantic-aware system significantly improves resource efficiency by transmitting only meaningful information, reducing unnecessary communication overhead, and enabling adaptive, intelligent decision-making across the UE-satellite-MEC hierarchy.



System Model – Mathematical Formulation

1. System Sets

$$\mathcal{U} = \{1, 2, \ldots, U\} \quad (1)$$

$$\mathcal{S} = \mathcal{S}_r \cup \mathcal{S}_t \quad (2)$$

$$\mathcal{S}_r \cap \mathcal{S}_t = \emptyset \quad (3)$$

2. Task Model

$$\mathcal{T}_u(t) = \left(S_u(t),\ F_u(t),\ T_u^{\max},\ w_u(t),\ \rho_u(t)\right) \quad (4)$$

$$\tilde{S}_u(t) = \rho_u(t)\ S_u(t) \quad (5)$$

3. Decision Variables

$$\mathbf{a}_u(t) = \left(y_u(t),\ \rho_u(t),\ r_u(t)\right) \quad (6)$$

$$y_u(t) = \begin{cases} 0, & \text{Local execution} \\ 1, & \text{Regenerative satellite processing} \\ 2, & \text{Transparent satellite forwarding} \end{cases} \quad (7)$$

4. Semantic Accuracy

$$\mathcal{A}_u(t) = f\left(\rho_u(t),\ y_u(t),\ r_u(t)\right) \quad (8)$$

$$\mathcal{A}_u(t) \geq \mathcal{A}_u^{\min} \quad (9)$$

5. Queue Dynamics

$$Q_u(t+1) = \max\left\{Q_u(t) - \mu_u(t),\ 0\right\} + \lambda_u(t) \quad (10)$$

$$\mu_u(t) = g\left(\rho_u(t),\ \mathrm{CSI}_u(t),\ r_u(t)\right) \quad (11)$$

6. Optimization Objective

$$\min \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\delta_1 E_u(t) + \delta_2 \Delta_u(t)\right] \quad (12)$$

7. Constraints

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[Q_u(t)\right] < \infty \quad (13)$$

$$T_u(t) \leq T_u^{\max}, \quad \forall u, \quad \forall t \quad (14)$$

$$\mathcal{A}_u(t) \geq \mathcal{A}_u^{\min}, \quad \forall u, \quad \forall t \quad (15)$$

$$0 < \rho_u(t) \leq 1, \quad \forall u, \quad \forall t \quad (16)$$

8. DRL State and Action

$$\mathbf{s}_t = \left(Q_u(t),\ \mathrm{CSI}_u(t),\ \Delta_u(t),\ E_u(t),\ r_u(t)\right) \quad (17)$$

$$\mathbf{a}_t = \left(y_u(t),\ \rho_u(t),\ r_u(t)\right) \quad (18)$$

9. Age of Information (AoI) Dynamics

$$\Delta_u(t+1) = \begin{cases} 1, & \text{if an update is received at time slot } t \\ \Delta_u(t) + 1, & \text{otherwise} \end{cases} \quad \forall u,\ \forall t \quad (19)$$

10. Lyapunov Virtual Queues (for Stability and Delay)

$$Z_u(t+1) = \max\left\{Z_u(t) + Q_u(t) - Q_u^{\text{th}},\ 0\right\} \quad (20)$$

$$D_u(t+1) = \max\left\{D_u(t) + \left(T_u(t) - T_u^{\max}\right),\ 0\right\} \quad (21)$$

Notation: $\mathcal{U}$: set of UEs; $\mathcal{S}_r$: set of regenerative satellites; $\mathcal{S}_t$: set of transparent satellites; $S_u(t)$: raw data size; $F_u(t)$: required CPU cycles; $T_u^{\max}$: maximum delay; $w_u(t)$: task priority; $\rho_u(t)$: semantic compression ratio; $y_u(t)$: execution mode; $r_u(t)$: routing path; $\lambda_u(t)$: task arrival rate; $\mu_u(t)$: service rate; $Q_u(t)$: queue length; $E_u(t)$: energy consumption; $\Delta_u(t)$: Age of Information.

The mathematical model represents a semantic-aware NTN system where each equation defines a specific functional component. The set equation defines the group of UEs participating in the system, while the satellite set separates regenerative and transparent satellites. The task model describes each UE task in terms of data size, computation load, deadline, priority, and semantic compression ratio. The compressed data expression shows how semantic processing reduces transmitted information. The decision variable defines the joint control actions taken by each UE, including execution mode, compression level, and routing path. The semantic accuracy function captures the quality of received information, and the constraint ensures that minimum semantic quality is always maintained. The queue evolution equation models backlog dynamics based on arrivals and service rates. The service rate function depends on compression level, channel state information, and routing decisions, reflecting system adaptability under changing network conditions. The optimization objective minimizes the long-term trade-off between energy consumption and AOI, ensuring both efficiency and freshness of data. The stability constraint guarantees that queues remain bounded over time, preventing system congestion. The delay constraint ensures that tasks are completed within their maximum allowable time.

Finally, the DRL formulation defines the system state and action space used for learning-based optimization. The state includes queue status, channel conditions, AoI, energy levels, and routing decisions, while the action determines execution mode, compression, and routing. This enables adaptive decision-making in highly dynamic NTN environments.

5.3.2 Preliminary results

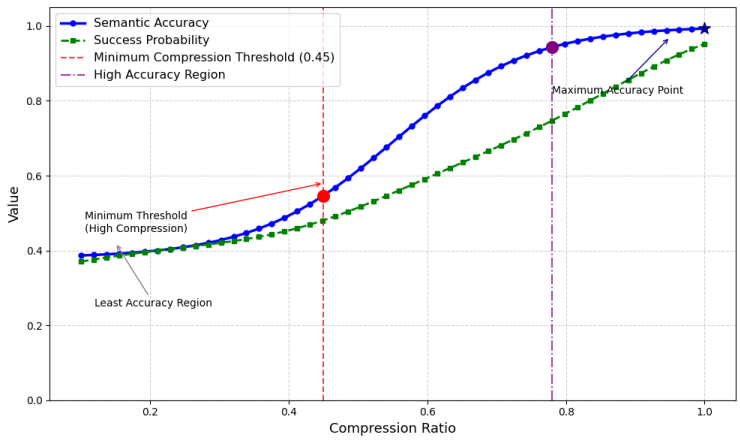


Figure 61: Impact of different compression ratios on semantic accuracy and task success probability

Impact of Semantic Compression on Accuracy and Task Success: To better understand the role of semantic compression in the proposed framework, we analyze the relationship between compression ratio, semantic accuracy, and task success probability. This is important because

compression is not only a communication optimization tool, but also directly affects the quality of task execution in NTN-assisted semantic systems.

In the proposed enhanced system, adaptive compression is integrated into the decision-making process of each UE, where the compression ratio $\rho_u(t)$ is dynamically adjusted based on task priority, channel conditions, and available energy resources. Figure 61 illustrates the impact of different compression ratios on semantic accuracy and task success probability. As shown, semantic accuracy increases with the compression ratio, meaning that less aggressive compression preserves more semantic information and improves task fidelity.

When the compression ratio falls below a threshold of approximately $\rho_u(t) < 0.45$, the semantic accuracy degrades significantly. In this region, excessive compression removes critical semantic features, leading to poor task reconstruction and a sharp drop in task success probability. This represents the low-fidelity operating region of the system. In contrast, when the compression ratio exceeds approximately $\rho_u(t) > 0.78$, the system enters a high-performance operating regime. In this region, semantic accuracy remains above 0.9, indicating that the extracted semantic information is sufficiently preserved while still benefiting from reduced transmission overhead.

The task success probability closely follows the same trend as semantic accuracy, confirming that reliable task execution in NTN systems is strongly dependent on semantic quality rather than raw data transmission. These results validate the effectiveness of the proposed adaptive semantic compression mechanism. By dynamically selecting the compression ratio based on system state information, the framework achieves a balance between communication efficiency and semantic fidelity, which is not possible in conventional fixed-rate transmission schemes.

5.4 SEMANTIC-AWARE RESOURCE ALLOCATION

The proposed architecture builds on deep joint source-channel coding (JSCC) for semantic transmission [146], semantic communication modelling [147], QoE-based semantic-aware resource allocation [148], and the O-RAN RIC/xApp/rApp control architecture [149]. As shown in Figure 62, it comprises three functional layers - a Knowledge Base at the top, a decision-making layer in the middle, and an execution layer at the bottom - connected by control signals (solid arrows) and feedback paths (dashed arrows). At the top, the Knowledge Base holds the learned mappings that characterise the underlying JSCC (the considered semantic encoder/decoder) link: the number of semantic symbols, the resulting semantic rate, and the achievable semantic fidelity at each operating point. These mappings are kept current through the Knowledge base update block on the right of the figure, which continuously ingests observed network state and refines the fidelity estimates. The fidelity modelling behind the Knowledge Base is implemented by two lightweight neural networks, the forward and inverse (backward) predictors - that have learned the quality surface of the underlying JSCC system.

The forward predictor answers: given that we transmit at this BPP under this channel condition, what image quality can we expect? The inverse predictor reverses this mapping: given a desired quality target and the current channel SNR, what is the minimum BPP we need to allocate?

The decision-making layer centres on the Semantic Controller xApp (QoE), which reads semantic rate and fidelity values from the Knowledge Base and combines them with per-user preference weights supplied by a dedicated Preference rApp. In parallel, the Traffic Management (TM) rApp translates SLA and intent information into slower-timescale resource management policy updates, distributing the information flows between the network nodes. The Semantic Controller xApp and the TM rApp both feed into the Resource Allocation (RA) xApp below.

The execution layer consists of the RA xApp and the observable Network state (UE parameters, channel conditions, cell assignments). The RA xApp uses the per-user BPP decisions received from above to allocate the resources in the radio environment depending on transmission conditions and the SLA needs; the resulting network state is then looped back up to the Knowledge base update, closing the control loop and allowing the system to continuously track varying channel conditions.

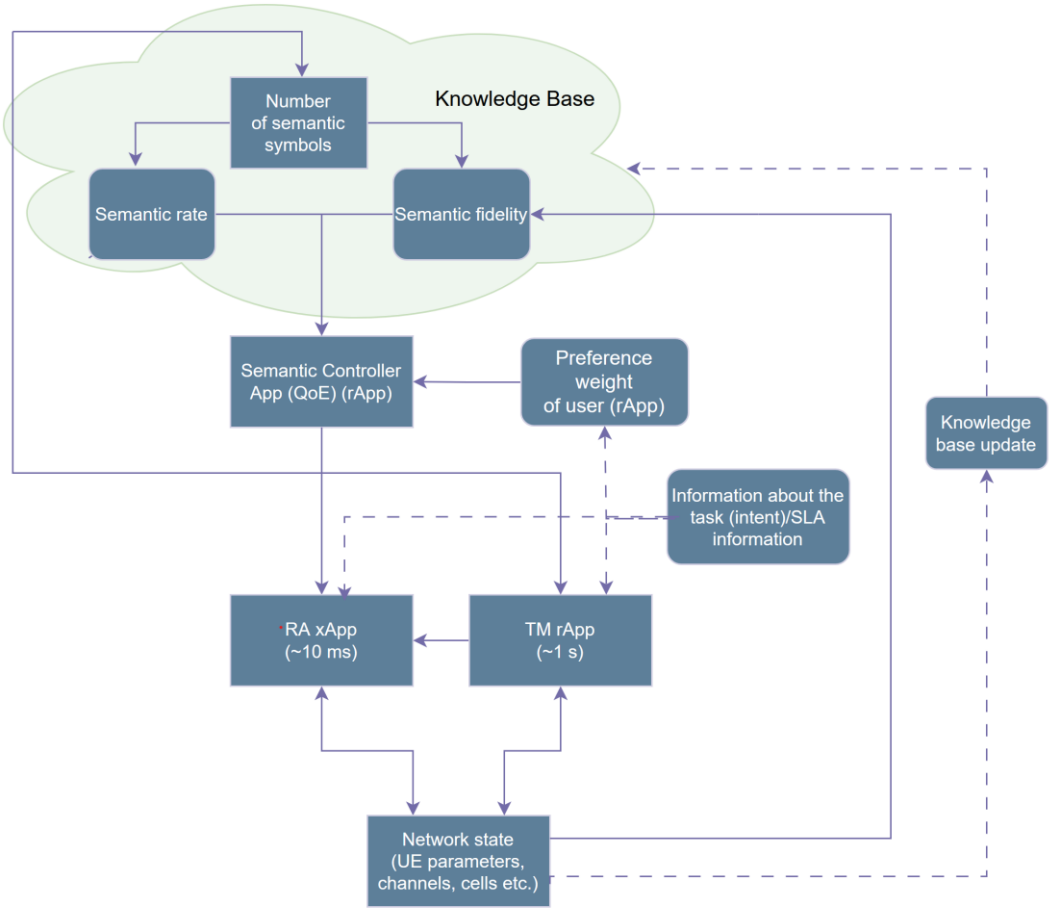


Figure 62: Proposed O-RAN-integrated semantic communication control architecture with closed-loop QoE management across Non-RT and Near-RT RIC layers.

5.4.1 Resource Allocation Problem Formulation

This section provides the theoretical foundation for the proposed semantic-aware resource allocation framework. Based on the system architecture introduced earlier, we formulate the optimization problem in terms of user QoE, which depends on semantic rate and semantic fidelity rather than only conventional bit-level metrics.

Let B denote the set of network nodes, U the set of users, and M in the set of available resource blocks. The general optimization problem can be expressed as:

$$\frac{max}{\{x_{b,u}\},\{a_{b,u,m}\},\{k_u\}} \sum_{b\in B} QoE_u$$

Where:

- $x_{b,u}$ is a binary variable representing association of UE u with node b;

- $a_{b,u,m}$ is a binary variable indicating the allocation of resource block m to UE u in node b;
- $k_u$represents the semantic encoding compression level.

Such problem formulation is multidimensional; thus, we split it into three subproblems: SP1 - semantic compression optimization, SP2 - traffic Management (associating UEs or information flows with network nodes), and SP3 - resource allocation in each node.

Semantic-based QoE modelling

Unlike conventional QoE formulations based on bit-level metrics, the proposed model captures user experience directly at the semantic level.

For user u, the semantic QoE is expressed as:

$$QoE_u = w_u G_R(R^u_{sem}) + (1 - w_u) G_A(F^u_{sem})$$

Where $w_u \in [0,1]$ is a user-specific preference parameter that balances semantic-rate efficiency and semantic fidelity. A larger w_u gives more importance to efficient semantic information delivery, while a smaller $w_u$prioritizes the quality of the reconstructed semantic content. The quantities $R^u_{\text{sem}}$and F^u_{sem} denote the two underlying semantic components used by the QoE model: the semantic rate and the semantic fidelity, respectively. They are introduced below before being mapped by $G_R(\cdot)$ and $G_A(\cdot)$ into normalized QoE scores.

The semantic rate characterizes the efficiency of semantic information delivery:

$$R_{sem} = \frac{I_{sem}(S)}{C}$$

where $I_{sem}(S)$ denotes the amount of semantic information contained in the representation *S*, while *C* is the transmission cost, e.g., the number of used radio resource blocks, symbols, or time units.

Semantic fidelity quantifies the accuracy of semantic information delivery and measures how well the reconstructed semantic representation preserves the intended meaning of the source. It is defined as a normalized task dependent similarity metric.

$$F_{sem} = \phi(S, \hat{S})$$

where $\phi(.)$ denotes a semantic-domain performance function reflecting task success at the receiver. Depending on the considered application, $\phi(.)$ may correspond to semantic similarity, task accuracy, or perceptual quality metrics, as commonly assumed in semantic-aware QoE modelling.

The normalized semantic rate and fidelity contributions are given by:

$$G_R(R_{sem}^u) = \frac{1}{1 + \exp\left(a_R\left(\tilde{R} - R_{sem}^u\right)\right)}$$

$$G_A(F_{sem}^u) = \frac{1}{1 + \exp\left(a_A\left(\tilde{F} - F_{sem}^u\right)\right)}$$

where $\tilde{R}$ and $\tilde{F}$ denote semantic rate and semantic fidelity thresholds, respectively, and a_R and a_A control the steepness of the transition. This formulation provides a smooth bounded mapping from semantic performance to QoE, consistent with the semantic rate–fidelity framework and established semantic-aware QoE models.

Multi-Dimensional Semantic Fidelity

The choice of fidelity metrics depends on the type of content being transmitted: for speech this might be intelligibility or MOS-like measures, for text the exact token match, and for structured data the task-level accuracy. Since this work considers image transmission, the metrics must capture both low-level signal fidelity and higher-level perceptual and semantic content preservation. Image quality is assessed across five complementary metrics:

PSNR (pixel accuracy), SSIM (structural fidelity), LPIPS (perceptual similarity via a pre-trained network), CLIP similarity (semantic content preservation via a vision-language model), and a composite Semantic Fidelity Score (SFS) that combines all four.

This multi-dimensional view is essential for O-RAN integration: it exposes trade-offs invisible to single-metric evaluation and allows rApps to express QoE requirements in application-appropriate units, a surveillance system may specify a minimum SSIM, while a semantic search application may require a minimum CLIP score.

Fidelity Predictors and Closed-Loop Rate Adaptation.

The core novelty of this contribution is a pair of lightweight neural predictors that together enable closed-loop fidelity control. The key operating variable is BPP (Bits per Pixel), or bits per pixel, which denotes the effective compression rate used by the JSCC encoder. Lower BPP means that fewer bits are transmitted per image pixel, reducing bandwidth cost but usually lowering reconstruction fidelity; higher BPP preserves more visual and semantic detail, but consumes more radio resources. As shown in the architecture diagram (**Figure 63**Figure 63), the forward predictor maps any BPP and SNR operating point to expected values of all five quality metrics, providing near-instant estimates without running the full JSCC pipeline. It uses a shared fully connected backbone with GELU activations and dropout, followed by five independent output heads corresponding to PSNR, SSIM, LPIPS, CLIP, and SFS. The inverse predictor (Figure 64**Figure 64**) inverts this relationship: given a desired fidelity target and the

current channel SNR, it outputs the minimum BPP sufficient to meet that target. It is also implemented as a compact fully connected network, with a Softplus output layer used to ensure a positive BPP prediction. Together, they implement a simple feedback loop: whenever channel conditions change, the inverse predictor recomputes the required BPP and the encoder quantiser is updated accordingly. The operator sets the quality target once; the system maintains it autonomously as both BPP and SNR vary, with no lookup tables and no receiver feedback. These predictors populate the Knowledge Base entries, semantic rate and semantic fidelity, that the Semantic Controller xApp queries at run time. This is what makes closed-loop rate adaptation efficient: instead of repeatedly running the full JSCC pipeline to explore the quality space, the predictors provide near-instant estimates that can be queried hundreds of times per second.

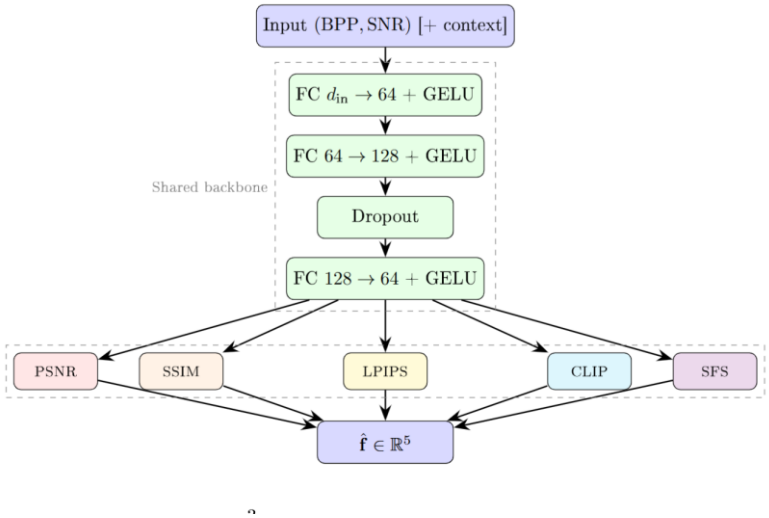


Figure 63: Forward predictor used to estimate quality metrics from BPP and SNR operating conditions.

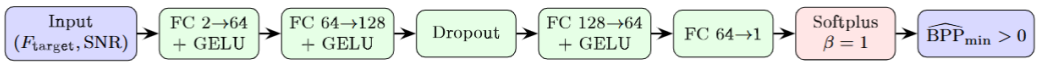


Figure 64: Inverse predictor used to estimate the minimum BPP required to satisfy a target fidelity level under a given SNR condition.

Quality-of-Experience Management and O-RAN Integration

The decision-making layer is embedded in a broader QoE management framework for multi-user networks. The Semantic Controller xApp is implemented as a DQN agent that acts as a central per-cell controller, selecting the optimal compression level (BPP) for every active user. At each step the agent receives two groups of external data: network-side information (the current channel quality of each user) and QoE-side information supplied from above (the user's preference weight between fidelity and rate, the minimum acceptable fidelity and rate, and a global feasibility threshold). Based on these inputs it chooses one compression level per user from the set 3–8 bits per pixel, with a reward that penalises both over-compression (poor image

quality) and under-compression (wasted bandwidth) and drops to zero whenever a user falls below the feasibility threshold. The Q-network shown in Figure 65Figure 65 is a simple feed-forward network with three hidden layers of 256 ReLU units, trained with experience replay and a target network; exploration is annealed over the first 1 500 episodes, after which the policy converges to near-optimal aggregate QoE. The RA xApp executes the chosen BPP in the fast 10ms loop, while the TM rApp at the non-RT RIC translates SLA and intent information into QoE parameter updates via the A1 interface, with the Knowledge Base acting as a shared quality model at all control layers. The detailed structure of RA and TM apps will be defined in the next stage of this work; however, RL approach is envisaged.

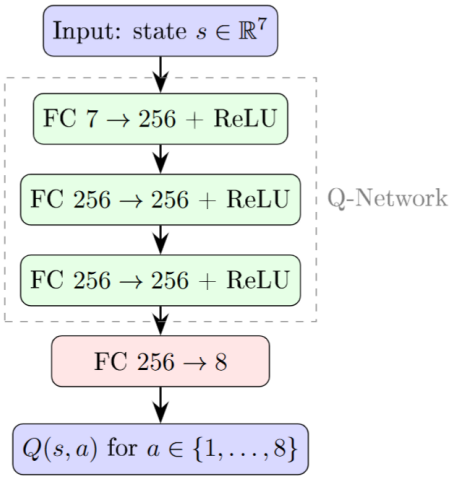


Figure 65: DQN agent Q-network architecture: three hidden layers of 256 neurons with ReLU activations. Output is 8 Q-values corresponding to 8 quantisation levels.

5.4.2 Numerical results

JSCC Training and Quality Surface

The JSCC encoder-decoder was trained jointly over a broad range of channel conditions and compression rates, yielding a single model that covers the entire quality-rate-SNR operating space without retraining. **Error! Reference source not found.** shows heatmaps of perceptual quality metrics as a function of BPP and channel SNR (dB). Left: LPIPS (lower is better) - values decrease consistently as BPP and SNR increase, reaching a minimum of 0.26 at high rates and good channel condition. Right: CLIP Similarity (higher is better) - values approach 0.85 at favourable operating points, confirming strong semantic preservation. **Error! Reference source not found.** Results for selected perceptual and semantic metrics, exhibits smooth and predictable behavior at all evaluated operating points, confirming the elimination of the cliff effect and validating the end-to-end training strategy.

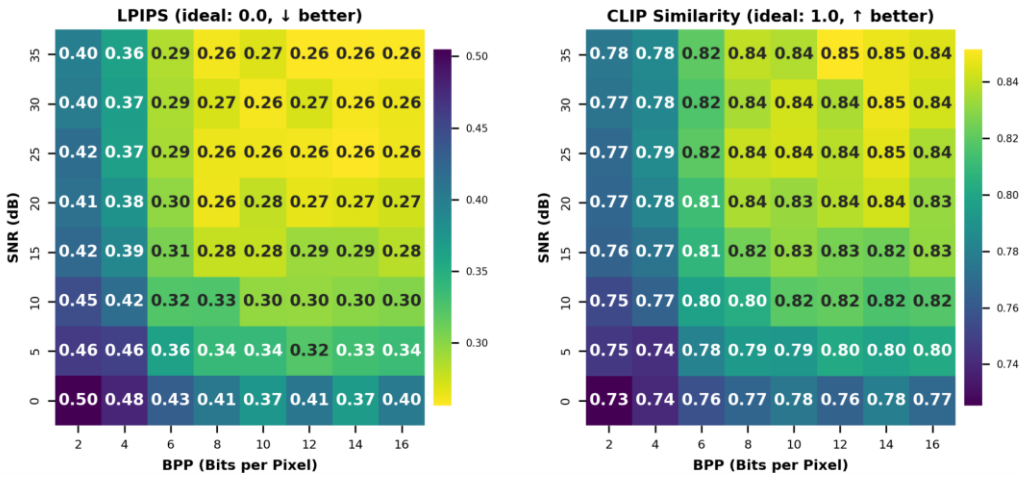


Figure 66: LPIPS and CLIP Similarity heatmaps for different operating points.

Fidelity Predictors: Set-and-Maintain Rate Control

Two lightweight neural predictors were trained offline on real quality measurements collected from the JSCC system. The forward predictor accurately reproduces the quality surface across all operating conditions; the inverse (backward) predictor reverses this mapping, answering the rate-allocation question directly: given a desired quality level and the current channel conditions, it returns the minimum BPP to allocate. Figure 67 shows the learned inverse surfaces for CLIP (left) and LPIPS (right). The colour encodes the minimum BPP required for each (target quality, SNR) pair. Low-to-moderate targets remain cheap (green, BPP ≈ 2–4); extreme targets drive a sharp rate spike (red), regardless of channel quality. Each operating point on the surface corresponds to a unique (target quality, SNR) context, and an rApp maintains a quality target simply by reading off the required BPP at every channel update.

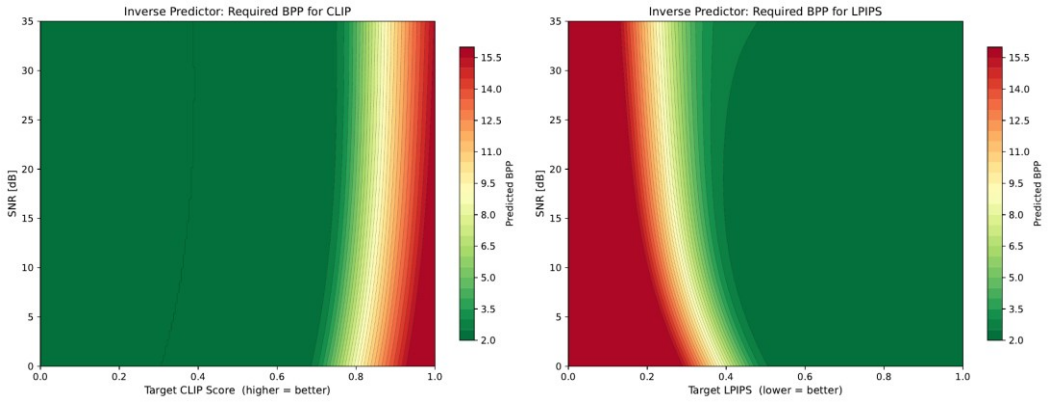


Figure 67: Inverse predictor surfaces showing the minimum BPP required to achieve target CLIP and LPIPS quality levels under different SNR conditions.

QoE Results

The DQN-based Semantic Controller xApp was evaluated in a multi-cell, multi-user O-RAN scenario where each user is characterised by individual preference weights balancing semantic rate and fidelity. The agent converges to near-ceiling aggregate QoE with full per-user feasibility, demonstrating that adaptive rate selection efficiently handles user heterogeneity across varying channel conditions. Fixed-rate baselines, by contrast, either sacrifice QoE for cell-edge users at high rates or waste bandwidth at low rates, confirming the advantage of learned, channel-aware rate control. Figure 68 shows the learning curve, comparison against baselines, and the learned BPP distribution - which mixes low- and high-rate actions rather than settling on a single fixed rate. Top left: the mean QoE per link increases over 2000 training episodes and converges close to the trained DQN reference value. Top right: the learned policy mainly selects 4 BPP and 14 BPP, with additional mass at 6, 10, and 12 BPP, showing that the controller learns an adaptive allocation strategy rather than collapsing to a single fixed-rate rule. Bottom: the DQN policy achieves 113% of the best fixed-BPP baseline, with mean QoE = 0.79 compared with 0.69 for the best fixed setting, Fixed 6 BPP. Higher fixed-rate settings from 10 to 16 BPP fall to roughly 57-59% of the best fixed baseline, highlighting the cost of using overly high BPP when semantic rate is also part of the QoE objective.

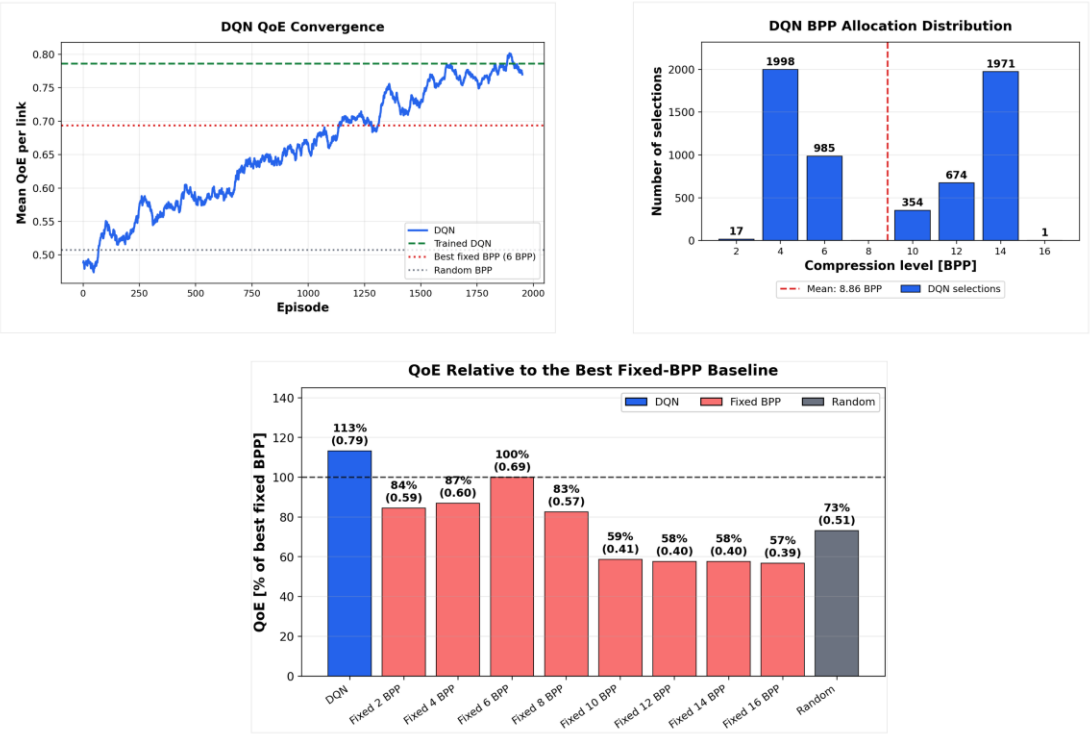


Figure 68: DQN evaluation results.

5.5 PERCEPTION-AWARE SEMANTIC SCHEDULING WITH REMOTE PROCESSING

The literature suggests that a semantic-aware network architecture is critical for the optimization of remote-rendered VR performance from a QoE perspective. For this reason, we consider a multi-tier semantic scheduling framework consisting of several distributed semantic-aware network elements. The proposed research direction has two main components. First, a suitable motion-prediction module should be adapted to the available sensor data and to the communication and computation characteristics of the network. This module should support the placement of semantic processing functions within the network while targeting reduced end-to-end latency and communication overhead under a desired QoE constraint. Second, a tailored semantic agentic unit should be designed to exploit the current network state and semantic descriptors in order to support scheduling decisions among users belonging to the same semantic class. The objective of this unit is to determine user-specific scheduling weights and quality targets that improve the overall QoE while remaining compatible with the available network resources and link-quality constraints.

5.5.1 Problem Formulation

In this section, we study the scheduling and association problem for users with high data-rate and ultra-low-latency requirements, with particular focus on remote-rendered VR services. In the considered system, the motion information of each user, e.g., head orientation and viewing direction, is sent to an edge or remote server. The server predicts the user's future motion at time $t + \Delta t$ and pre-renders the scene corresponding to the predicted viewing direction. Once the actual motion information is available, the pre-rendered scene is transmitted to the user if the prediction error is below a predefined threshold; otherwise, the scene is re-rendered according to the actual motion information. This mechanism can reduce the perceived latency by exploiting motion prediction and speculative rendering [150].

The relevant latency metric for immersive VR is the motion-to-photon (M2P) delay, defined as the time between the user's physical motion and the corresponding visual update reaching the user's eyes. Excessive M2P delay degrades the user's QoE, and M2P values above $20\ ms$ are commonly associated with discomfort, dizziness, and nausea [151]. It should be noted that, when asynchronous reprojection techniques such as Asynchronous Timewarp (ATW) are employed, the terminal-side M2P loop can be partially decoupled from the network rendering and streaming latency [152]. However, the user experience remains affected by the network-side latency, since ATW provides only an approximate correction of the rendered view and

cannot fully replace the delivery of fresh cloud-rendered content. Excessive cloud rendering and streaming latency may still lead to visual artifacts such as black borders, degraded image quality, or view inconsistency. Therefore, we consider a target M2P/QoE latency threshold whose value depends on the degree of user interaction, service class, target visual quality, and the compensation mechanisms available at the terminal. Based on Cloud VR requirements reported in [152], and [153], this target can be in the order of tens of milliseconds. In the following, we denote the adopted target value by $\mathrm{M2P}_{\max}$. Unlike other semantic approaches that mainly focus on viewport-aware rendering, scene-level semantic coding, or importance-aware video compression, here we focus on how user interaction and motion dynamics affect network-side scheduling and association decisions. In particular, the traffic generated by the remote rendering process depends on the user's motion and can be predicted using deep learning models such as LSTM networks [150], [152]. This predicted traffic information, together with network-side measurements and previous user experience, is used to improve the QoE of users sharing the same radio resources.

Let $\mathcal{U}$ denote the set of users, and let

$$\mathcal{C} = \mathcal{C}_{\mathcal{T}} \cup \mathcal{C}_{\mathcal{N}} \quad (1)$$

be the set of candidate serving cells, where $\mathcal{C}_{\mathcal{T}}$ and $\mathcal{C}_{\mathcal{N}}$ denote the sets of TN and network NTN cells, respectively. At each discrete time step t, the decision variable is the serving cell selected for user u, denoted by

$$c_u(t) \in \mathcal{C}. \quad (2)$$

The goal is to find a user-centric association policy that minimizes the long-term occurrence of undesirable M2P experiences while respecting the available radio resources, avoiding excessive handovers, and satisfying a radio-link reliability requirement expressed through a target block error rate (BLER).

M2P Delay Model

The complete M2P delay contains several components, including sensing, uplink transmission of motion information, prediction, rendering, video encoding, core and transport network delay, RAN delay, video decoding, and display delay. However, since the proposed decision process is assumed to be implemented by an xApp using RAN-side reports, it is not realistic to assume that all end-to-end delay components are directly observable through E2 reports.

Therefore, we decompose the M2P delay into an external component and an E2-estimable RAN component:

$$g_{u,t}\big(c_u(t)\big) = T_{u,t}^{\text{ext}} + \hat{T}_{u,c_u(t),t}^{\text{DL,E2}}, \quad (3)$$

where $g_{u,t}\big(c_u(t)\big)$ denotes the M2P estimate used by the xApp for user u at time t when associated with cell $c_u(t)$. The term $T_{u,t}^{\text{ext}}$ contains delay components that are external to the E2-observable RAN reports, such as sensing delay, uplink motion-reporting delay, prediction delay, rendering delay, video encoding delay, core/transport delay before the packet reaches the RAN, video decoding delay, and display delay. These terms may be estimated from application-layer telemetry or treated as exogenous with respect to the RAN association decision.

The E2-estimable downlink RAN delay is defined as

$$\hat{T}_{u,c,t}^{\text{DL,E2}} = \hat{T}_{u,c,t}^{\text{CU-UP}} + \hat{T}_{u,c,t}^{\text{DU}} + \hat{T}_{u,c,t}^{\text{Uu}} \quad (4)$$

Here, $\hat{T}_{u,c,t}^{\text{CU-UP}}$ represents the estimated delay contribution at the centralized unit user-plane side, including user-plane buffering and PDCP/SDAP-related processing. The term $\hat{T}_{u,c,t}^{\text{DU}}$ represents the estimated delay contribution at the distributed unit, including RLC/MAC buffering, scheduling delay, PRB availability, and high-PHY processing effects. The term $\hat{T}_{u,c,t}^{\text{Uu}}$ represents the estimated radio-interface delay, including the effects of signal-to-interference-plus-noise ratio (SINR), modulation and coding scheme (MCS), hybrid automatic repeat request (HARQ) retransmissions, and radio transmission over the air interface.

The QoE degradation associated with M2P is modeled through a non-decreasing cost function $f(\cdot)$, where larger M2P values induce higher cost. For example, $f(\cdot)$ may be designed to sharply penalize M2P values exceeding the VR comfort threshold.

SINR constraint from target BLER

In addition to the latency-related M2P cost, the association decision must satisfy a RAN-side reliability requirement. In this section, the radio-link reliability requirement is represented through a target residual BLER, denoted by $\text{BLER}^{\star}$. This target BLER is translated into an SINR threshold.

Let $\Gamma_{m,L}(\gamma)$ denote the BLER--SINR mapping for MCS m when at most L HARQ transmissions are allowed. This mapping can be obtained from analytical approximation, or calibrated BLER curves. The SINR threshold corresponding to the target BLER is defined as

$$\gamma_{m,L}^{\star} = \inf\{\gamma : \Gamma_{m,L}(\gamma) \le \text{BLER}^{\star}\}. \quad (5)$$

Therefore, when user u is associated with cell $c_u(t)$ at time t, the selected cell must satisfy

$$\gamma_{u,c_u(t),t} \geq \gamma^{\star}_{m_u(t),L_u^{max}(t)}, \qquad \forall u \in \mathcal{U}, \forall t, \qquad (6)$$

were $\gamma_{u,c_u(t),t}$ is the SINR experienced by user u from its selected serving cell, $m_u(t)$ is the selected MCS, and $L_u^{max}(t)$the maximum number of HARQ transmissions considered for user u at time t. The value of $L_u^{max}(t)$ould be consistent with the latency budget used in the M2P model, since additional HARQ retransmissions improve reliability but also increase delay.

If a fixed MCS and a fixed HARQ budget are assumed, (6) can be simplified to

$$\gamma_{u,c_u(t),t} \geq \gamma^{\star}, \qquad \forall u \in \mathcal{U}, \forall t, (7)$$

where $\gamma^{\star}$ is the precomputed SINR threshold corresponding to the target BLER*.

Optimization Problem

At each time step, the association decision must satisfy the limitation on available PRBs, the cost induced by handovers, and the SINR threshold associated with the target BLER. Let $\rho_{u,c,t}$ denote the number of PRBs required to serve user u through cell c at time t. The total number of occupied PRBs in cell c is then

$$p_{\text{rb}}(c,t) = \sum_{u\in\mathcal{U}}\left(\mathbb{1}\{c_u(t) = c\}\rho_{u,c,t}\right), \qquad (8)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. Let $p_{\text{rb}}^{max}(c)$ note the maximum number of available PRBs in cell c.

The optimization problem is formulated as

$$\min_{\{c_u(t)\}} \max_{u\in\mathcal{U}} \left(lim\ sup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} f\left(g_{u,t}(c_u(t))\right)\right) \qquad (9)$$

$$\text{subject to} \quad \sum_{u\in\mathcal{U}} \mathbb{1}\{c_u(t) = c\}\rho_{u,c,t} \leq p_{\text{rb}}^{max}(c), \quad \forall c \in \mathcal{C}, \forall t$$

$$K_u(t)X_u(t) \leq W, \quad \forall u \in \mathcal{U}, \forall t$$

$$\gamma_{u,c_u(t),t} \geq \gamma^{\star}_{m_u(t),L_u^{max}(t)}, \quad \forall u \in \mathcal{U}, \forall t$$

$$c_u(t) \in \mathcal{C}, \quad \forall u \in \mathcal{U}, \forall t$$

The first constraint ensures that the total PRB demand of users associated with each cell does not exceed the available PRB budget of that cell. The second constraint limits frequent handovers. The third constraint ensures that the selected serving cell satisfies the minimum SINR required to meet the target BLER under the assumed MCS and HARQ configuration. The binary handover indicator is defined as

$$X_u(t) = \begin{cases} 1, & c_u(t) \neq c_u(t-1) \\ 0, & c_u(t) = c_u(t-1) \end{cases} \qquad (10)$$

The handover cost is modeled as

$$K_u(t) = k_0 e^{-\delta(t - t_{\text{ho}}(u))}, \qquad (11)$$

where $t_{\text{ho}}(u)$ is the time of the most recent handover of user u, $k_0 > 0$ is the initial handover cost, and $0 < \delta < 1$ controls the rate at which the handover cost decays with time. This model assigns a larger cost to handovers occurring shortly after a previous handover, thereby discouraging ping-pong behavior and excessive mobility overhead. The threshold W defines the maximum admissible handover cost.

Problem (9) is a mixed-integer nonlinear optimization problem. The integer nature comes from the discrete cell-association decision $c_u(t)$ and the binary handover indicator $X_u(t)$, while the nonlinearity arises from the M2P cost function, the coupling between cell association and PRB consumption, and the dependence of the estimated delay and SINR on time-varying RAN measurements. Therefore, instead of solving (9) directly using a closed-form model, we formulate the problem as a MDP and apply reinforcement learning (RL) to learn an association policy from the observed network states.

MDP Formulation

The MDP is defined by the tuple

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle, (12)$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the state transition probability, $\mathcal{R}$ is the reward function, and γ is the discount factor.

The state observed for user u at time t is denoted by $s_u(t) \in \mathcal{S}$. It contains the user identifier, the current serving cell, the candidate neighboring cells, and E2-observable RAN measurements. A representative state vector can be written as

$$s_u(t) = \left[u,\ c_u(t),\ \mathcal{C}'_u(t),\ \{\text{SINR}_{\text{u,c,t}}\}_{c \in \mathcal{C}'_u(t)},\ \{\text{PRB}_{\text{c,t}}\}_{c \in \mathcal{C}'_u(t)},\ g_{u,t}\big(c_u(t)\big),\ K_u(t)\right], \qquad (13)$$

where $\mathcal{C}'_u(t) \subseteq \mathcal{C}$ is the set of feasible serving and neighboring cells for user u at time t. The variables $\text{SINR}_{\text{u,c,t}}$ and $\text{PRB}_{\text{c,t}}$ denote, respectively, the reported SINR and PRB utilization of cell c at time t. Additional RAN measurements, such as throughput, buffer status, CQI/MCS, HARQ statistics, RLC statistics, and MAC-layer transport block statistics, can also be included when available.

The action space is defined as

$$\mathcal{A}_u(t) = \{\text{HO}(c) : c \in \mathcal{C}'_u(t)\} \cup \{\text{no-HO}\}. \quad (14)$$

The action HO(c) indicates that user u is handed over to cell c, while no-HO means that the user remains associated with its current serving cell.

The reward function is designed to encourage actions that reduce the estimated M2P cost while penalizing unnecessary handovers. A suitable reward for user u at time $t+1$ is

$$r_u(t+1) = f\left(g_{u,t}\left(c_u(t)\right)\right) - f\left(g_{u,t+1}\left(c_u(t+1)\right)\right) - \lambda K_u(t+1)X_u(t+1),$$

where $\lambda \geq 0$ controls the relative importance of the handover penalty. This reward is positive when the selected action reduces the M2P-related cost more than the cost induced by handover, and negative otherwise. Therefore, the learned policy aims to improve the user's perceived QoE while avoiding frequent cell changes.

The objective of the RL agent is to learn a policy $\pi(a|s)$ that maximizes the expected discounted cumulative reward:

$$\pi^{\star} = \arg\max_{\pi} E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{u\in\mathcal{U}} r_u(t)\right].$$

In implementation, constraints such as the PRB budget, handover-cost threshold, and SINR threshold can be handled through action masking, constraint-aware action selection, or large negative penalties for infeasible actions.

Mapping to 3GPP Service Availability and Reliability

This section maps the optimization objective to the 3GPP concepts of communication service availability and reliability. According to 3GPP TS 22.261, communication service availability is the percentage of time during which the end-to-end communication service is delivered according to the specified QoS, divided by the time during which the system is expected to deliver that service [155]. The same specification defines reliability, in the context of network-layer packet transmissions, as the percentage of transmitted packets successfully delivered to a given system entity within the time constraint required by the targeted service [155].

In the considered VR service, the main QoS requirement is that the M2P delay remains below a target threshold. The M2P-based service-availability indicator for user u at time t is defined as

$$A_{u,t}^{\mathrm{M2P}} = \mathbb{1}\{g_{u,t}\left(c_u(t)\right) \leq \mathrm{M2P}_{\max}\}. \quad (17)$$

The corresponding long-term M2P service availability of user u is

$$\mathcal{A}_u^{\mathrm{M2P}} = lim\ inf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} A_{u,t}^{\mathrm{M2P}}. \quad (18)$$

Thus, the objective in (9) can be interpreted as a soft service-unavailability minimization. In particular, if the cost function is chosen as

$$f(g) = \mathbb{1}\{g > \mathrm{M2P}_{\max}\}, \qquad (19)$$

then the objective in (9) becomes the minimization of the worst-user long-term M2P service unavailability:

$$\min_{\{c_u(t)\}} max_{u\in\mathcal{U}}\left(1 - \mathcal{A}_u^{\mathrm{MP}}\right) \quad (20)$$

Then a smooth increasing cost function is used instead of (19), the objective should be interpreted as a smooth surrogate for service-unavailability minimization, where larger M2P violations receive larger penalties.

The RAN-side reliability requirement is captured through the SINR constraint in (6). Since the threshold $\gamma^{\star}_{m,L}$ is chosen such that the residual BLER after at most L HARQ transmissions satisfies BLER*, satisfying the SINR constraint implies that the selected radio link meets the target BLER condition. Therefore, a RAN-side reliability indicator can be defined as

$$B_{u,t}^{\mathrm{RAN}} = \mathbb{1}\left\{\gamma_{u,c_u(t),t} \geq \gamma^{\star}_{m_u(t),L_u^{max}(t)}\right\}. \quad (21)$$

If one VR update, frame, or scheduling decision is associated with each time step, the long-term RAN-side reliability of user u can be approximated as

$$\mathcal{R}_u^{\mathrm{RAN}} = lim\ inf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} B_{u,t}^{\mathrm{RAN}}. \quad (22)$$

This definition is consistent with the 3GPP interpretation of service availability as the fraction of time during which the service is delivered according to the required QoS. Here the QoS condition is represented by the joint satisfaction of the M2P deadline and the radio-link BLER requirement. It should be noted that $\mathcal{R}_u^{\mathrm{RAN}}$ is a RAN-side reliability approximation, not a full end-to-end service reliability metric, because it does not explicitly include packet losses or failures occurring outside the RAN, such as core-network drops, application-layer errors, decoding failures, or display-side failures.

5.5.2 Preliminary Results

As a preliminary step, we evaluate the behavior of the proposed M2P-aware cost function before performing full system-level simulations. The goal of this analysis is not to quantify the final gain of the proposed scheduler, but to verify that the optimization objective correctly reflects the QoE degradation associated with excessive M2P delay.

Let

$$\Delta_{u,t} = g_{u,t}(c_u(t)) \qquad (23)$$

denote the estimated M2P delay experienced by user u at time t under the selected serving cell $c_u(t)$. We define the M2P cost as

$$\mathrm{f}(\Delta_{\mathrm{u,t}}) = \begin{cases} 0, & \Delta_{u,t} \le a, \\ \eta\left((\Delta_{u,t} - a)/(\mathrm{M2P_{max}} - a)\right)^2, & a < \Delta_{u,t} \le \mathrm{M2P_{max}}, \\ \eta + \mu\left(e^{\kappa(\Delta_{u,t} - \mathrm{M2P_{max}})} - 1\right), & \Delta_{u,t} > \mathrm{M2P_{max}}, \end{cases} \quad (24)$$

where a is the delay value below which the user experience is considered fully acceptable, $\mathrm{M2P_{max}}$ is the M2P threshold beyond which discomfort may occur, and η, μ, and κ control the slope and severity of the penalty. This design creates three QoE regions. For $\Delta_{u,t} \le a$, the cost is zero, meaning that the scheduler does not distinguish between users whose M2P delay is already comfortably below the target. For $a < \Delta_{u,t} \le \mathrm{M2P_{max}}$, the cost increases smoothly, encouraging the scheduler to act before the M2P delay reaches the critical threshold. For $\Delta_{u,t} > \mathrm{M2P_{max}}$, the cost grows exponentially, strongly penalizing associations that lead to potentially uncomfortable VR experiences.

This cost function is suitable for the proposed user-centric association problem because it does not only minimize the average delay but also prioritizes users approaching or exceeding the M2P comfort limit. Therefore, two candidate serving cells with similar resource consumption may be ranked differently if one of them moves the user closer to the M2P violation region. In this way, the cost function transforms the estimated delay into a QoE-aware scheduling signal.

The preliminary behavior of (24) confirms that the objective in (9) acts as a soft service-unavailability minimization. When the estimated M2P delay remains below the acceptable region, the contribution to the objective is negligible. As the delay approaches $\mathrm{M2P_{max}}$, the cost increases, and after the threshold it rises sharply. This behavior is consistent with the intended design of the scheduler: users whose predicted M2P delay is close to or above the comfort threshold are given higher priority in the association decision. In the next stage, the cost function in (24) will be used to compare the proposed M2P-aware association policy against baseline policies such as strongest-SINR association, load-aware association, and handover-threshold-based association.

5.6 THE SEM-NTN FRAMEWORK

5.6.1 Algorithm description

The proposed AI algorithm is part of the SEM-NTN (Semantic Non-Terrestrial Network) framework, which introduces a semantic-aware communication paradigm integrated with O-

RAN architecture and NTN as given in

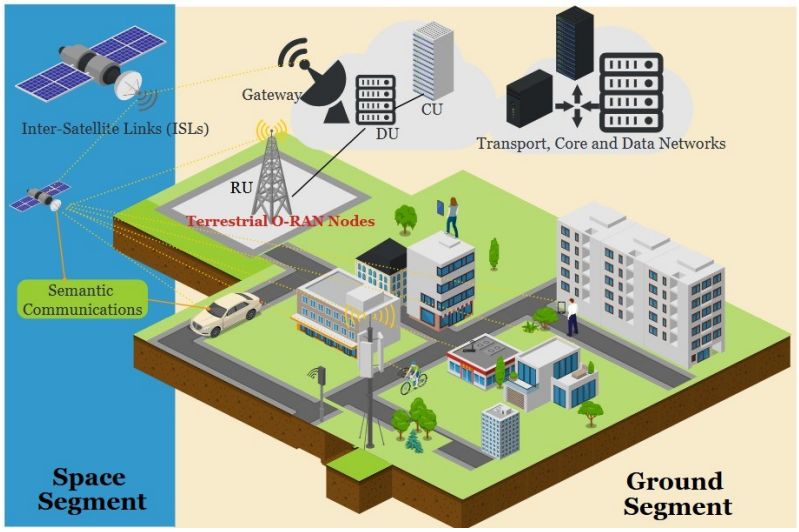


Figure 69.

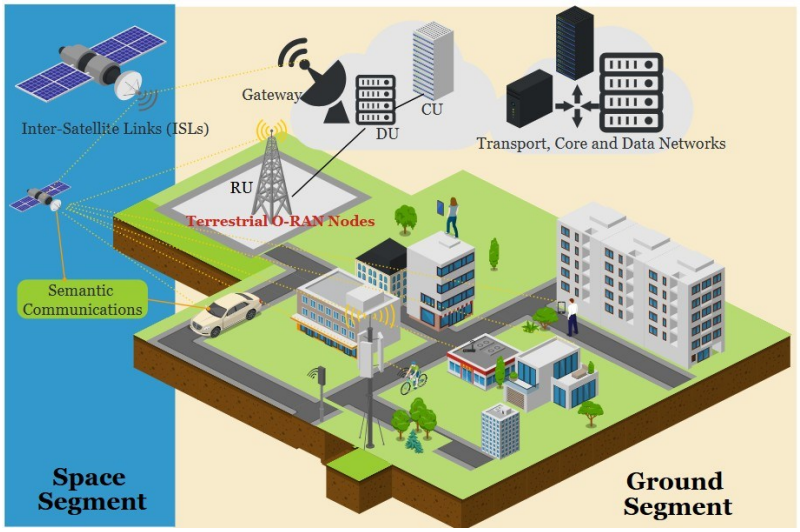


Figure 69: General illustration of semantic communication within the SEM-NTN system where TN and NTN components are natively integrated within 6G networks.

Unlike traditional communication systems that focus on transmitting raw bits, the proposed approach leverages artificial intelligence to extract and transmit only meaningful information. This significantly reduces communication overhead while maintaining the performance of downstream AI tasks. The framework combines semantic feature extraction, adaptive compression, and AI-driven decision-making to enable efficient and low-latency communication in 6G satellite-terrestrial integrated networks.

The algorithm operates through a closed-loop AI-driven pipeline consisting of four main components: the Semantic-Aware Monitoring System (SA-MS), the Semantic-Aware Analytics Engine (SA-AE), the Semantic-Aware Decision Engine (SA-DE), and the Actuator (ACT).

These components are tightly integrated with O-RAN's Near-Real-Time and Non-Real-Time RICs, enabling both real-time adaptation and long-term optimization as given in Figure 70.

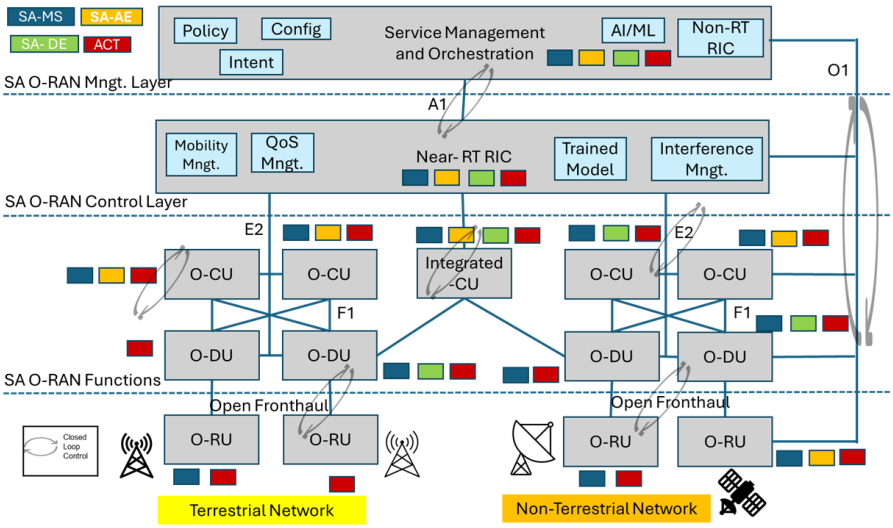


Figure 70: Semantic-aware closed loop management and control with tuples (SA-MS, SA-AE, SA-DE and ACT) in integrated NTNs and TNs within SEM-NTN framework

Algorithm Workflow

1. Semantic Feature Extraction (SA-MS): The SA-MS is responsible for collecting network and data information and performing semantic feature extraction. Instead of forwarding raw data, it identifies and retains only the most relevant contextual information. In the presented work, semantic importance is modeled using a semantic complexity variable, which captures the richness of the data content. Although this is simulated as a random variable, in real deployments it can be derived from lightweight AI models such as object detection or scene understanding.

2. AI-driven Semantic Analytics (SA-AE): Following this, the SA-AE processes the extracted semantic features using machine learning and deep learning models. This module performs higher-level analytics, such as identifying traffic patterns, predicting mobility events, and estimating the importance of data for transmission. By focusing only on meaningful features, the system significantly reduces unnecessary data transmission, improves spectrum efficiency, and minimizes computational overhead. This is particularly important in NTN environments where bandwidth is limited and latency constraints are strict.

3. Adaptive Compression Decision (SA-DE): The core intelligence of the algorithm lies in the SA-DE, which dynamically determines the appropriate compression level based on semantic complexity, available link capacity, and delay constraints. The compression factor is adaptively selected such that semantically important data is preserved while less relevant information is aggressively compressed. The decision process ensures that the maximum transmission delay

constraint is satisfied, enabling real-time communication. If the network conditions are constrained, the algorithm enforces stronger compression to meet delay requirements; conversely, when sufficient bandwidth is available, it allows higher-quality transmission to improve AI inference accuracy. This adaptive behavior enables a fine-grained trade-off between communication efficiency and application performance.

4. Closed-loop Network Optimization (ACT + RIC Integration): Finally, the ACT component applies the decisions by reconfiguring network elements such as the O-CU, O-DU, and O-RU, adjusting bandwidth allocation, and optimizing interference management strategies. This completes the closed-loop system, where continuous feedback from the network allows the algorithm to refine its decisions over time. The integration with O-RAN RICs ensures that these optimizations can be performed both at near-real-time and non-real-time scales, aligning with the principles of AI-native network orchestration.

From an operational perspective, the algorithm can be understood as solving a constrained optimization problem that aims to minimize transmission delay while maintaining acceptable AI inference accuracy. The key decision variable is the compression factor, which is dynamically adjusted within a predefined range. The effective compression factor is determined as the minimum of the semantically desired compression and the maximum allowable compression under delay constraints. This ensures that the system remains within latency limits while maximizing the usefulness of transmitted data.

The proposed approach goes significantly beyond the state-of-the-art in several aspects. First, it extends semantic communication concepts beyond terrestrial networks by integrating them into NTN environments, addressing unique challenges such as satellite latency and limited bandwidth. Second, it introduces a fully closed-loop AI-driven architecture, whereas existing works typically focus on isolated components such as compression or resource allocation. Third, the algorithm enables semantic-aware adaptive compression, which dynamically adjusts data representation based on content relevance rather than applying uniform compression. Fourth, it jointly optimizes communication performance and AI inference accuracy, unlike traditional systems that treat these aspects separately. Finally, it incorporates semantic awareness into O-RAN-based resource allocation and slicing, enabling intelligent and context-aware network management. These contributions collectively position the proposed algorithm as a key enabler for AI-native 6G networks.

5.6.2 Preliminary results

Preliminary evaluations of the proposed SEM-NTN framework demonstrate the potential of semantic-aware communications for integrated terrestrial and non-terrestrial 6G networks

[156]. The experiments were conducted using CIFAR-10 image datasets combined with AI inference models including YOLOX for object detection [157] and BiSeNet V2 for semantic segmentation [158]. The goal was to analyze the trade-off between transmission latency and AI inference accuracy under varying satellite link capacities and semantic-aware compression levels.

The simulations considered LEO satellite links with capacities ranging from 100 Mbps to 1 Gbps. Semantic-aware adaptive compression was compared against conventional uniform detailed compression. In the semantic-aware approach, the compression factor dynamically changes according to the semantic relevance and complexity of the transmitted content. Images with lower semantic complexity are compressed more aggressively, while semantically rich content preserves higher detail to maintain inference quality.

The obtained results in Figure 71 show significant improvements in transmission efficiency. At lower satellite bandwidths, semantic-aware communication considerably reduces transmission delay compared to uniform compression. For example, at a 100 Mbps satellite link, semantic-aware compression achieves approximately 0.001 s delay, whereas conventional uniform compression introduces nearly 0.008 s delay. This corresponds to nearly 87.5% reduction in transmission latency. Similar gains are maintained across different bandwidth levels, demonstrating the suitability of semantic communication for bandwidth-constrained NTN environments. Note that in this analysis, we have ignored the transmission delay but relative gains of semantic-aware compression are expected to be valid due to common transmission delays in both approaches, i.e., semantic and non-semantic.

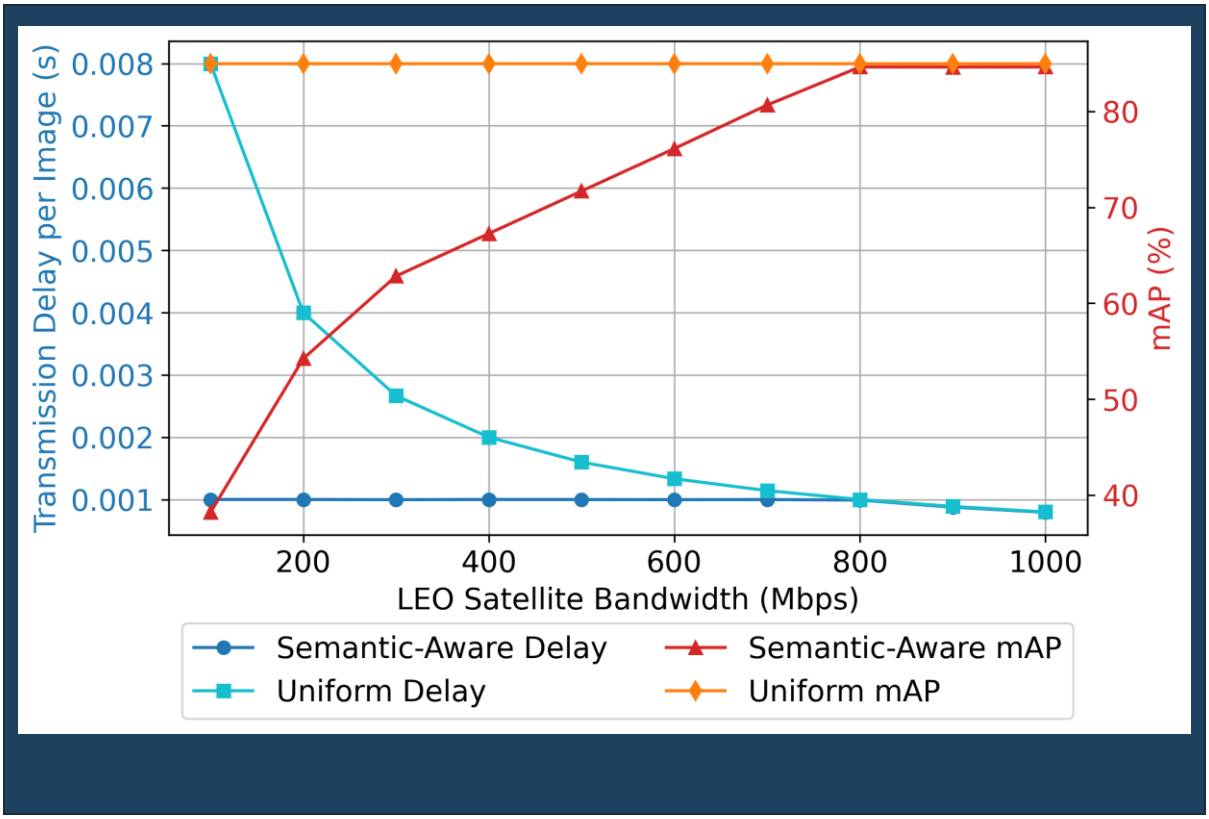


Figure 71: Comparison of transmission delay with the capacity of LEO satellite links (using CIFAR-10) for semantic and uniform compression

The results also highlight the adaptive behavior of SEM-NTN under varying network conditions as in Figure 72. While aggressive semantic compression reduces delay, it also impacts AI inference accuracy. At low bandwidths, object detection accuracy measured through mAP remains around 38–60%, depending on the compression level. However, as the available bandwidth increases, the system gradually relaxes compression constraints, enabling higher semantic fidelity. At bandwidths close to 1 Gbps, the semantic-aware approach achieves approximately 85% mAP, effectively matching the performance of conventional uncompressed transmission.

Additional experiments analyzed the relationship between the maximum allowable transmission delay and inference performance. Under a fixed satellite bandwidth of 500 Mbps, tighter latency constraints force the system to apply stronger semantic compression. At a maximum delay constraint of 0.5 ms, the achieved mAP is approximately 60.5%, while the segmentation accuracy reaches around 55.6% mIoU. When the allowable delay increases to 2–5 ms, the semantic-aware framework achieves approximately 84.6% mAP and 77.6% mIoU, nearly equivalent to full-quality transmission. These results confirm that SEM-NTN can intelligently adapt semantic compression levels to satisfy latency constraints while maintaining high AI inference performance.

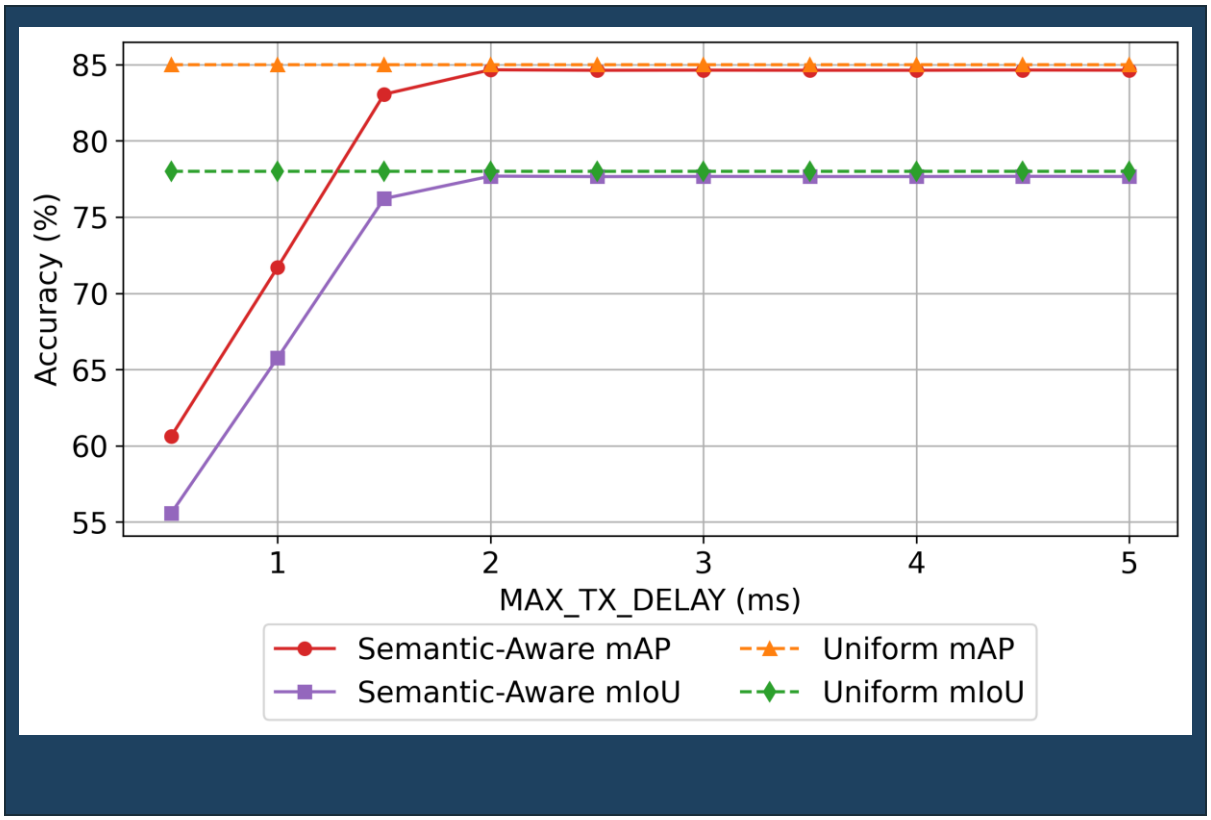


Figure 72: The accuracy metrics versus MAX_TX_DELAY for fixed satellite bandwidth of 500 Mbps.

The preliminary results further validate the effectiveness of semantic-aware O-RAN integration. By embedding semantic processing and AI-driven optimization into the RICs, the framework supports dynamic resource allocation, semantic-aware slicing, and intelligent NTN/TN switching. The semantic-aware tuples, including SA-MS, SA-AE, SA-DE, and ACT, enable closed-loop optimization where network monitoring, analytics, decision-making, and actuation continuously adapt to changing network conditions and application requirements.

In summary, the preliminary evaluations demonstrate that SEM-NTN provides an efficient and scalable solution for future AI-native 6G integrated networks. The framework successfully balances latency, bandwidth efficiency, and AI inference accuracy, making it highly suitable for real-time, bandwidth-constrained, and mission-critical applications operating across terrestrial and satellite domains.

6 CONFLICT RESOLUTION FOR INTEGRATED COMPONENTS

6.1 INTRODUCTION

Future 6G systems will be characterized by strong heterogeneity (terrestrial and non-terrestrial segments, Open RAN, edge–cloud continuum) and by an AI-native operational paradigm, where multiple intelligent controllers act concurrently on shared resources. In such environments, conflicts among integrated components become a first-order problem: parallel control loops may compete for the same radio/compute resources, enforce incompatible policies, or trigger oscillations that degrade stability, QoS and energy efficiency. The purpose of this chapter is to present the UNITY-6G approach to conflict resolution as a combination of architectural mechanisms and domain-specific algorithms that enable safe and scalable coordination across integrated components.

A key contribution is the explicit recognition that conflicts manifest at different scopes and timescales, ranging from local clashes within a single domain to cross-domain inconsistencies emerging at higher orchestration levels. In particular, the chapter adopts a multi-level conflict view (e.g., intra-domain vs. inter-domain vs. orchestrator-level conflicts) and couples it with resolution strategies that can be enforced either locally (through domain KPIs/policies) or globally (via negotiation and coordination mechanisms). This layered approach is aligned with the need to support autonomous domains while preserving end-to-end coherence in an integrated architecture.

Within this framework, the chapter presents a set of algorithms that collectively cover complementary aspects of conflict handling:

1. **O-RAN Domain conflict management**:

- Contribution to the *architectural foundation* for conflict handling in O-RAN, leveraging policy/knowledge components and an interaction fabric that supports both local mitigation and cross-domain escalation.
- Introduction to COMIX, a conflict resolution algorithm targeting O-RAN xApp/rApp ecosystems, where multiple apps concurrently optimize different (and potentially competing) KPIs. COMIX focuses on detecting and resolving conflicts using an explicit conflict model and decision logic that can be integrated into a closed-loop pipeline.
- Extension of COMIX with a distributed/decentralized approach, leveraging mechanisms such as smart-contract-like coordination and event-driven updates to improve scalability,

auditability, and resilience when centralized coordination is partially unavailable or undesirable.

2. **Predictive conflict avoidance via probabilistic forecasting:** Conflicts often arise because resource contention is detected "too late". To anticipate contention, the chapter includes a probabilistic forecasting approach for resource allocation (e.g., in O-RAN contexts), integrating models such as GPVAR and Temporal Fusion Transformers (TFT) (benchmarked against multivariate LSTMs) in a cloud-native, containerized rApp pipeline. By producing *uncertainty-aware forecasts* rather than point predictions, this family of methods enables proactive decisions that reduce the probability of policy clashes and last-moment contention.

3. **Conflict-aware optimization in wireless routing:** In integrated wireless systems, "conflicts" also emerge as interference coupling: the routing choice for one flow changes the feasible rates of others. The DPIR framework addresses this by combining predictive traffic modeling, graph-native topology encoding, path-level representations and reinforcement-learning-based route selection to optimize network utility under interference coupling. This reframes routing as a coordinated, conflict-aware decision problem rather than a set of independent shortest-path selections. This algorithm is introduced in Section 3.2.2.4.

Overall, the chapter positions conflict resolution not as a single algorithm, but as an AI-native, multi-domain capability: conflicts are detected via monitoring and analytics, resolved through policy/learning/negotiation, and enforced via actuators in closed loops - consistent with the overarching UNITY-6G control principles.

The conflict resolution algorithms will be implemented in the DMO of the UNITY-6G architecture.

6.2 STATE OF THE ART

Below we categorize conflict-resolution-related approaches and summarize the current SoA for each category, highlighting the main trends and open gaps that motivate the UNITY-6G contributions presented in this chapter.

6.2.1 O-RAN xApp/rApp conflict management

In O-RAN ecosystems, multiple xApps/rApps may concurrently optimize different objectives (throughput, latency, energy, fairness, slice KPIs), leading to policy collisions and instability. The SoA includes:

- rule-based conflict detection/mitigation in near-RT RIC settings
- SMO/non-RT RIC mechanisms for more proactive detection
- modular monitoring pipelines that track app actions and resulting KPI deviations.

Recent research moves toward learning-based conflict identification (e.g., dynamic conflict graphs and data-driven detection) and towards cooperative optimization (team learning / multi-agent formulations) to align app behaviours. These directions aim to reduce oscillations and improve robustness under changing traffic and topology.

Conflict detection and resolution in the O-RAN domain have attracted considerable attention, with numerous studies examining the problem from both architectural and algorithmic perspectives [159]. Regarding the former, the authors in [160] introduced an architectural framework for conflict mitigation within the Near-RT RIC, leveraging predefined rules and real-time monitoring to detect and resolve conflicts in alignment with O-RAN's modular design principles. Another study proposed a proactive conflict detection mechanism integrated into the SMO framework, aiming to identify potential conflicts early and prevent performance degradation through continuous monitoring of multi-xApp interactions [161]. Considering the algorithmic advancements for conflict mitigation in O-RAN, the authors in [162] presented a mobility-aware energy optimization strategy to handle conflicts at the xApp level, particularly those caused by competing objectives such as energy efficiency and mobility management. Another line of work proposed a data-driven, learning-based method for constructing and labelling conflict graphs, enabling the system to dynamically infer hidden relationships among xApps, parameters, and KPIs [163]. Finally, the authors in [165] introduced a team-based learning framework designed to encourage cooperative decision-making among xApps, thereby reducing indirect and implicit conflicts during resource allocation. Overall, although these studies offer valuable contributions toward conflict detection and resolution in O-RAN, there remains a clear gap in the literature regarding use case-driven validation and proof-of-concept implementations. In particular, few works provide quantitative comparisons of different conflict resolution strategies under realistic network conditions.

Despite progress, gaps remain in realistic, use-case-driven validation; consistent quantitative comparisons among methods; and integration patterns that scale across multiple vendors/apps while preserving accountability and safe enforcement.

6.2.2 Cross-domain / orchestrator-level conflict resolution

Beyond single-domain xApp conflicts, integrated 6G architectures introduce cross-domain contention (radio vs. compute vs. transport), and conflicts between domain autonomy and end-to-end intents. The SoA is increasingly architectural: it relies on layered coordination where

local controllers resolve local conflicts, while higher-level orchestrators arbitrate end-to-end constraints using policy and negotiation.

Modern approaches rely on hierarchical and federated coordination mechanisms, including policy-based arbitration and message-based exchange of constraints among domain orchestrators, to resolve escalated conflicts when local resolution is insufficient [167][194].

The major open issues are scalability (number of domains and decision loops), convergence guarantees under partial observability, and the need for mechanisms that remain operational even when a central coordinator is unavailable or intermittently reachable.

6.2.3 Distributed and decentralized conflict resolution

A growing strand of SoA recognizes that conflict resolution in open, multi-stakeholder networks requires auditability and sometimes decentralized governance [195]. This motivates event-driven and decentralized coordination patterns, where conflict states and resolutions are logged and verifiable inside secure DLT networks.

Distributed variants of conflict resolution mechanisms (including smart-contract-like coordination and event emission) are being explored to support scalability and robustness especially relevant when coordination must persist under partial connectivity or administrative boundaries.

Trade-offs remain between decentralization and reaction time, as well as the complexity of maintaining consistent views of conflict state across multiple controllers and domains.

6.2.4 Predictive conflict avoidance via forecasting

6G is expected to operate as a federation of administrative domains rather than a single-operator system. In this setting, an IDMO coordinates multiple DMOs, each responsible for its own resources, policies, and service-level objectives. Locally optimal decisions taken by individual DMOs frequently collide with one another or with IDMO-level policies, producing inter-domain conflicts over resource allocation, SLA commitments, routing, or policy enforcement. ETSI ZSM has identified conflict detection and resolution between interacting closed loops as one of the core unresolved problems of cross-domain automation [168].

Because the DMOs sit in different administrative domains, there is no *a priori* reason for one domain to accept that the IDMO's resolution is fair, nor that the same resolution logic has been applied consistently across cases. Inter-domain conflict management must therefore address two distinct requirements at once: a *decision-making* requirement producing a valid resolution

and a *trust* requirement making that resolution verifiable and auditable across administrative boundaries.

Blockchain and smart contracts are a natural fit for the trust requirement, and recent work has been used extensively to anchor SLA monitoring, breach detection, and penalty enforcement across providers in 5G/6G networks [169] [170]. However, the smart contracts used in these approaches embed hardcoded resolution logic written against a small, pre-specified set of known agreement types. This assumption does not survive contact with 6G: domain types, resource combinations, SLA structures, and policy contexts are heterogeneous and evolve over the system's lifetime, and conflict scenarios cannot all be anticipated at design time. Writing a new smart contract for each novel conflict scenario does not scale as the number of domains grows and the conflict space evolves.

Two complementary lines of work have attempted to make contract generation more adaptive. First, LLM-based approaches generate Solidity directly from natural-language specifications [171]. These provide flexibility but offer no verifiability rails: recent evaluations show that LLM-generated smart contracts frequently exhibit severe security flaws even when syntactically correct Second, domain-specific languages (DSLs) for contract specification, such as Symboleo and its code-generation toolchain Symboleo2SC, provide formal grammars and verifiable to Solidity or Hyperledger Fabric [172]. These provide structure and verifiability but require human authorship for every new case.

Neither line of work is sufficient for an inter-domain orchestrator that must *autonomously* generate, verify, and deploy resolution contracts as novel conflicts arise at runtime.

Many conflicts arise from reactive management: contention is detected after it occurs. The SoA therefore increasingly uses forecasting to anticipate demand and prevent last-moment clashes. In O-RAN and cloud-native pipelines, multivariate forecasting is used to predict resource utilization and to inform proactive scheduling/allocation.

The state of practice is shifting from point forecasts to probabilistic forecasting, which provides confidence intervals and better supports risk-aware decisions under uncertainty. Approaches integrating advanced models (e.g., GPVAR, TFT) and benchmarking against LSTM-based baselines are representative of this trend, often deployed as containerized services for integration into management loops.

Open issues include robustness under distribution shifts, explainability of probabilistic decisions to operators, and mapping predictive outputs into enforceable policies that do not conflict with other controllers' objectives.

6.2.5 Conflict-aware routing and interference-coupled optimization

In wireless networks, conflicts materialize as interference coupling: the performance of one flow depends on the joint set of simultaneously active links [197]. Traditional routing methods often ignore this coupling or treat it indirectly via additive costs.

The SotA increasingly combines traffic prediction, graph-based representations of topology/link state, and reinforcement learning to select route profiles that maximize global utility under interference constraints. DPIR-like designs represent this direction by explicitly modelling coupling and selecting coordinated route profiles rather than independent per-flow shortest paths.

Key gaps include scalability of the joint action space, stability under rapidly varying channel conditions, and integration of routing decisions with other domain controllers (RAN scheduling, compute placement), which is essential in integrated 6G deployments.

6.2.6 Agentic/AI-native closed-loop conflict handling

A very recent direction is to embed conflict resolution into agentic control loops, where reasoning components can select tools (forecasting, anomaly detection, optimization) and use memory of past conflicts to improve decisions. This is particularly relevant in integrated settings, where conflicts are multi-causal and policies evolve [167] [198].

The SoA still lacks standardized evaluation protocols, shared datasets for conflict scenarios, and operational guardrails for safe, deterministic enforcement in real networks.

6.3 O-RAN DOMAIN ALGORITHM

6.3.1 Layered Conflict Management

In the proposed concept of a unified 6G network architecture based on the O-RAN vision [173], each domain within the architecture is treated as an independent and autonomous entity, despite relying on the same hardware resources. In contrast to standard solutions [174][175] used in O-RAN, such an architecture (Figure 73) introduces the challenge of implementing multi-domain, multi-agent conflict management both within individual domains and across domains, under the supervision of the IDMO.

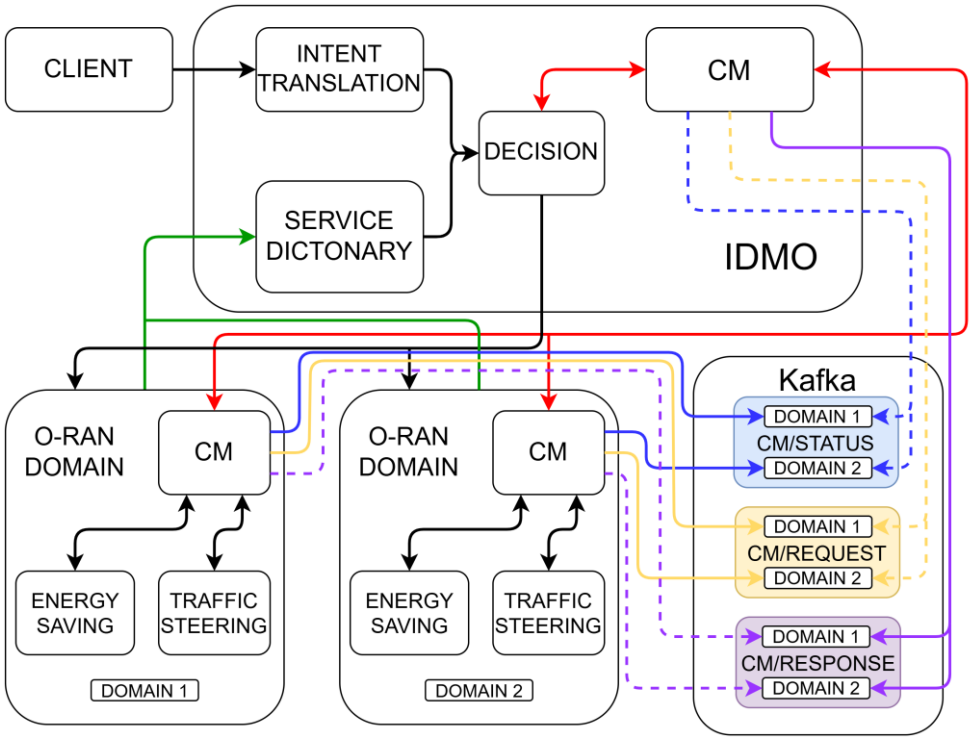
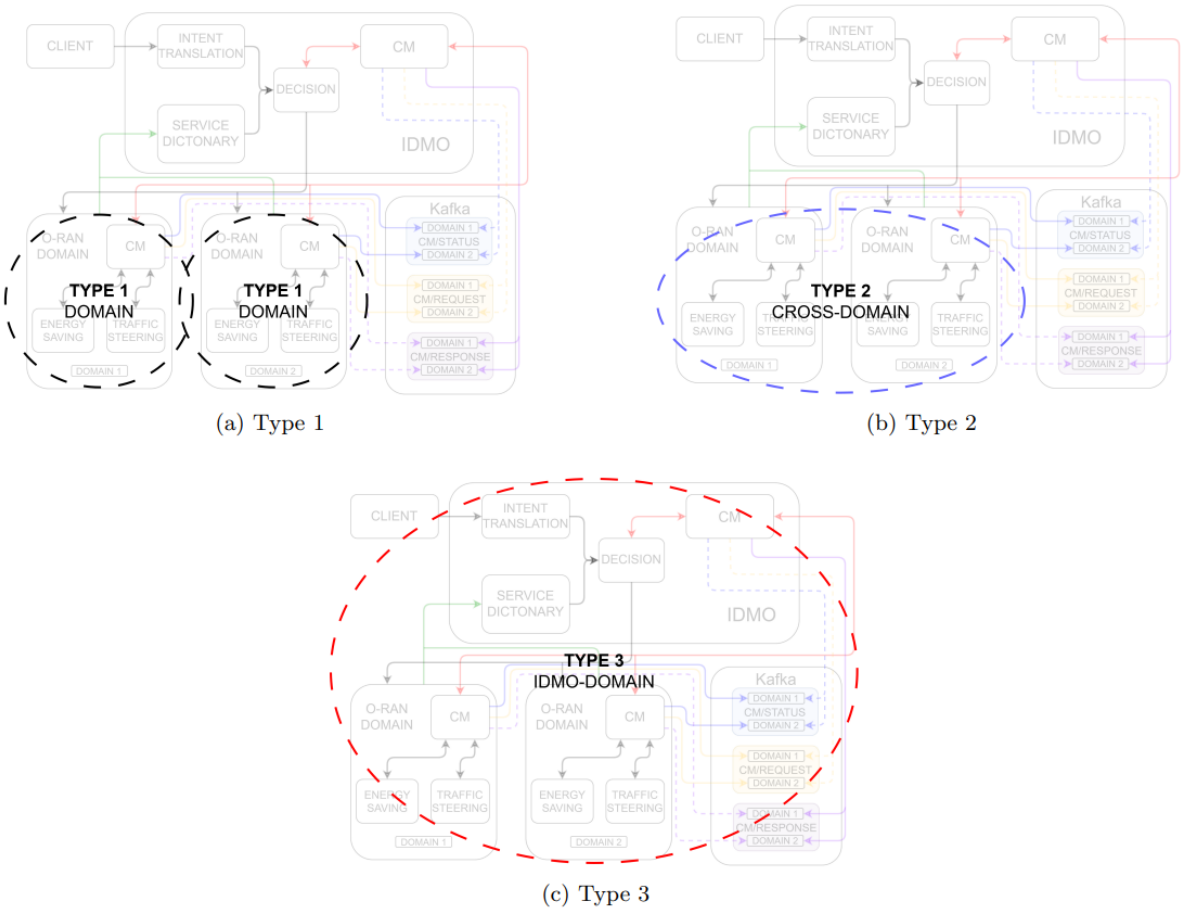


Figure 73: System architecture

This approach requires the use of a local conflict resolution module in each existing domain. A single target domain will include an individual set of xApps provided by different vendors, whose decisions may modify the same parameters and thus lead to conflicts. We define such conflicts as Type 1 conflicts (Figure 74), i.e., intra-domain conflicts.

Since all domains operate on the same hardware resources, changes introduced by xApps within one domain may cause conflicts across multiple domains. This category is defined as Type 2 conflicts (Figure 74). The final category is Type 3 conflicts (Figure 74), which arise from IDMO decisions, such as the activation of additional services. Solving this type of conflict requires communication and negotiation between the module in the DMO and the IDMO.



(a) Type 1

(b) Type 2

(c) Type 3

Figure 74: Areas of occurrence of types of conflicts

For this purpose, a Kafka broker is used. Each module from DMO or IDMO is subscribed to the poper topic. The STATUS topic stores information on currently detected conflicts and their types. The REQUEST/RESPONSE topics are used when negotiation between the DMO and the IDMO is initiated to determine parameters that allow a Type 3 conflict to be avoided or resolved. In the case of excessive network load during an attempt to launch additional services, negotiation is performed to reduce the requirements for service processing or to stop the service activation process due to the current network conditions. The methods for resolving conflicts of each type are presented in Table 12.

Table 12: Conflict types

	Conflict level	**Resolution module**	**Decision**
Type 1	Intra-domain	Domain level	Based on network parameters and KPIs
Type 2	Inter-domain	Domain-IDMO level	Based on network parameters and KPIs
Type 3	IDMO-domain	IDMO level	Based on negotiation between IDMO and DMO

6.3.2 Conflict Resolution Framework for O-RAN

We here introduce a CMF for O-RAN xApps, designed to handle conflicting RU-level objectives, though this can be generalized to the rest of the O-RAN components. Unlike conventional CMF approaches that rely on reactive or rule-based methods, the proposed scheme incorporates a Network Digital Twin (NDT) to proactively simulate the impact of competing xApp actions. This enables more accurate and reliable conflict resolution while reducing risks to live network operation. Aligned with O-RAN CMF principles, the framework adopts a generalizable conflict classification approach, focusing on resolving conflicts between xApps in practical scenarios, improving decision accuracy and minimizing network disruption.

The considered architecture of the O-RAN environment is shown in Figure 75. The SMO hosts the non-RT RIC, which operates on long timescales to provide high-level guidance to the near-RT RIC via the A1 interface. To this end, intent-based policies (e.g., throughput, energy, SLA targets) are transmitted over A1 to guide conflict resolution in the near-RT RIC, ensuring alignment between strategic objectives and real-time control. They may also enforce constraints (e.g., energy or data rate thresholds) that are incorporated into the Conflict Resolver as resolution policies. The near-RT RIC is the architectural entity that enables real-time optimization and conflict management through several core components:

- **xApp Catalog & Active xApps:** A repository of vendor-specific xApps that can be instantiated based on network needs. Active xApps (e.g., throughput-maximization or energy-efficiency xApps) generate control actions using AI/ML techniques, which are evaluated for conflicts.
- **RIC Database & Databus:** The database stores RAN metrics and historical data collected via the E2SM interface, while the databus facilitates data exchange among xApps and the CMF.
- **CMF:** Responsible for detecting and resolving conflicts among xApps. The Conflict Detector identifies conflict types based on parameter interactions and historical data, while the Conflict Resolver applies policies (e.g., priority-, SLA-, or fairness-based) to select optimal actions. It integrates a Network Digital Twin (NDT) to simulate and evaluate candidate actions before deployment, enabling proactive decision-making.
- **NDT:** A virtual replica of the network used to predict the impact of xApp actions on KPIs (e.g., energy efficiency, throughput). By validating actions prior to execution, it reduces the risk of performance degradation compared to reactive approaches.

- **RAN Environment:** Includes E2-connected nodes (RUs, DUs, CUs) and network dynamics such as traffic, mobility, interference, and resource allocation. The near-RT RIC enforces decisions via E2SM Control messages and receives feedback through E2SM Reports.

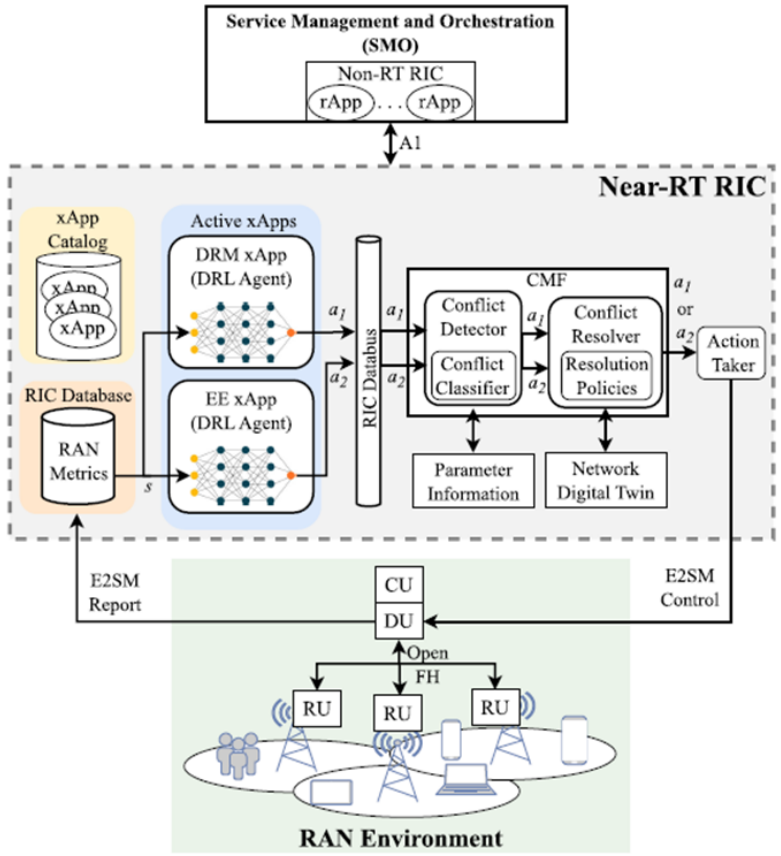


Figure 75: General O-RAN architecture considered for the conflict detection and resolution framework [160]

The Generalized Conflict Management scheme for the O-RAN xApps developed in the framework of the Unity-6G project is shown in Figure 76. It aims to identify direct three types of conflicts, namely the direct (DC), indirect (IC), and implicit conflicts among xApps. DCs and ICs are detected proactively, while implicit conflicts are identified reactively through KPI degradation. The Conflict Detector (CD) processes decisions from active xApps, each producing control parameters (CPs) to optimize specific KPIs. Since CPs may affect multiple KPIs across xApps, the MNO defines the CP–KPI relationships a priori. After inference, all xApp decisions are forwarded to the CD via the RIC Databus. The CD consists of the following modules:

- **CP/KPI Extraction:** Identifies CPs and affected KPIs for each xApp from the global CP set and KPI set.
- **DC Checker:** Detects direct conflicts by identifying xApps controlling common CPs. Conflicting decisions are forwarded for resolution, while CP/KPI data proceed for further analysis.

- **CP Clustering:** Groups CPs by the KPIs they influence using an association matrix, forming KPI-based clusters.
- **IC Checker:** Identifies indirect conflicts by detecting xApps whose CPs belong to the same KPI cluster. Conflicting decisions are sent to the Conflict Resolver; otherwise, they proceed to execution.
- **Action Forwarder:** Routes decisions either to the Conflict Resolver (if conflicts exist) or to the Action Taker.
- **KPI Notifier:** Monitors KPI performance and detects implicit conflicts via unexpected degradation. When triggered, it updates CP–KPI clusters to reflect newly observed dependencies.

CP clusters are dynamically updated when new xApps are introduced, new decisions are generated, or KPI degradation reveals previously unknown CP–KPI relationships.

An Illustrative example is also shown in Figure 76. Considering four xApps with CP set and KPI set, a DC is detected between two xApps controlling the same CP (Figure 76 b). Moreover, clustering then reveals ICs where different CPs impact the same KPI (Figure 76 c). Later, KPI degradation may expose hidden dependencies, leading to the identification of implicit conflicts as ICs over time. Figure 76 d depicts the overall conflict graph among the four xApps, including the direct, indirect and implicit conflicts.

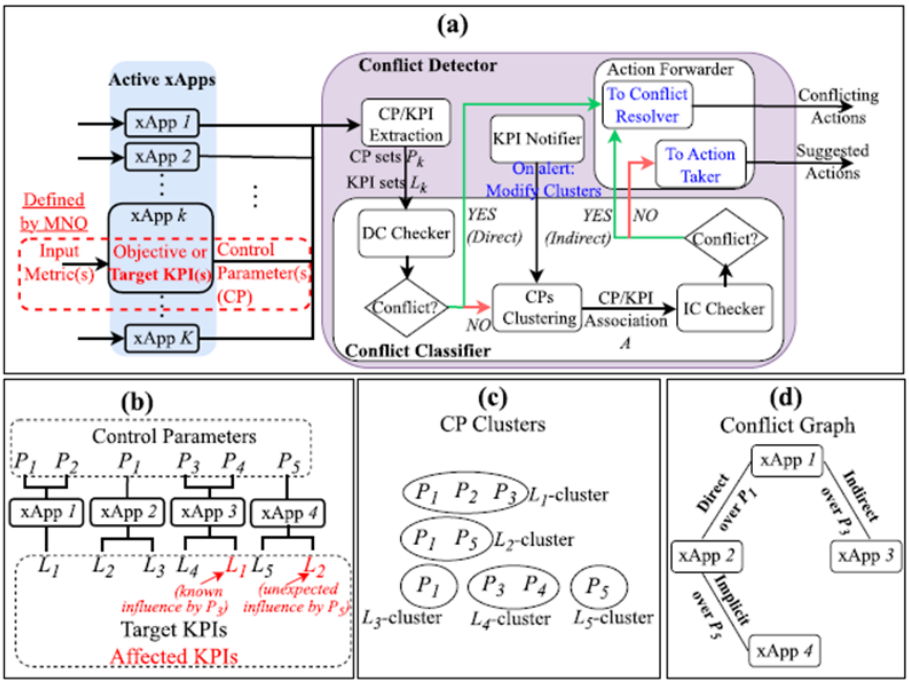


Figure 76: Generalized Direct/Indirect Conflict Detector. (a) Internal Architecture; (b) Example of CPs/KPIs associated with 4 xApps; (c) CP Clusters per KPI; (d) Conflict Graph [175].

6.3.3 Distributed Approach to COMIX

Starting from the COMIX paper [175], an analysis has been performed to understand which parts of the conflict resolution algorithm can be moved in a Smart Contract (SC) in order to distribute and decentralize it at least in some parts to make it more resilient.

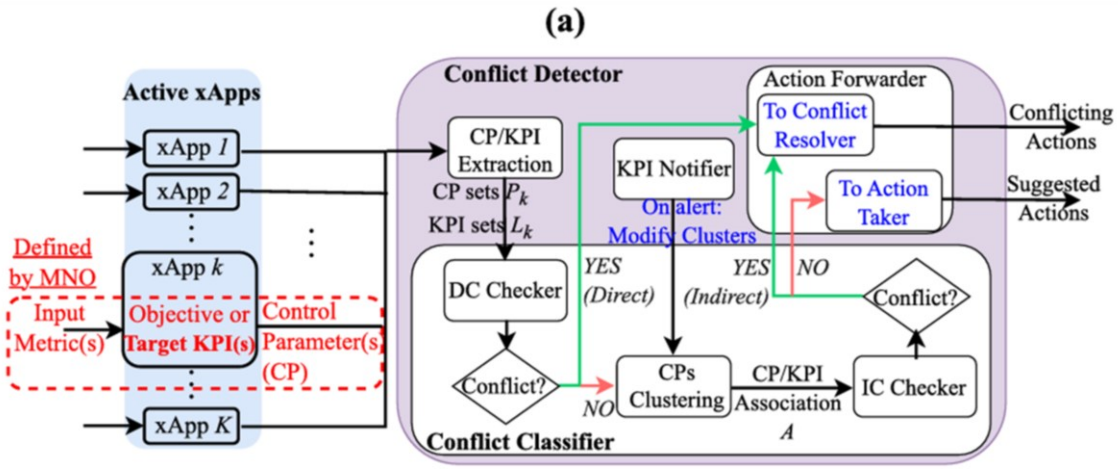


Figure 77: Conflict Detector COMIX scheme

In Figure 77 is possible to see the conflict detector scheme, from that schema is it possible to extrapolate the logic that looks for conflicts, the big white box can be coded inside a SC. The logic to detect a conflict can be summarized with the schema in Figure 78. There are two kinds of conflicts:

- Direct Conflicts: when two different xApps have between their control parameters one or more parameters in common (e.g., xApp1 and xApp2 have in common P1).
- Indirect Conflicts: when two different xApp work on the same KPI (e.g. xApp1 and xApp3 have in common L1).

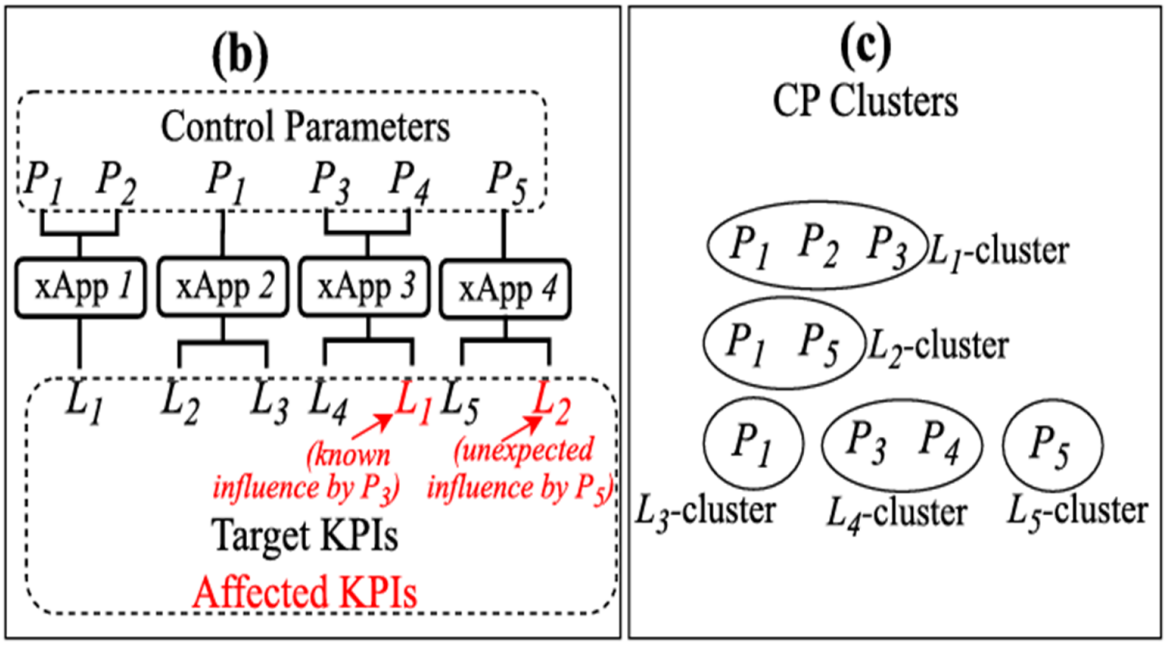


Figure 78: Conflicts Detection representation

The SC needs the CP/KPI extraction module to register an xApp with its correlated CP and KPI; it also needs a KPI notifier in case a xApp is noted to work indirectly also on a KPI not know previously. From the COMIX it is possible to note that conflicts can be detected easily if inserting CP and KPI in matrices.

Table 13: Control Parameter Matrix

	CP_1	**CP_2**	**CP_3**	**CP_4**	**CP_5**
xApp1	1	1	0	0	0
xApp2	1	0	0	0	0
xApp3	0	0	1	1	0
xApp4	0	0	0	0	1

Table 14: Target KPI Matrix

	KPI_1	**KPI_2**	**KPI_3**	**KPI_4**	**KPI_5**
xApp1	1	0	0	0	0
xApp2	0	1	1	0	0
xApp3	1	0	0	1	0
xApp4	0	1	0	0	1

In Table 13 and Table 14 it is possible to immediately understand if a conflict is present if in the same column there is more than one 1. In the tables the conflicts are highlighted. The SC uses a similar logic to check if a conflict is present, a CP or KPI can be assigned to more than a xApp, but in that case the SC will emit an event specifying which KPI or CP has a conflict and which xApps are conflicting on it. The events can be listened from conflict resolver and action taker that with the digital twin can then suggest which is the best action to take to resolve

the conflict. Once the action has been put in place it is possible to update the corresponding xApp to remove the conflict from the SC table.

6.4 CONFLICT RESOLUTION SMART-CONTRACT GENERATOR

The proposed **Conflict Resolution Smart-Contract Generator (CR-SC Generator)** combines the flexibility of LLM-driven generation with the verifiability of DSL-based compilation, and anchors both in an on-chain trust layer. The contribution beyond the SoA is threefold: (i) a **two-stage LLM-plus-DSL generation pipeline** in which the LLM targets a constrained Conflict Resolution DSL rather than Solidity directly, and a deterministic DSL-to-Solidity compiler emits the on-chain artefact bounding the LLM's output space and ensuring the deployed code is produced by a rule-based compiler; (ii) a **verification-gated feedback loop** in which contract-level checks (DSL conformance, safety properties, policy alignment) return control to the LLM on failure, so that only verified contracts reach the DLT; and (iii) a **precedent-based generation memory** that retrieves prior (context, DSL, contract) tuples to guide the LLM, enabling the system to reuse or adapt past resolutions instead of regenerating from scratch. Unlike other works, which target conflicts with a static on-chain matrix, the CR-SC Generator addresses inter-domain conflicts where the resolution logic itself is not known in advance.

6.4.1 Proposed Architecture

The overall architecture is illustrated in Figure 79 and organises four layers: a per-domain **DMO layer**, a centralised **IDMO layer**, a **CR-SC Generator**, and a **DLT Trust Layer**. The DMO layer observes and reports conflicts; the IDMO layer classifies them and formats them into structured context; the CR-SC Generator synthesises a verified smart contract from that context; and the DLT Trust Layer anchors both the reported conflict and the deployed contract, providing the cross-domain audit surface.

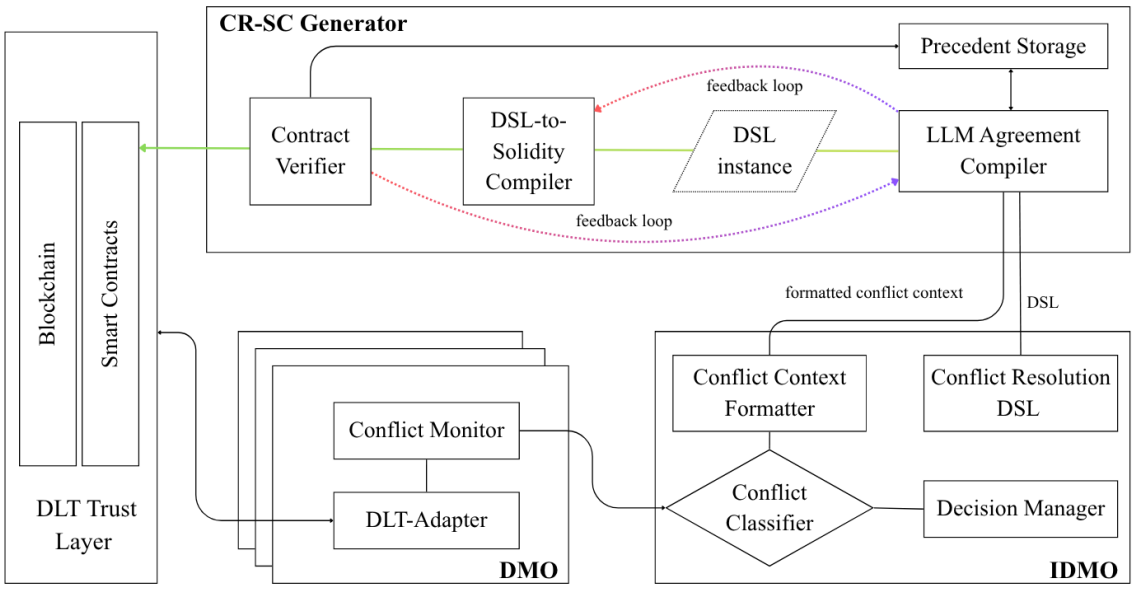


Figure 79: CR-SC Generator architecture for inter-domain conflict resolution: DMO, IDMO, and DLT trust layers

The following components are defined:

- **DMO layer (per-domain, replicated).** Each DMO contains a *Conflict Monitor* that observes local telemetry, policy state, and SLA state, and raises a conflict event whenever a local decision clashes with a peer domain or an IDMO-level constraint, and a *DLT-Adapter* that submits the event together with its evidence to the DLT Trust Layer, producing a tamper-evident record before any resolution begins.
- **DLT Trust Layer.** Realized as a blockchain with a set of smart contracts, it maintains the shared state of truth across domains. It logs conflict claims, hosts the generated resolution contracts, and enforces their execution.
- **IDMO layer.**
  - *Conflict Classifier* . triages each incoming conflict into a type (resource contention, SLA breach, policy conflict, routing dispute) and routes the workflow accordingly.
  - *Decision Manager* : orchestrates the resolution loop: it queries Precedent Storage, decides whether to reuse, adapt, or generate from scratch, and governs escalation to human-in-the-loop review for high-stakes cases.
  - *Conflict Context Formatter*: serializes the classified conflict into a standardized, machine-readable object (parties, claims, constraints, applicable SLAs and policies, evidence hashes), forming the grounded input to the LLM.
  - *Conflict Resolution DSL*: the target grammar the LLM must produce: a small, verifiable vocabulary of resolution primitives that constrains the LLM's output and makes

downstream verification tractable. This design follows the rationale of formal contract-specification DSL, adapted to the inter-domain conflict setting.

- CR-SC Generator.
  - ***LLM Agreement Compiler*** : receives the formatted context together with the DSL grammar and retrieved precedents, and produces a candidate *DSL instance*, an intermediate, human-readable specification of the proposed resolution logic.
  - ***DSL-to-Solidity Compiler*** : deterministically transpiles (a source-to-source compiler, that translates code from one high-level language to another at the same level of abstraction) the DSL instance into a Solidity contract. This step is *rule-based rather than LLM-driven*, so the final on-chain artefact can be re-audited independently of the model.
  - ***Contract Verifier*** : performs static checks on the resulting Solidity, covering DSL-to-Solidity fidelity, standard safety properties (reentrancy, bounded gas, access control), and conformance with IDMO policy constraints. On failure, two feedback loops return control to the LLM Agreement Compiler: a structural loop from the DSL-to-Solidity Compiler (rejecting malformed DSL instances) and a semantic loop from the Contract Verifier (rejecting verified but unsafe contracts).
  - ***Precedent Storage*** : on success, the (context, DSL, contract) tuple is persisted. It serves both as a retrieval corpus for future generations and as an auditable trail of how resolutions have been produced over time.

The framework operates through following phases:

1. A DMO *Conflict Monitor* detects a local decision that breaches a peer SLA or IDMO policy.
2. The *DLT-Adapter* anchors the event and its evidence to the DLT Trust Layer, creating an immutable record before resolution begins.
3. The IDMO *Conflict Classifier* categorizes the anchored event.
4. The *Decision Manager* consults *Precedent Storage* and selects the generation strategy (reuse, adapt, or generate from scratch).
5. The *Conflict Context Formatter* produces the structured context object.
6. The *LLM Agreement Compiler* generates a candidate DSL instance grounded in the context, DSL grammar, and retrieved precedents.
7. The DSL instance is exposed as an auditable intermediate artefact the natural intervention point for governance review.

8. The *DSL-to-Solidity Compiler* deterministically emits Solidity; the purple feedback loop returns malformed instances to the LLM.

9. The *Contract Verifier* checks the Solidity and, on pass, triggers on-chain deployment and enforcement on the DLT Trust Layer; the red feedback loop returns unsafe contracts to the LLM. The resulting contract and its DSL are appended to Precedent Storage for future reuse.

7 OVERVIEW OF THE ALGORITHMS

Table 15 summarizes the proposed algorithms in this deliverable.

Table 15: Overview of the proposed algorithms

Algorithm category	Algorithm	KPIs
Distributed AI-powered edge-cloud continuum and network & service resource management in dynamic multi-tenant environments	Agentic orchestration framework	KPI 37
	Green orchestration through multi criteria decision analysis	KPI 5, 7, 14, 16
	Agentic Intelligence for Non-Terrestrial Networks	KPI 17, 21
	Intent-Based Transport Configuration	KPI 6, 7, 8, 17, 20, 21, 38, 43
	SLO-based Proactive Autonomous Network Orchestration	KPI 19, 37
	The aeriOS framework	KPIs related to: PoC 1 ES1, ES2, PoC 2 ES4, ES5, ES6
	TN-NTN Traffic Offloading	KPI 20
	Energy-Efficient Decentralized Computation Offloading in the Continuum	KPI 40
	Clustered Federated Learning for Managed Wi-Fi	KPI 1,43
	ML-Based Dynamic Network Routing	KPI 7, 8, 16, 17, 20, 38, 42
	Network Recovery & Disaster Resilience via Multi-Agent Reinforcement Learning	KPI 19, 31
	Hy-DREAM for Anomaly Detection	KPI 31
	Radio Failure Prediction and Localized Recovery Estimation For Transport Wireless Links	KPI 17, 19, 31, 37
	Multivariate Probabilistic forecasting for resource allocation	KPI 17
	Container Forecasting Engine	KPI 15
Algorithms for sustainable integrated networks and energy-efficient enablers	Joint planning of access and xhaul resources	KPI 6, 17, 20, 31, 38, 42, 43
	Container Forecasting Engine	KPI 43
	NTN-based task offloading	KPI 23, 35, 45
	A resource efficient cloud controller for large scale wifi networks	KPI 1, 43
	Energy aware path computation algorithm for network slice provisioning	KPI 16, 43
	Conditional handover prediction	KPI 2, 4, 5, 7, 8
	Cell switching in integrated networks	KPI 16, 43
	AI based energy efficient massive MIMO	KPI 43
	Joint beam-user allocation strategy	KPI 43

	AI-empowered multivariate probabilistic forecasting a key enabler for sustainability in open RAN	KPI 15, 16, 17, 43
Semantic communication	AOI-aware semantic communication	KPI 2, 5, 7, 8, 15, 16, 17, 19, 22, 23, 27, 33, 34, 35, 37, 38, 40, 43, 45 (Valid for the semantic communication group)
	Semantic-aware resource allocation	
	Perception-Aware Semantic Scheduling With Remote Processing	
	The SEM-NTN framework	
Conflict resolution for integrated components	Layered Conflict Management	KPI 21 (valid for the conflict resolution group)
	Conflict Resolution Framework for O-RAN	
	Distributed Approach to COMIX	
	Conflict Resolution Smart-Contract Generator	

8 AI ETHICS BY DESIGN

Within European research and innovation projects, the principles of *ethics by design* play a central role: ethical, legal and societal implications are expected to be considered from the earliest design phases, rather than addressed only ex post as purely descriptive reflections. This means shaping the design so that future compliance is not made impossible or disproportionately costly in later stages.

Indeed, the integration of **AI-driven semantic communication** components introduces ethical and legal risks that require targeted safeguards under EU regulatory frameworks, including the GDPR and the AI Act. Semantic communication components, operating on traffic metadata and contextual information, allow the 6G network to "interpret" meaning and intent in communications to optimise resources and service quality. However, this capability can make it difficult for network operators to understand why the system makes specific decisions, and can generate inferences about user behaviours that, in some cases, constitute personal data under GDPR. Moreover, by enabling automated interpretation of contextual meaning and intent, such components may lead to **opaque decision-making processes, limiting transparency, explainability, and ex-post auditability of network reconfiguration actions**. To mitigate these risks, those components should be designed in compliance with transparency, inference limitation, human oversight and attention to infrastructural bias.

Policy-driven AI decisions combining predefined rules and automated conflict-resolution mechanisms may obscure responsibility for network outcomes, as decisions emerge from the interaction between predefined policies and automated conflict-resolution mechanisms. All this can **undermine accountability**, particularly where automated decisions produce differentiated impacts on users' access to, or quality of, network services. Policy logic must therefore be readable, traceable and clearly linked to organisational roles and responsibilities.

From a data protection perspective, even when operating primarily on network metadata, semantic inference mechanisms may generate **indirect profiling or behavioural patterns** that qualify as personal data under the GDPR. This raises risks related to **purpose creep, excessive inference, and unintended secondary use, as well as effects comparable to automated decision-making impacting individuals**.

In parallel, **AI-based network optimisation** may embed **infrastructural biases, systematically prioritising certain services, devices, or geographical areas, thereby raising fairness and non-discrimination concerns**.

Table 16 summarizes the identified risks and related requirements.

Table 16: Identified Risks and related requirements

Risk	Ethics Requirement	Technical Requirement
Opaque and non-explainable decisions	Transparency obligations under GDPR (Arts. 5, 25) and explainability under AI Act.	Decision logging, model versioning, and human-readable explanations of AI-driven network actions.
Indirect profiling and purpose creep	Purpose limitation and data minimisation principles under GDPR.	Purpose-bound inference, minimal data schemas, and technical prevention of unauthorised secondary uses.
Infrastructural bias and discriminatory outcomes	Fairness and non-discrimination obligations under the AI Act.	Fairness metrics, constrained optimisation logic, and periodic bias monitoring.
Excessive automation and loss of human oversight	Meaningful human oversight (AI Act) and safeguards against fully automated decisions (GDPR Art. 22).	Human-in-the-loop/on-the-loop thresholds, override mechanisms, and safe fallback modes.
Accountability and responsibility principles under GDPR and AI Act.	Unclear allocation of responsibility and accountability	Role-based governance, policy provenance logging, and traceable accountability mechanisms.

To address these risks, safeguards have been embedded in the above described algorithms.

Focusing on semantic communication, it is considered an important type of service within Unity-6G, relying on extensive use of AI algorithms, which may introduce opacity in making decisions. While the network can benefit from semantic communications with better adaptability and optimization of the use of resources, it is true that some decisions taken, especially when the meaning of information is accounted for, might not be obvious and fully explainable to the operator. However, while several solutions are developed for semantic communications within Unity-6G framework, the use of semantic information mostly treats semantic communications as another type of service with dynamic compression capabilities or extracting only task-specific, network-relevant features, such as Age of Information or importance. Therefore, limited impact of the transparency and explainability of the solutions is envisaged. It is worth mentioning that the UNITY-6G architecture introduces a “performance monitoring” pipeline from the AI/ML subsystem for validation and debugging purposes. The information collected in this pipeline can be also used in implementation of the decision logging mechanisms, where for each significant decision made (i.e., one having relevant impact on network configuration or performance) the information on the relevant input data (input parameters having impact on the decision), the description of the system state (in a general but explainable way), the corresponding result of AI algorithm and the final decision taken in

the network will be stored. With this logging mechanism, the description of the system state and the corresponding decision will be stored in an understandable and human-readable form for every significant decision made.

Semantic processing is strictly limited to task-specific, network-relevant concepts that are necessary for optimisation objectives such as energy efficiency, resource utilisation, and information freshness or others. Semantic features are derived only at the input level and describe technical properties rather than user identity, behaviour, or personal characteristics. No profiling of individuals is performed. To enforce data minimisation by design, only the required fields are collected and stored for optimisation loops. These may include parameters such as link capacity, queue occupancy, compression factor, coarse semantic complexity score, and KPI-related outputs (e.g., mAP or mIoU estimates). Raw images or video streams are not retained by default. When temporary access to raw data is strictly necessary for debugging or evaluation, retention is time-limited and access-controlled. Monitoring schemas are based on aggregated counters or histograms rather than per-sample payload storage, thereby avoiding persistent data traces.

Regarding the technical enforcement of purpose binding, the following solutions will be considered:

- Every data item and inference output will be tagged with: purpose_id (e.g., SEM_COMPRESS_CONTROL, RAN_SLICE_OPT, DEBUG_EVAL), scope (TN/NTN domain, cell/beam id), sensitivity level, retention_ttl.
- Policy Engine (e.g. Non-RT RIC/SMO or DMO-IDMO) will distribute allow/deny rules (ABAC style) so only authorized apps can subscribe/use specific purpose tags. Distributed Ledger Technologies will be used for this purpose more specifically.
- For enforcement, mechanisms such as: (i) **topic-based isolation** (separate message buses/topics per purpose; no “wildcard” subscriptions); (ii) **data-plane guards** at ingestion where forwarding of payloads lacking a valid purpose tag or exceeding schema will be blocked will be applied especially during IDMO design and implementation.
- For auditing, every access will be logged as (who, what, purpose_id, time) to demonstrate technical prevention of secondary use.

Before enabling any new semantic-based inference, a standardized Impact Assessment workflow is going to be executed. In the procedure, first capability will be described, i.e. what

the inference predicts/infers, required inputs, outputs, and intended network action. Then, risk will be classified to answer questions such as

- Does it infer attributes about individuals/users? (profiling risk),
- Does it trigger actions that materially affect service access/quality? (significant automated decision-making)
- Could outputs be repurposed for other uses? (secondary use risk)

Then, **data minimization and Purpose binding reviews** will be executed. Later, human override and transparency will be enabled with explainable templates and rollback strategy. Outputs include a short “impact assessment record” with decision rationale and mitigations.

Moving to the developed AI/ML algorithms for network control and resource allocation, they inherently target the minimization of systematic discriminations and also avoid infrastructure bias that may be present due to network equipment. To this end, the prioritisation of services or geographical areas is limited to the operation of the network service providers and does not rely on the output of ML models. For instance, the mission critical services are prioritised in case of emergency, but this is not done during the typical network operation. The algorithms related to network reconfiguration and allocation of network resources based on reinforcement learning have been trained in simulated scenarios with no bias included in the reward function. Moreover, the algorithms are trained in diverse scenarios that reflect various conditions of the network, for instance diverse temporal evolution of the network traffic in different areas (urban, rural, etc.). It is worth noting that the decisions of the ML models depend on the underlying infrastructure. In essence, rural geographical areas will include fewer radio equipment (e.g., Base stations) and lower quality of services (in terms of throughput and in general QoS). However, this discrimination is embedded in the network infrastructure and not in the design of the AI/ML algorithms. To overcome these potential systematic discriminations, the designed AI/ML algorithms include several fairness metrics in their optimization objective, as detailed below. Finally, a performance monitoring mechanism is also included in the system architecture, aiming at detecting potential biases in the real network after the model deployment.

Regarding the inclusion of fairness metrics, several such metrics are included in the optimization problems that ensure the fairness among the users and services. In specific, metrics such as Jain's fairness index are included to the optimization objectives (fair

distribution of user QoS) when allocating resources to the users. Moreover, violations in the provided services are introduced per network slice, guaranteeing the fairness among the users of each slice, i.e., the users of an ultra-high reliable and low-latency network slice exhibit the same QoS. IIn addition, to provide fairness in the whole network a multi-level fairness analysis is considered, where the information on network resource distribution fairness will be collected both based on the geographical area and on the service-type level. As UNITY6G considers coexistence of both traditional binary communications and semantic communications, we consider also use of QoS/QoE-based metrics, e.g. such as the one proposed in [1], to provide means of comparing the performance and fairness for both the binary and semantic transmission. Finally, in the case of integrated networks, i.e., TN and NTN, fairness metrics will be evaluated taking into account the peculiarities of individual network deployments, for instance the visibility time of the flying platform.

The optimization objectives of the ML algorithms are general, for instance relying on the maximization of the user QoS/QoE, as well as the maximization of the system energy efficiency. In this context, the bias that can be potentially introduced through the optimization objectives is minimal. However, additional constraints on the optimized parameters can be formulated either involving human (operator) decisions or being policy-based. Finally, fairness among the users or among the geographical areas can be included in the optimisation objective, e.g., by using proportional fairness approaches in the traffic scheduling decisions.

The performance of models that are deployed and operating in the network is monitored periodically to ensure that the accuracy levels are maintained through “evaluation” or “performance monitoring” pipeline from the AI/ML subsystem of the architecture. The performance of these models can be tested against required accuracy levels (in case of an ML prediction algorithm) or in terms of KPI violation, where the KPIs can be defined by the end-user. If degradation of the model performance is detected (e.g., due to environmental drift), the user can stop its operation and initiate a re-training process.

Regarding policy-driven AI decisions, combining predefined rules and automated conflict-resolution mechanisms may obscure responsibility for network outcomes.

The 3 modes can be supported by the Unity-6G architecture depending on the AI/ML task and algorith.

- ***Monitoring Mode***: the monitoring of network KPIs are acknowledged to the network operator and no high-level network reconfiguration (in the management domain) is conducted without his approval. Moreover, the architecture involves a performance

monitoring feature of ML model outputs to periodically check the accuracy of the deployed models and notify the operator in case of model degradation. The monitoring mode certainly aligns with the longer time scales of the management layer (e.g., rApps in the O-RAN architecture).

- ***Assisted Mode***: The assisted model aligns with the architecture of Unity-6G, since it can be incorporated in the hierarchical scheme in the IDMO layer, especially in the case of conflict management. To this end, higher-level conflicts (e.g., conflicts that arise between intents from different operators and span, in general, across multiple domains) will not be applied automatically but required confirmation by a human. Similar considerations are also valid for higher-level policies (e.g., service-level agreements or KPI violations) that affect many network users. The assisted mode also aligns with the rApp time scales.
- ***Automatic Mode***: Unity-6G architecture supports ML models that are required to work in near real-time (or in real-time), since they need to operate in an automatic mode. This category includes models like xApps and dApps, where time scales are <1s and no human intervention is needed. This mode can be used in the applications of scheduling, radio control including bandwidth allocation, beamforming, etc. Moreover, conflicts that arise between these intelligent applications will be also resolved in an automatic mode (CM inside a specific domain) and a log will be generated to explain to the human operator the model output.

For instance a beamforming ML model that needs to “track” the user needs to operate in ms time scales and therefore, must be in an automatic mode. On the other hand, higher level policies suggested by AI algorithms that affect multiple users are not required to operate in such short time scales and can be executed either in an assisted mode (e.g., recommendation for the placement of the network functions in a chain) or simply be used for monitoring purposes (e.g., forecasting of the mobile traffic). In case of automatic mode, the logs can (in this case, must) contain information about who and why a certain decision was made, explaining why the model performed an operation and with the opportunity for a human operator to impose a different network configuration. All those data can be anonymized if contain relevant personal information.

For instance, when using the DLT, every operation done is written to an immutable block, which represents a log. When using the DLT, any personal data that could be needed by a smart contract is anonymized since it will be recorded into the blockchain. This is done using public keys and/or random identifiers. Every transaction, recorded in the DLT, contains the

public key that sent the transaction and all the data required by that function call to work. It should be noted that the identifiers or content that may qualify as personal data will not be stored as plain text, since this does not conform with the data protection principles. To this end, several precautions will be considered if personal data are associated with on-chain storage, including encryption of the data before storing it, hashing of the personal data and the use of cryptographic commitments.

Finally, policy-driven AI decisions and conflict resolution mechanisms can undermine accountability, particularly where automated decisions produce differentiated impacts on users' access to, or quality of, network services. In every distributed decision-making system, there exist a risk of distributed and blurred responsibility. In the Unity6G context, we are dealing with: (1) inter-domain and (2) intra-domain conflict management (CM), as well as (3) CM module residing inside IDMO. As the intra-domain solutions are beyond the project scope (and are typically subject to the decision of a particular service provider), the intra-domain CM, as well as IDMO-related conflict management, have to be designed carefully to mitigate the ethical risks. The following rules have to be followed while designing any decision making scheme and solution.

- For a proper definition of the CM within Unity6G, we will define the strict responsibility areas of each entity involved in the conflict management process. All actors will be defined, and their roles will be clearly specified. Moreover, the coordination rules defined within the SBMA will protect the overall architecture from violating the agreed responsibility zones. In order to avoid any external impact, the zero-trust layer will be involved.
- The acting policy applied in each CM module (inside the domain, inside IDMO) will be clearly defined. However, the CM rules applied within the domain are typically delivered by the domain provider. At the same time, the IDMO owner will be the one responsible for policy definition. The rules for defining the policy are subject to research and will be clearly discussed.
- In the whole project, we assume the presence of the inner-loop (from monitoring to decision making and acting); in the context of CM, the mechanism to identify the impact of the applied CM rules is obligatory to keep the performance at the highest possible level (IDMO has to monitor the deployment of the services); thus storage of the decisions and tracking of it is a inherent part of the system.

9 CONCLUSIONS

This deliverable, D4.1, is the intermediate report of WP4 and has presented the first collection of the AI-driven algorithms developed within WP4. Following a common methodology, each contribution has been described, positioned with respect to the state of the art and mapped onto the UNITY-6G architecture. The algorithms have been organised into four main categories: distributed AI for the edge-cloud continuum, sustainable and energy-efficient enablers, semantic communication, and conflict resolution.

The main outcomes of each technical section can be summarised as follows.

Section 3 introduced a broad family of distributed AI algorithms for network and service resource management in dynamic multi-tenant environments, grouped into network orchestration, resource allocation / traffic load-balancing / routing, and forecasting and prediction. The orchestration contributions share a common AI-native control loop built on the MS-AE-DE-ACT pattern and the hierarchical IDMO/DMO structure. The resource-allocation and forecasting algorithms collectively target service continuity, service-recovery time, and link utilisation.

Section 4 focused on sustainability and energy efficiency across the integrated network. The network-orchestration-and-planning algorithms and the radio-resource-management algorithms are predominantly oriented towards the energy-efficiency target, CAPEX/OPEX reduction and cross-domain latency and reliability. The power-grid measure-forecasting activity complements these by acting as an enabler, supplying grid-aware “energy-colour” data to support the training of the distributed-AI algorithms.

Section 5 defined a common semantic-communication architecture together with four algorithmic algorithms: semantic-aware task offloading, semantic-aware resource allocation, perception-aware semantic scheduling, and the SEM-NTN framework. These solutions exploit meaning-based compression, QoE-aware rate adaptation and learning-based offloading to reduce communication overhead, improve NTN reliability and preserve SLA and QoS guarantees under semantic operation.

Section 6 addressed the coordination problem that arises when multiple intelligent controllers act concurrently on shared resources. Adopting a multi-level view of conflicts (intra-domain, inter-domain and orchestrator-level), it presented the layered conflict-management architecture for the O-RAN domain, the COMIX conflict-resolution framework, its distributed smart-contract realisation, and a predictive conflict-avoidance approach based on an LLM-

plus-DSL Conflict-Resolution Smart-Contract Generator anchored in a DLT trust layer. Collectively these mechanisms target the reduction of the performance penalty due to conflicts.

By design, this deliverable is an intermediate report: it establishes the preliminary design and the architectural positioning of each algorithm in the architecture and provides the first preliminary results. The subsequent WP4 deliverables, D4.2 and D4.3 will build on this baseline by consolidating the above algorithms; extending the preliminary evaluations; and progressing the integration of the selected algorithms into the project PoCs and experimental scenarios, in close coordination with the architecture, digital-twin and validation activities of UNITY-6G.

REFERENCES

[1] S. L. a. C. L. M. Correa, “Supporting MANOaaS and heterogenous MANOaaS deployment within the zero-touch network and service management framework,” *IEEE Communications Standards Magazine,* vol. 8, pp. 4-11, 2024.

[2] A. a. K. A. a. V. C. Mekrache, “Intent-based management of next-generation networks: An LLM-centric approach,” *Ieee Network,* vol. 38, pp. 29-36, 2024.

[3] D. a. T. K. a. A. B. a. T. G. C. a. T. C. a. D. S. a. B. A. Brodimas, “Towards intent-based network management for the 6G system adopting multimodal generative AI,” *2024 Joint European Conference on Networks and Communications \& 6G Summit (EuCNC/6G Summit),* pp. 848--853, 2024.

[4] D. M. a. C. A. a. S. A. Manias, “Towards intent-based network management: Large language models for intent extraction in 5g core networks,” in *2024 20th International Conference on the Design of Reliable Communication Networks (DRCN)*, IEEE, 2024, pp. 1-6.

[5] N. a. F. M. a. Q. S. a. W. A. G. Gruver, “Large language models are zero-shot time series forecasters,” *Advances in neural information processing systems,* vol. 36, pp. 19622--19635, 2023.

[6] D. a. W. Z. a. L. X. a. X. Q. a. Z. T. a. Z. R. a. T. L. a. Z. P. Jiang, “Toward Synthetic Network Traffic Generating in NTN-enabled IoT: A Generative AI Approach,” *IEEE Internet of Things Journal,* vol. 12, pp. 2174--2187, 2024.

[7] M. a. C. Y. a. S. G. a. M. M. a. S. Y. a. Z. X. a. M. J. a. V. D. a. L.-C. P. a. G. S. M. a. o. Shetty, “Building ai agents for autonomous clouds: Challenges and design principles,” in *Proceedings of the 2024 ACM Symposium on Cloud Computing*, 2024, pp. 99--110.

[8] D. B. a. K. K. a. D. B. Acharya, “Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey,” *IEEE Access,* vol. 13, pp. 18912-18936, 2025.

[9] J. White, “Building living software systems with generative \& agentic AI,” *arXiv preprint arXiv:2408.01768,* 2024.

[10] R. a. T. S. a. L. Y. Zhang, “Toward agentic AI: Generative information retrieval inspired intelligent communications and networking,” *arXiv preprint arXiv:2502.16866,* 2025.

[11] S. a. M. J. Araujo, “An agentic approach for dynamic software-defined network management using large language models,” in *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, IEEE, 2024, pp. 221--226.

[12] I. a. L. A. a. N. N. Chatzistefanidis, “Maestro: LLM-Driven Collaborative Automation of Intent-Based 6G Networks,” *IEEE Networking Letters,* vol. 6, pp. 227--231, 2024.

[13] A. a. R. P. Habib, “Llm-based intent processing and network optimization using attention-based hierarchical reinforcement learning,” in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2025, pp. 1--6.

[14] European Telecommunications Standards Institute (ETSI). (n.d.). Zero-touch network and Service Management (ZSM). Retrieved from https://www.etsi.org/technologies/zero-touch-network-service-management

[15] O-RAN Alliance. (n.d.). O-RAN Architecture Description. Retrieved from https://www.o-ran.org/specifications

[16] 3rd Generation Partnership Project (3GPP). (n.d.). Service-Based Management Architecture (SBMA).

[17] Hexa-X. (n.d.). A flagship for B5G/6G vision and intelligent fabric of technology enablers connecting human, physical, and digital worlds. Retrieved from https://hexa-x.eu/

[18] ADROIT6G. (n.d.). Distributed Artificial Intelligence-Driven Open and Programmable Architecture for 6G Networks. Retrieved from https://adroit6g.eu/

[19] 3GPP ’TS 28.530 Management and orchestration; Concepts, use cases and requirements Release 18 2024’

[20] Foukas, Xenofon, et al. "Network slicing in 5G: Survey and challenges." *IEEE communications magazine* 55.5 (2017): 94-100.

[21] Leivadeas, Aris, and Matthias Falkner. "A survey on intent-based networking." *IEEE Communications Surveys & Tutorials* 25.1 (2022): 625-655.

[22] Clemm, Alexander, et al. "Intent-based networking-concepts and definitions." (2022): 1-23

[23] 3GPP 'TS 28.312 Management and orchestration; Intent driven management services for mobile networks 2025'

[24] Coronado, Estefania, et al. "Zero touch management: A survey of network automation solutions for 5G and 6G networks." *IEEE Communications Surveys & Tutorials* 24.4 (2022): 2535-2578.

[25] 3GPP 'TS 28.105 Management and Orchestration; Artificial Intelligence/Machine Learning (AI/ML) Management Release 18'

[26] S. C. Neguerole *et al.*, "Scalable and Adaptive Security Framework for the IoT-Edge-Cloud Continuum," *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*, Chania, Crete, Greece, 2025, pp. 350-357, doi: 10.1109/CSR64739.2025.11130022.

[27] Trakadas, P.; Masip-Bruin, X.; Facca, F.M.; Spantideas, S.T.; Giannopoulos, A.E.; Kapsalis, N.C.; Martins, R.; Bosani, E.; Ramon, J.; Prats, R.G.; et al. A Reference Architecture for Cloud–Edge Meta-Operating Systems Enabling Cross-Domain, Data-Intensive, ML-Assisted Applications: Architectural Overview and Key Concepts. *Sensors* **2022**, *22*, 9003. https://doi.org/10.3390/s22229003

[28] Gkonis, P.; Giannopoulos, A.; Trakadas, P.; Masip-Bruin, X.; D'Andria, F. A Survey on IoT-Edge-Cloud Continuum Systems: Status, Challenges, Use Cases, and Open Issues. *Future Internet* **2023**, *15*, 383. https://doi.org/10.3390/fi15120383

[29] L. Belcastro, F. Marozzo, A. Orsino, D. Talia and P. Trunfio, "Navigating the Edge-Cloud Continuum: A State-of-Practice Survey," in *IEEE Access*, vol. 14, pp. 40622-40647, 2026, doi: 10.1109/ACCESS.2026.3673012

[30] S. M. Shahid, Y. T. Seyoum, S. H. Won, and S. Kwon, "Load balancing for 5G integrated satellite-terrestrial networks," *IEEE Access*, vol. 8, pp. 132144–132156, 2020.

[31] M. Bello, P. Pillai, and A. S. Sadiq, "An intelligent load balancing algorithm for 5G-satellite networks," *Int. J. Satell. Commun. Netw.*, vol. 42, no. 5, pp. 329–353, 2024.

[32] C. Wan and B. Li, "DQN-based network selection and load balancing for LEO satellite-terrestrial integrated networks," in *Proc. 7th World Conf. Computing and Communication Technologies (WCCCT)*, pp. 233–238, 2024.

[33] F. Minani et al., "Channel prediction and fair resource allocation for NTN uplinks by LSTM and deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 24, no. 10, pp. 8311–8330, 2025.

[34] S. R. Trankatwar, H. Straulino, P. Djukic, and B. Kantarci, "FTA-NTN: Fairness and throughput assurance in non-terrestrial networks," *arXiv:2601.19078*, 2026.

[35] Y. Li, F. Qi, Z. Wang, X. Yu and S. Shao, "Distributed Edge Computing Offloading Algorithm Based on Deep Reinforcement Learning," in IEEE Access, vol. 8, pp. 85204-85215, 2020, doi: 10.1109/ACCESS.2020.2991773.

[36] A. E. Giannopoulos, I. Paralikas, S. T. Spantideas and P. Trakadas, "HOODIE: Hybrid Computation Offloading via Distributed Deep Reinforcement Learning in Delay-Aware Cloud-Edge Continuum," in IEEE Open Journal of the Communications Society, vol. 5, pp. 7818-7841, 2024, doi: 10.1109/OJCOMS.2024.3514456.

[37] A. Giannopoulos, I. Paralikas, S. Spantideas, N. Nomikos and P. Trakadas, "PDPPnet: Prioritized Delay-aware and Peer-to-Peer Task Offloading in Cloud-Edge Continuum with Double Dueling Deep Q-Networks," 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 2024, pp. 1-8, doi: 10.1109/CAMAD62243.2024.10943043.

[38] A. Hussain and M. Fahad, "Semi-supervised clustered federated learning based indoor localization with non-iid data," The Journal of Supercomputing, vol. 81, no. 8, p. 962, Jun 2025.

[39] Y. Liu, Z. Sun, Y. Xiao et al., “Clustered federated learning for distributed wireless sensing,” in GLOBECOM 2024 - 2024 IEEE Global Communications Conference, 2024, pp. 367–372.

[40] B. Xie, X. Dong, and C. Wang, “An improved k-means clustering intrusion detection algorithm for wireless networks based on federated learning,” Wireless Communications and Mobile Computing, vol. 2021, no. 1, p. 9322368, 2021.

[41] D. Salami, F. Wilhelmi, L. Galati-Giordano et al., “Distributed learning for wi-fi ap load prediction,” in GLOBECOM 2024 - 2024 IEEE Global Communications Conference, 2024, pp. 968–973.

[42] N. Pavlidis, V. Perifanis, S. F. Yilmaz et al., “Federated learning in mobile networks: A comprehensive case study on traffic forecasting,” IEEE Transactions on Sustainable Computing, vol. 10, no. 3, pp. 576–587, 2025.

[43] Ying, Lei, et al. "On combining shortest-path and back-pressure routing over multihop wireless networks." *IEEE/ACM Transactions on Networking* 19.3 (2010): 841-854.

[44] Liu, Keqin, and Qing Zhao. "Adaptive shortest-path routing under unknown and stochastically varying link states." *2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. IEEE, 2012

[45] Amar, Omer, Ilana Sarfati, and Kobi Cohen. "An online learning approach to shortest path and backpressure routing in wireless networks." *IEEE Access* 11 (2023): 57253-57267.

[46] Rao, Zheheng, et al. "DAR-DRL: A dynamic adaptive routing method based on deep reinforcement learning." *Computer Communications* 228 (2024): 107983.

[47] Chai, Yuan, and Xiao-Jun Zeng. "Delay-and interference-aware routing for wireless mesh network." *IEEE Systems Journal* 14.3 (2020): 4119-4130.

[48] Paul, Raz, Kobi Cohen, and Gil Kedar. "Multi-flow transmission in wireless interference networks: A convergent graph learning approach." *IEEE Transactions on Wireless Communications* 23.4 (2023): 3691-3705.

[49] 3GPP ‘TS 23.501 System Architecture for the 5G System (5GS) Release 18’

[50] Zhang, Yongqiang, Mustafa A. Kishk, and Mohamed-Slim Alouini. "A survey on integrated access and backhaul networks." *Frontiers in Communications and Networks* 2 (2021): 647284.

[51] Esposito, Christian, et al. "On the disaster resiliency within the context of 5G networks: The RECODIS experience." *Proceedings of EuCNC 2018* (2018).

[52] Shand, Mike, and Stewart Bryant. *IP fast reroute framework*. No. rfc5714. 2010.

[53] Rehman, Anees U., Rui L. Aguiar, and Joao Paulo Barraca. "Fault-tolerance in the scope of software-defined networking (sdn)." *IEEE access* 7 (2019): 124474-124490.

[54] Zhang, Yongqiang, Mustafa A. Kishk, and Mohamed-Slim Alouini. "A survey on integrated access and backhaul networks." *Frontiers in Communications and Networks* 2 (2021): 647284.

[55] Ni, Sze-Yao, et al. "The broadcast storm problem in a mobile ad hoc network." *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. 1999.

[56] O. C. K. C. a. M. B. A. Sebbar, “Real-time anomaly detection in SDN architecture using integrated SIEM and machine learning for enhancing network security,” GLOBECOM, pp. 1795-8000, 2023.

[57] G. R. a. P. O. J. Dromard, “Online and scalable unsupervised network anomaly detection method,,” IEEE TNSM Journal, vol. 14, pp. 34-47, 2016.

[58] F. S. A. B. N. A. E. a. A. S. M. Injadat, “Bayesian optimization with machine learning algorithms towards anomaly detection,” GLOBECOM, 2018.

[59] Hasan, Kazi, et al. "A Generalized GNN-Transformer-Based Radio Link Failure Prediction Framework in 5G RAN." *IEEE Transactions on Machine Learning in Communications and Networking* (2025)

[60] Lim, Bryan, et al. "Temporal fusion transformers for interpretable multi-horizon time series forecasting." *International journal of forecasting* 37.4 (2021): 1748-1764

[61] Salinas, David, et al. "High-dimensional multivariate forecasting with low-rank gaussian copula processes." *Advances in neural information processing systems* 32 (2019)

[62] (TIP) Telecom Infra Project, “Open RAN Integration Guidelines,” 2024.

[63] Alliance, NGMN, “6G Requirements and Design Considerations,” 2025.

[64] Alliance, O-RAN, “RIC Architecture Description v3.0,” 2024.

[65] Alliance, O-RAN, “Governance of O-RAN ALLIANCE e.V. in Compliance with WTO Principles White Paper,” pp. https://mediastorage.o-ran.org/white-papers/O-RAN.O-RAN-ALLIANCE-in-Compliance-with-WTO-Principles-white-paper-2023-07-v02.pdf, July 2023

[66] F. Rossi, M. Nardelli and V. Cardellini, "Horizontal and Vertical Scaling of Container-Based Applications Using Reinforcement Learning," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 2019, pp. 329-338, doi: 10.1109/CLOUD.2019.00061.

[67] B. Suleiman, M. J. Alibasa, Y.-Y. Chang, and A. Anaissi, "Predictive Auto-scaling: LSTM-Based Multi-step Cloud Workload Prediction," in *Service-Oriented Computing – ICSOC 2023 Workshops* (Lecture Notes in Computer Science, vol. 14518)

[68] S. Alharthi, A. Alshamsi, A. Alseiari, A. Alwarafy, "Auto-Scaling Techniques in Cloud Computing: Issues and Research Directions," Sensors, 2024. DOI: 10.3390/s24175551

[69] T. Tournaire, H. Castel-Taleb, E. Hyon, "Efficient Computation of Optimal Thresholds in Cloud Auto-scaling Systems," ACM ToMPECS, 2023. DOI: 10.1145/3603532

[70] H. Yuan, S. Liao, "A Time Series-Based Approach to Elastic Kubernetes Scaling," Electronics, 2024. DOI: 10.3390/electronics13020285

[71] Nguyen, Tam N. "Holistic cold-start management in serverless computing cloud with deep learning for time series." (2024): 312-325

[72] https://aeros-project.eu/objectives/

[73] D2.7 – aerOS architecture definition (2), D2.7 – aerOS architecture definition (2), https://aeros-project.eu/wp-content/uploads/2024/07/D2.7.pdf

[74] D3.3 – Final distributed compute infrastructure specification and implementation, https://aeros-project.eu/wp-content/uploads/2025/11/D3.3.pdf

[75] aeriOS Project, Eclipse Foundation, https://projects.eclipse.org/projects/iot.aerios

[76] aeriOS Repository, https://github.com/eclipse-aerios/resources

[77] Greidi, Ran, et al. "DPIR: Deep Predictive Interference-Aware Routing for Wireless Networks." *2025 61st Allerton Conference on Communication, Control, and Computing Proceedings*. Allerton Conference on Communication, Control, and Computing, 2025.

[78] V. Weerackody, K. Benson, and S. Roy, “Who needs base stations when we have side links?,” IEEE Communications Society (CTN), 2023.

[79] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus, “High-dimensional multivariate forecasting with low-rank gaussian copula processes,” Advances in neural information processing systems, vol. 32, 2019.

[80] B. Lim, S. Ö. Arik, N. Loeff y T. Pfister, «Temporal fusion transformers for interpretable multi-horizon time,» *International Journal of Forecasting,* vol. 37, nº 4, p. 1748–1764, 2021.

[81] Mohamed Mekki, Nassima Toumi, & Adlen Ksentini. (2022). Benchmarking on Microservices Configurations and the Impact on the Performance in Cloud Native Environments (1.0) [Data set]. LCN 2022, 47th Annual IEEE Conference on Local Computer Networks, Edmonton, Canada. Zenodo. https://doi.org/10.5281/zenodo.6907619

[82] H. Chase, LangChain, 2022, [Online]. Available: https://github.com/langchain-ai/langchain

[83] LangGraph, 2024, [Online]. Available: https://github.com/langchain-ai/langgraph

[84] LangSmith, 2023, [Online]. Available: https://github.com/langchain-ai/langsmith-sdk

[85] Shehzad, M. K., Ahmad, A., Hassan, S. A., & Jung, H. (2021). Backhaul-aware intelligent positioning of UAVs and association of terrestrial base stations for fronthaul connectivity. IEEE Transactions on Network Science and Engineering, 8(4), 2742–2755.

[86] Yao, J., & Ansari, N. (2017). Joint caching in fronthaul and backhaul constrained C-RAN. In *2017*

IEEE Global Communications Conference (GLOBECOM), 1–6.

[87] Fayad, A., Cinkler, T., & Rak, J. (2023). 5G/6G optical fronthaul modeling: cost and energy consumption assessment. Journal of Optical Communications and Networking, 15(9), D33–D46. DOI: 10.1364/JOCN.486547.

[88] Tezergil, B., & Onur, E. (2022). Wireless backhaul in 5G and beyond: Issues, challenges and opportunities. *IEEE Communications Surveys & Tutorials, 24*(4), 2579–2632.

[89] Larsen, L. M., Ruepp, S., Berger, M. S., & Christiansen, H. L. (2022). RAN design guidelines for energy efficient 5G mobile xHaul networks. In *Proceedings of the 14th International Conference on Communications (COMM)*, 1–6.

[90] Klinkowski, M. (2024). Optimized planning of DU/CU placement and flow routing in 5G packet xHaul networks. *IEEE Transactions on Network and Service Management, 21*(1), 232–248.

[91] N. Kazemifard and D. Rahmani, “QoS-aware placement of MEC servers in an O-RAN architecture,” Cluster Computing, vol. 29, no. 3, p. 136, 2026, doi: 10.1007/s10586-025-05887-9.

[92] S. Ghasvarianjahromi, A. Kiani, A. Xiang, J. Kaippallimalil, T. Saboorian, and N. Ansari, “AI/ML-based QoS recommendations for energy optimization in 6G networks,” IEEE Networking Letters, vol. 8, pp. 44–48, 2026, doi: 10.1109/LNET.2026.3660985.

[93] Y. Lin, W. Jiao, Y. Zhang, C. Li, F. Shu, and J. Li, “Joint task scheduling and resource allocation for semantic-aware VEC: A Lyapunov-guided multi-objective reinforcement learning approach,” IEEE Transactions on Communications, vol. 74, pp. 3814–3829, 2026, doi: 10.1109/TCOMM.2026.3657402.

[94] X. Zhao, F. Hou, P. Yuan, C. Wang, J. Zhang, and H. Jin, “Recommendation-enabled caching strategy with age of information in edge-cloud networks,” IEEE Transactions on Network Science and Engineering, vol. 13, pp. 1538–1553, 2026, doi: 10.1109/TNSE.2025.3597136.

[95] X. Zou, R. Xie, Z. Xiong, G. Xie, Q. Tang, T. Huang, C. Yuen, and Z. Han, “Optimizing task offloading in dynamic satellite-terrestrial integrated computing power networks: A time-space-aware DRL approach,” IEEE Transactions on Cognitive Communications and Networking, vol. 12, pp. 3631–3649, 2026, doi: 10.1109/TCCN.2025.3628477.

[96] J. Deng, S. F. Hasan, H. Zhou, S. Al-Ahmadi, M.-S. Alouini, and D. B. da Costa, “AI-native Open RAN for non-terrestrial networks: An overview,” IEEE Open Journal of the Communications Society, vol. 7, pp. 1033–1057, 2026, doi: 10.1109/OJCOMS.2026.3656366.

[97] J. Paparrizos and L. Gravano, “k-shape: Efficient and accurate clustering of time series,” in Proceedings of the 2015 ACM SIGMOD international conference on management of data, 2015, pp. 1855–1870.

[98] T. W. Liao, “Clustering of time series data—a survey,” Pattern recognition, vol. 38, no. 11, pp. 1857–1874, 2005.

[99] R. J. Hyndman, E. Wang, and N. Laptev, “Large-scale unusual time series detection,” in 2015 IEEE international conference on data mining workshop (ICDMW). IEEE, 2015, pp. 1616–1619.

[100] A. Baz, J. Logeshwaran, Y. Natarajan, and S. K. Patel, “Enhancing mobility management in 5g networks using deep residual lstm model,” Applied Soft Computing, vol. 165, p. 112103, 2024.

[101] C. Lee, H. Cho, S. Song, and J.-M. Chung, “Prediction-based conditional handover for 5g mm-wave networks: A deep-learning approach,” IEEE Vehicular Technology Magazine, vol. 15, no. 1, pp. 54–62, 2020.

[102] W.-H. Yang, T. T. Phan, Y.-A. Chen, J.-X. Lin, and C.-Y. Li, “Smart handover between 5g and wi-fi over quic at network edge for maximizing throughput,” in 2024 International Computer Symposium (ICS). IEEE, 2024, pp. 127–132.

[103] M. Salamatmoghadasi, A. Mehrabian, H. Yanikomeroglu, and G. Kaddoum, "Sustainable Vertical Heterogeneous Networks: A Cell Switching Approach With High Altitude Platform Station," IEEE Transactions on Green Communications and Networking, vol. 10, pp. 1951–1966, 2026.

[104] D. Rössler, A. Fastenbauer, P. Svoboda, and M. Rupp, "Power Consumption Reduction by Switching Off Base Stations," in Proc. 66th International Symposium ELMAR, Zadar, Croatia, pp. 65–68, Sept. 2024.

[105] C.-W. Huang and P.-C. Chen, "Joint Demand Forecasting and DQN-Based Control for Energy-Aware Mobile Traffic Offloading," IEEE Access, vol. 8, pp. 66588–66597, 2020.

[106] Z. Rezaei and B. S. Ghahfarokhi, "Energy and Spectrum Efficient Cell Switch-Off with Channel and Power Allocation in Ultra-Dense Networks: A Deep Reinforcement Learning Approach," Computer Networks, vol. 234, p. 109912, 2023.

[107] X. Huang, S. Tang, Q. Zheng, D. Zhang, and Q. Chen, "Dynamic Femtocell gNB On/Off Strategies and Seamless Dual Connectivity in 5G Heterogeneous Cellular Networks," IEEE Access, vol. 6, pp. 21359–21368, 2018.

[108] N. Rajule, M. Venkatesan, S. Pawar, R. Menon, A. Thorat, and H. Magar, "Deep Learning-Based Adaptive Base Station Switching for Energy-Efficient Mobile Networks," in Proc. 9th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, Aug. 2025.

[109] C. Tang, "Base Station Switch-Off Strategy for Energy-Efficient Beyond 5G Wireless Networks," in Proc. International Conference on Embedded Systems, Mobile Communication and Computing (EMC²), pp. 12–17, 2026.

[110] E. S. L. H. J. &. D. M. Björnson, “Designing Multi-User MIMO for Energy Efficiency: When is Massive MIMO the Answer?,” *WCNC ,* pp. 242-247, 2014.

[111] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?", IEEE Wireless Communications, vol. 18, no. 5, pp. 40-49, Oct. 2011, doi: 10.1109/MWC.2011.6056691.

[112] I. A. Hemadeh, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, "Millimeter-Wave Communications: Physical Channel Models, Design Considerations, Antenna Constructions, and Link-Budget", IEEE Communications Surveys & Tutorials, vol. 20, no. 2, pp. 870-913, 2018, doi: 10.1109/COMST.2017.2783541.

[113] M. B. Booth, V. Suresh, N. Michelusi, and D. J. Love, "Multi-Armed Bandit Beam Alignment and Tracking for Mobile Millimeter Wave Communications", IEEE Communications Letters, vol. 23, no. 7, pp. 1244-1248, Jul. 2019, doi: 10.1109/LCOMM.2019.2919016.

[114] M. Hussain and N. Michelusi, "Second-Best Beam-Alignment via Bayesian Multi-Armed Bandits", in Proc. IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013578.

[115] M. Feng, B. Akgun, I. Aykin, and M. Krunz, "Beamwidth Optimization for 5G NR Millimeter Wave Cellular Networks: A Multi-Armed Bandit Approach", in Proc. IEEE International Conference on Communications (ICC), 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500381.

[116] Y. Zhang, S. Basu, S. Shakkottai, and R. W. Heath Jr., "MmWave Codebook Selection in Rapidly-Varying Channels via Multinomial Thompson Sampling", in Proc. 22nd International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '21), 2021, pp. 151-160, doi: 10.1145/3466772.3467044.

[117] I. Nasim, A. S. Ibrahim, and S. Kim, "Learning-Based Beamforming for Multi-User Vehicular Communications: A Combinatorial Multi-Armed Bandit Approach", IEEE Access, vol. 8, pp. 219891-219902, 2020, doi: 10.1109/ACCESS.2020.3043301.

[118] J. Dou, X. Liu, S. Qie, J. Li, and C. Wang, "Spectrum Allocation and User Scheduling Based on Combinatorial Multi-Armed Bandit for 5G Massive MIMO", Sensors, vol. 23, no. 17, Art. no. 7512, 2023, doi: 10.3390/s23177512.

[119] F. Pase, M. Giordani, S. Cavallero, M. Schellmann, J. Eichinger, R. Verdone, and M. Zorzi, "A Distributed Neural Linear Thompson Sampling Framework to Achieve URLLC in Industrial IoT", IEEE Transactions on Wireless Communications, vol. 25, pp. 12996-13010, 2026.

[120] S. Malta, P. Pinto, and M. Fernandez-Veiga, "Using Reinforcement Learning to Reduce Energy Consumption of Ultra-Dense Networks With 5G Use Cases Requirements", IEEE Access, vol. 11, pp. 5417-5428, 2023, doi: 10.1109/ACCESS.2023.3236980

[121] M. Hoffmann and P. Kryszkiewicz, "Reinforcement Learning for Energy-Efficient 5G Massive MIMO: Intelligent Antenna Switching", IEEE Access, vol. 9, pp. 130329-130339, 2021, doi:

10.1109/ACCESS.2021.3113461.

[122] M. H. M. Elsayed and M. Erol-Kantarci, "Radio Resource and Beam Management in 5G mmWave Using Clustering and Deep Reinforcement Learning", in Proc. IEEE Global Communications Conference (GLOBECOM), 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9322401.

[123] O. C. K. C. a. M. B. A. Sebbar, “Real-time anomaly detection in SDN architecture using integrated SIEM and machine learning for enhancing network security,” *GLOBECOM,* pp. 1795-8000, 2023.

[124] N. Piovesan, D. López-Pérez, A. De Domenico, X. Geng, H. Bao, and M. Debbah, “Machine learning and analytical power consumption models for 5g base stations,” IEEE Communications Magazine, vol. 60, no. 10, pp. 56–62, 2022.

[125] N. Piovesan, D. López-Pérez, A. De Domenico, X. Geng, and H. Bao, “Power consumption modeling of 5g multi-carrier base stations: A machine learning approach,” in ICC 2023-IEEE International Conference on Communications, IEEE, 2023.

[126] K. Bandara, C. Bergmeir, and S. Smyl, “Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach,” Expert systems with applications, vol. 140, p. 112896, 2020.

[127] A. L´opez-Oriona, P. Montero-Manso, and J. A. Vilar, “Time series clustering based on prediction accuracy of global forecasting models,” Knowledge-Based Systems, p. 113649, 2025.

[128] M. L. Bartsioka, A. Giannopoulos, and S. Spantideas, “Intelligent Dynamic Handover via AI-assisted Signal Quality Prediction in 6G Multi-RAT Networks,” arXiv.org, 2025. https://arxiv.org/abs/2510.14832.

[129] N. Piovesan, D. López-Pérez, A. De Domenico, X. Geng, H. Bao, and M. Debbah, “Machine learning and analytical power consumption models for 5g base stations,” IEEE Communications Magazine, vol. 60, no. 10, pp. 56–62, 2022.

[130] L. M. Contreras, G. S. Illán and J. V. Martínez, "Decarbonization Level Agreements in the Provision of Transport Network Slices," 2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN), Paris, France, 2025, pp. 69-73, doi: 10.1109/ICIN64016.2025.10943084.

[131] D. de la Osa Mostazo, P. Armingol Robles, Ó. G. de Dios and J. Pedro Fernández-Palacios Giménez, "Lessons learned from IP routers power measurements and characterization," 2024 15th International Conference on Network of the Future (NoF), Castelldefels, Spain, 2024, pp. 245-253, doi: 10.1109/NoF62948.2024.10741444

[132] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus, “High-dimensional multivariate forecasting with low-rank gaussian copula processes,” Advances in neural information processing systems, vol. 32, 2019.

[133] B. Lim, S. ¨O. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” International Journal of Forecasting, vol. 37, no. 4, pp. 1748–1764, 2021.

[134] Hsieh, T.-Y., Wang, S., Sun, Y., & Honavar, V. (2021). Explainable multivariate time series classification. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 607–615.

[135] N. Piovesan, D. López-Pérez, A. De Domenico, X. Geng, and H. Bao, “Power consumption modeling of 5g multi-carrier base stations: A machine learning approach,” in ICC 2023-IEEE International Conference on Communications, IEEE, 2023.

[136] D. López-Pérez, A. De Domenico, N. Piovesan, and M. Debbah, “Data-driven energy efficiency modelling in large-scale networks: An expert knowledge and ml-based approach,” IEEE Transactions on Machine Learning in Communications and Networking, 2024.

[137] D. López-Pérez, A. De Domenico, N. Piovesan, G. Xinli, H. Bao, Qitao, and M. Debbah, “A survey on 5g radio access network energy efficiency: Massive mimo, lean carrier design, sleep modes, and machine learning,” IEEE Communications Surveys & Tutorials, vol. 24, no. 1, 2022.

[138] A. Ghasemi and M. Hashemi, “Enhancing time-series forecasting with age of information and data relevance in task-oriented, energy-efficient wireless communication,” IEEE Transactions on

Green Communications and Networking, vol. 10, pp. 2738–2750, 2026, doi: 10.1109/TGCN.2026.3684517.

[139] J. Luo and N. Pappas, "On the role of age and semantics of information in remote estimation of Markov sources," IEEE Transactions on Communications, vol. 74, pp. 7406–7419, 2026, doi: 10.1109/TCOMM.2026.3681576.

[140] A. Tahenni, M. A. Ouameur, M. Bagaa, D. Massicotte, S. Salmi, F. A. Pereira de Figueiredo, and A. Ksentini, "A survey on 6G and O-RAN intelligence: Semantic protocols, protocol learning, and AI-enabled semantic protocols," Computer Networks, vol. 279, p. 112143, 2026, doi: 10.1016/j.comnet.2026.112143.

[141] Puligheddu, C., Ashdown, J., Chiasserini, C. F., & Restuccia, F. (2023). SEM-O-RAN: Semantic O-RAN slicing for mobile edge offloading of computer vision tasks. *IEEE Transactions on Mobile Computing*, *23*(7), 7785-7800.

[142] Li, P., & Aijaz, A. (2023, November). Open RAN meets semantic communications: A synergistic match for open, intelligent, and knowledge-driven 6G. In *2023 IEEE Conference on Standards for Communications and Networking (CSCN)* (pp. 87-93). IEEE.

[143] Chatterjee, A., Mandal, D. R., Dutta, S., & Das, G. (2025, November). Semantic-Aware MAC Scheduling for XR Traffic in 6G Networks. In *2025 IEEE 35th International Telecommunication Networks and Applications Conference (ITNAC)* (pp. 1-6). IEEE.

[144] X. Chen, D. Feng, W. Jiang, Q. Luo, G. Chen, and Y. Sun, "QoE-driven multi-task offloading for semantic-aware edge computing systems," IEEE Transactions on Network Science and Engineering, vol. 13, pp. 7100–7117, 2026, doi: 10.1109/TNSE.2026.3662899.

[145] E. Zeydan, L. Blanco, C. J. Vaca-Rubio, M. Caus, and K. Dev, "Semantic non-terrestrial communications with Open RAN-enabled 6G," IEEE Communications Standards Magazine, vol. 9, no. 4, pp. 72–78, 2025, doi: 10.1109/MCOMSTD.2025.3606165.

[146] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep Joint Source-Channel Coding for Wireless Image Transmission," IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 3, pp. 567–579, 2019, doi: 10.1109/TCCN.2019.2919300.

[147] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep Learning Enabled Semantic Communication Systems," IEEE Transactions on Signal Processing, vol. 69, pp. 2663–2675, 2021, doi: 10.1109/TSP.2021.3071210.

[148] L. Yan, Z. Qin, C. Li, R. Zhang, Y. Li, and X. Tao, "QoE-Based Semantic-Aware Resource Allocation for Multi-Task Networks," IEEE Transactions on Wireless Communications, vol. 23, no. 9, pp. 11958–11971, 2024, doi: 10.1109/TWC.2024.3386807.

[149] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1376–1411, 2023, doi: 10.1109/COMST.2023.3239220.

[150] X. Hou, J. Zhang, M. Budagavi, and S. Dey, "Head and body motion prediction to enable mobile VR experiences with low latency," in 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–7.

[151] Qualcomm Technologies, Inc., "Making Immersive Virtual Reality Possible in Mobile," Qualcomm Technologies, Inc., San Diego, CA, USA, White paper, Mar. 2016. [Online]. Available: https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/whitepaper-makingimmersivevirtualrealitypossibleinmobile.pdf

[152] Huawei iLab, "Cloud VR Network Solution White Paper," Huawei Technologies Co., Ltd., White Paper, 2018. [Online]. Available: https://www-file.huawei.com/-/media/corporate/pdf/ilab/2018/cloud_vr_network_solution_white_paper_2018_en_v1.pd

[153] GSMA, "Cloud AR/VR Whitepaper," GSMA, White Paper, Apr. 26, 2019. [Online]. Available: https://www.gsma.com/solutions-and-impact/technologies/networks/gsma_resources/cloud-ar-vr-whitepaper-2/

[154] Y. Zhang, Z. Jia, L. Jiang, Q. Li, X. Zhang, and Z. Guo, "Modeling virtual reality traffic with head movement in remote rendering," in ICC 2025—IEEE International Conference on Communications, IEEE, 2025, pp. 1–6.

[155] 3GPP, “Service requirements for the 5G system; Stage 1,” Technical Specification TS 22.261, Release 18, 2025

[156] Zeydan, Engin, et al. "Semantic non-terrestrial communications with open ran-enabled 6g." *IEEE Communications Standards Magazine* (2025).

[157] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.

[158] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation,” *International Journal of Computer Vision*, vol. 129, pp. 3051–3068, 2021.

[159] P. B. D. Prever, S. D’Oro, L. Bonati, M. Polese, M. Tsampazi, H. Lehmann, and T. Melodia, ‘‘PACIFISTA: Conflict evaluation and management in open RAN,’’ IEEE Trans. Mobile Comput., early access, May 20, 2025, doi: 10.1109/TMC.2025.3570632.

[160] C. Adamczyk and A. Kliks, ‘‘Conflict mitigation framework and conflict detection in O-RAN near-RT RIC,’’ IEEE Commun. Mag., vol. 61, no. 12, pp. 199–205, Dec. 2023.

[161] J. Armstrong, E. Fallon, and S. Fallon, ‘‘Pre-emptive conflict detection architecture for O-RAN service management and orchestration,’’ in Proc. IEEE Int. Conf. Ind. 4.0, Artif. Intell., Commun. Technol. (IAICT), Jul. 2024, pp. 335–340.

[162] Wadud, A., Golpayegani, F., & Afraz, N. (2025, May). xApp-Level Conflict Mitigation in O-RAN, a Mobility Driven Energy Saving Case. In IEEE INFOCOM 2025-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 1-6). IEEE.

[163] A. Zolghadr, J. F. Santos, L. A. DaSilva, and J. Kibilda, ‘‘Learning and reconstructing conflicts in O-RAN: A graph neural network approach,’’ in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2025, pp. 01–06.

[164] H. Zhang, H. Zhou, and M. Erol-Kantarci, ‘‘Team learning-based resource allocation for open radio access network (O-RAN),’’ in Proc. IEEE Int. Conf. Commun. (ICC), May 2022, pp. 4938–4943.

[165] G. R. a. P. O. J. Dromard, “Online and scalable unsupervised network anomaly detection method,,” *IEEE TNSM Journal,* vol. 14, pp. 34-47, 2016.

[166] F. S. A. B. N. A. E. a. A. S. M. Injadat, “Bayesian optimization with machine learning algorithms towards anomaly detection,” *GLOBECOM,* 2018.

[167] ETSI, “Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 1: Enablers,” ETSI, Sophia Antipolis, France, 2021.

[168] ETSI, “Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 3: Advanced topics,” ETSI, Sophia Antipolis, France, 2023.

[169] F. Javed and J. Mangues-Bafalluy, “Blockchain-based SLA management for 6G networks,” *Internet Technology Letters,* vol. 5, no. e742, p. 7, 2024.

[170] T. Faisal, M. Dohler, S. Mangiante and D. R. Lopez, “How to Design Autonomous Service Level Agreements for 6G,” *arXiv preprint arXiv:2204.03857,* 2022.

[171] A. Storhaug, J. Li and T. Hu, “Efficient Avoidance of Vulnerabilities in Auto-completed Smart Contract Code Using Vulnerability-constrained Decoding,” in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, 2023.

[172] A. Rasti, A. A. Anda, S. Alfuhaid, A. Parvizimosaed, D. Amyot, M. Roveri, L. Logrippo and J. Mylopoulos, “Automated generation of smart contract code from legal contract specifications with Symboleo2SC,” *Software and Systems Modeling,* vol. 24, no. 4, p. 1127–1156, 2025.

[173] L. Blanco, S. Via, C. J. Vaca-Rubio, E. Zeydan, L. M. C. Murillo, G. Kedar, G. R. Antonio, A. Antonopoulos, J. Haxhibeqiri, V. Maglogiannis et al., “UNITY-6G: UNified archITecture for Open RAN enabled Distributed, Scalable and SustainabilitY enhanced 6G Networks,” in 2025 Joint European

[174] M. A. Shami, J. Yan, and E. T. Fapi, “O-RAN xApps Conflict Management using Graph Convolutional Networks,” 2025, doi:10.48550/arXiv.2503.03523. [Online]. Available:

https://arxiv.org/abs/2503.03523

[175] A. E. Giannopoulos, S. T. Spantideas, G. Levis, A. S. Kalafatelis, and P. Trakadas, “COMIX: Generalized Conflict Management in O-RAN xApps—Architecture, Workflow, and a Power Control Case,” IEEE Access, vol. 13, pp. 116 684–116 700, 2025.

[176] ORANGE, “3GPP workshop on 6G,” Feb 27, 2025.

[177] M. Goel, “O-RAN (Open Radio Access Networks), Market size, Share, Growth Insights, and Forecast 2025-2032,” October 2025.

[178] Raksha Sharma, “Open RAN Market”, 2025, Report ID: ICT-SE-719288,” pp. https://dataintelo.com/report/open-ran-market

[179] R. Sharma, “O-RAN Test and Integration Market Research Report 2033,” Report ID :ICT-SE-87855, https://growthmarketreports.com/report/o-ran-test-and-integration-market, 2025.

[180] P. Newswire, “6G RAN to Approach $30 B by 2033,” https://www.prnewswire.com/news-releases/6g-ran-to-approach-30-b-by-2033-according-to-delloro-group-302272454.html .

[181] S. JU, “6G Green Network Technologies Roadmap,” 2025.

[182] X. Wang, T. Zhiqing, G. Jianxiong, M. Tianhui, W. Chenhao, W. Tian and J. Weijia, “Empowering edge intelligence: A comprehensive survey on on-device AI models.,” *ACM computing Surveys,* vol. 57, no. 9, pp. 1-39, 2025.

[183] Commission European, “Towards Secure and Open 6G Networks,” 2025.

[184] A. S. Fischio, K. Adrian, A.-T. Ahmed, E. Ahmed, N. Alexandros, M. Ali, M. Ali, K. Amirreza, D. D. Antonio, K. Athanasios, C. Bo, Y. Bo, W. Bohao and Carlo, “Large-Scale AI in Telecom: Charting the Roadmap for Innovation, Scalability, and Enhanced Digital Experiences,” *white paper,* 2025.sa

[185] Global, TN, “Open RAN Set to Capture 6-8% of Total RAN Market by End of 2024, Fueled by Initiatives from Fujitsu, Samsung, Rakuten Symphony, and Mavenir,” https://technode.global/prnasia/open-ran-set-to-capture-6-8-of-total-ran-market-by-end-of-2024-fueled-by-initiatives-from-fujitsu-samsung-rakuten-symphony-and-mavenir/, 2024.

[186] Group, FCC Open RAN Working, “Security Recommendations for Open Networks,” 2023.

[187] IA, 6G, “Strategic Research and Innovation Agenda (SRIA 2.0),” 2024.

[188] F. M. and al, “Economics of Open RAN Deployments,” *IEEE Commun. Standards Mag,* vol. 8, no. 2, pp. 12-21, 2024.

[189] Intelligence, Mordor, “Open RAN Market Size & Share Analysis - Growth Trends & Forecasts (2025 - 2030),” : https://www.mordorintelligence.com/industry-reports/open-ran-market.

[190] Research, Grand View, “Open RAN Market (2025 - 2030),” Report ID: GVR-4-68040-117-3, https://www.grandviewresearch.com/industry-analysis/open-ran-market-report, 2025.

[191] O-RAN ALLIANCE , “Circular economy guidelines on network equipment (O-RAN.SuFG.CE-v01.00),” 2022.

[192] 5G PPP Test, Measurement and KPIs Validation Working Group, “Beyond 5G/6G KPIs and Target Values,” 2022.

[193] D. Salinas et al., “*DeepAR: Probabilistic forecasting with autoregressive recurrent networks*”, International Journal of Forecasting, Volume 36, Issue 3, 2020.

[194] T. Taleb, I. Afolabi, K. Samdanis and F. Z. Yousaf, "On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control," in *IEEE Network*, vol. 33, no. 5, pp. 242-252, Sept.-Oct. 2019.

[195] C. Balachandran, P. A. C, G. Ramachandran and B. Krishnamachari, "EDISON: A Blockchain-based Secure and Auditable Orchestration Framework for Multi-domain Software Defined Networks," *2020 IEEE International Conference on Blockchain (Blockchain)*, Rhodes, Greece, 2020, pp. 144-153.

[196] Z. A. El Houda, H. Moudoud and L. Khoukhi, "Blockchain Meets O-RAN: A Decentralized Zero-Trust Framework for Secure and Resilient O-RAN in 6G and Beyond," *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, BC,

Canada, 2024, pp. 1-6.

[197] A. Waqas, H. Mahmood and N. Saeed, "Interference Aware Cooperative Routing for Edge Computing-Enabled 5G Networks," in IEEE Sensors Journal, vol. 22, no. 4, pp. 3777-3784, 15 Feb.15, 2022.

[198] M. Elkael *et al.*, "AgentRAN: An Agentic AI Architecture for Autonomous Control of Open 6G Networks," in *IEEE Communications Magazine*.

[199] M. B. Taha, Y. Sanjalawe, A. Al-Daraiseh, S. Fraihat and S. R. Al-E'mari, "Proactive Auto-Scaling for Service Function Chains in Cloud Computing Based on Deep Learning," in IEEE Access, vol. 12, pp. 38575-38593, 2024, doi: 10.1109/ACCESS.2024.3375772

[200] Marques, G., Senna, C., Sargento, S., Carvalho, L., Pereira, L., & Matos, R. (2024). "Proactive resource management for cloud of services environments. Future Generation Computer Systems", 150, 90-102.

[201] C. -N. Nhu and M. Park, "Dynamic Network Slice Scaling Assisted by Attention-Based Prediction in 5G Core Network," in *IEEE Access*, vol. 10, pp. 72955-72972, 2022.

[202] D. Dulas, J. Witulska, A. Wyłomańska, I. Jabłoński and K. Walkowiak, "Data-Driven Model for Sliced 5G Network Dimensioning and Planning, Featured With Forecast and "what-if" Analysis," in *IEEE Access*, vol. 12, pp. 50067-50082, 2024.

[203] D. Bega, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Paris, France, 2019, pp. 280-288.

[204] H. MQ and et al., "Recent advances in machine learning for network automation in the o-ran," MDPI Sensors. 2023; 23(21):8792, 2023.