D4.4: Preliminary report on AI-based radio resource management, RAN softwarisation and onboard processing

Revision: v1.0F

Work package	WP 4
Task	Task 4.3, Task 4.4 and Task 4.5
Due date	30/06/2024
Submission date	30/06/2024
Deliverable lead	CTTC
Version	V1.0F
Authors	Luis Blanco, Cristian J. Vaca-Rubio, Marius Caus, Musbah Shaat (CTTC) Justin Tallon (SRS) Nathan Borios, Fabrice Arnal, Mathieu Gineste (TAS-F) LI Zheng (ORANGE)
Reviewers	Eddi González (HSP), Stefano Cioni (ESA)
Abstract	This document reports the initial outcomes of Tasks 4.3 “Data-enhanced Radio Intelligent Controller Design”, Task 4.4 “Onboard Processing and Communication Capabilities” and Task 4.5 “RAN Softwarisation”. Building upon the O-RAN architecture and interfaces, radio resource management algorithms are conceived for near- and non-real time timescales. On the basis of a fully regenerative payload, the document provides a high-level description of the on-board processing capabilities. Finally, to enable the proof-of-concept, the required modifications to the gNB radio protocol and the inter-block communication have been investigated. The final outcomes on these activities will be reported in D4.7 in 2025.

Co-funded by the European Union 6GSNS
Grant Agreement No.: 101096573 Topic: HORIZON-JU-SNS-2022-STREAM-A-01-02
Call: HORIZON-JU-SNS-2022 Type of action: HORIZON-JU-RIA

Keywords	O-RAN, RRM, non-RT RIC, near-RT RIC, NTN, TN, OBP

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	01/04/2024	First draft: ToC definition	CTTC, CNIT, TAS, SRS
V0.2	16/05/2024	ORAN architecture and interfaces, time series forecasting	ORA, CTTC
V0.3	12/06/2024	Near-RT RIC	CNIT
V0.4	21/06/2024	Onboard processing and RAN softwarisation	SRS, TAS
V0.5	26/06/2024	Draft version for QA review	CTTC
V0.6	29/06/2024	QA review	HSP
V0.7	02/07/2024	AB review	ESA
V0.8	04/07/2024	Final version revised according to AB and QA reviews	CTTC
V1.0F	05/07/2024	Approved final version for EC portal submission	DLR

DISCLAIMER

5G-STARDUST (*Satellite and Terrestrial Access for Distributed, Ubiquitous, and Smart Telecommunications*) 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 101096573.

COPYRIGHT NOTICE

Project co-funded by the European Commission in the Horizon Europe Programme	
Nature of the deliverable:	**R**
Dissemination Level	

PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

EXECUTIVE SUMMARY

In this deliverable, 5G-STARDUST investigates aspects related to radio resource management (RRM), on-board processing (OBP) and radio access network (RAN) softwarisation. This is the intermediate version based on the work carried out within WP4in the first half of the project and will be further extended and concluded in the final release (D4.7) planned for the next year (2025).

The first part of the document is devoted to introducing a new paradigm for the RAN design and operation. The result is an open RAN (O-RAN) architecture that disaggregates network components, offering the possibility of integrating hardware and software from multiple vendors. This aspect allows to overcome the limited flexibility and reconfigurability of monolithic units. Within the O-RAN context, the components are connected via open interfaces and can be optimized by the RAN intelligent controller (RIC). Hence, O-RAN natively embeds intelligence in the RAN, which can provide an enormous value for RRM. Remarkably, O-RAN is specified on top of the 3GPP. Hence, it offers a framework to apply data-driven methods to manage the 3GPP-defined RAN.

To understand how artificial intelligence (AI) can be used for efficient RRM in the architecture defined in the Deliverable D3.2 [1], the document initially describes the machine learning (ML) framework offered by O-RAN. Especial emphasis is given to network functions that host the training and the ML algorithm as well as the control loops defined in O-RAN. A detailed discussion is devoted to the application of supervised, unsupervised and reinforcement learning.

To identify the class of RRM tasks that can be tackled through AI algorithms, an exhaustive analysis has been conducted across several sections of the document. The problems of interest can be divided into:

- Traffic prediction and offloading;
- Traffic anomaly detection;
- Power and bandwidth allocation;
- Radio resource allocation;
- Traffic Steering.

In alignment with the O-RAN specification, this deliverable classifies the problems according to the timescale. The entity that supports large time scale RAN optimization (>1s) is the non-real-time (non-RT) RIC. In this case, the software application that is designed to run on the non-RT RIC is referred to as rApp. For smaller timescales (between 10ms and 1 s), the RAN optimization relies on the near-real-time (near-RT) RIC. In this context, the xApps are applications that are designed to run on the near-RT RIC.

It is important to remark that the O-RAN architecture has been originally conceived from the TN perspective. To enable the non-terrestrial network (NTN) functionalities, it shall be noted that some modifications are required. As a first iteration, this deliverable analyses the gNB radio protocol enhancements and inter block connections that will be needed to integrate NTN in the O-RAN architecture.

To assess the potential gains of data-driven RRM in the context of NTN, an in-depth analysis is needed. Essentially, to evaluate the impact the specific characteristics of satellite-based communications may have on the performance results. For instance, the extensive coverage area and the short visibility windows, which are inherent to satellite systems. In this regard, an innovative solution has been conceived to enhance predictive models. The proposed solution leverages on the Kolmogorov-Arnold networks (KANs) to forecast satellite traffic. Numerical results obtained with the satellite network data set reveal that lower error metrics have been achieved with lower computational resources, when compared to the traditional solutions. Focusing on large timescales, the result is relevant for the residential broadband scenario, where the traffic load is low in night hours. Then, on the basis that users can access the network either through TN or NTN links, AI-based traffic predictions can be used to decide when NTN comes into play to provide coverage. For instance, when the traffic forecast is too low. When a given condition is met, TN cells or carriers could be switched off, so that all the traffic is diverted to the satellite. The immediate consequence is that the RAN consumption of the terrestrial segment can be reduced.

Drawing the attention to small timescales, the deliverable provides guidelines to tackle AI-based RRM in the context of user-centric beamforming. Indeed, the problem can be formulated to either maximize or minimize the objective function. The cost function to be optimized include, system capacity, per-user throughput and user fairness, to mention a few. The next iteration of this deliverable, from the near-RT RIC side, will assess the performance of an AI-based user scheduling for user-centric beamforming in NTN.

Finally, the document has conducted a high-level feasibility study on the development of regenerative payloads with OBP capabilities. Drivers and challenges are discussed. The functionalities and the network elements that should be hosted by the satellite in different network architectures are defined. This outcome is especially relevant from the RRM standpoint, as it would enable the possibility to place the RIC on-board of the satellite. This approach aims to reduce latency and optimize bandwidth, by reducing the number of interactions with on-ground equipment.

TABLE OF CONTENTS

Disclaimer 2
Copyright notice 2
1 INTRODUCTION 14
2 O-RAN ARCHITECTURE AND INTERFACES 16
2.1 O-RAN components 16
2.2 O-RAN interfaces 17
3 ML AND ORAN 18
3.1 ML Algorithms in O-RAN Architecture 18
3.1.1 Supervised learning 19
3.1.2 Unsupervised learning 19
3.1.3 Reinforcement learning 20
3.2 Mapping AI/ML functionalities into O-RAN Control Loops 20
3.3 AI flows in O-RAN Architecture 21
4 AI-BASED RADIO RESOURCE MANAGEMENT 23
4.1 Non-real time RIC 23
4.1.1 Non-real time RIC description 23
4.1.2 Radio resource management problems to be handled 24
4.1.3 AI-techniques for non-real time RRM 25
4.2 Near-real time controller 39
4.2.1 Near-real time RIC description 39
4.2.2 Radio resource management problems to be handled 41
4.2.3 AI-techniques for near-real time RRM 44
5 RAN SOFTWARIZATION 46
5.1 General Stack modifications to enable PoC 46
5.1.1 RIC – RAN interface to enable monitoring and control 46
5.1.2 API creation for RU integration at desired split level 47
5.1.3 RAN modifications for NTN 49
5.1.4 Development of 60KHz SCS to enable FR2 49
6 ONBOARD PROCESSING 51
6.1 Introduction, context & drivers 51
6.2 CU/DU splitting 53
6.3 Full gNB on-board 55
6.4 Full gNB on-board + UPF 56
7 CONCLUSIONS AND NEXT STEPS 58
8 REFERENCES 60

LIST OF FIGURES

FIGURE 1. O-RAN ARCHITECTURE 16
FIGURE 2 ML FRAMEWORK IN O-RAN – GENERAL PROCEDURE 18
FIGURE 3 SUPERVISED LEARNING MODEL TRAINING AND ACTOR LOCATIONS 19
FIGURE 4 UNSUPERVISED LEARNING MODEL TRAINING AND ACTOR LOCATIONS 20
FIGURE 5 REINFORCEMENT LEARNING MODEL TRAINING AND ACTOR LOCATIONS 20
FIGURE 6 CONTROL LOOPS IN O-RAN 21
FIGURE 7: ML MODEL LIFECYCLE EXAMPLE 22
FIGURE 18: RESIDENTIAL BROADBAND SCENARIO 24
FIGURE 8: EXAMPLE OF NORMALIZED SATELLITE TRAFFIC SERIES DATA WITH THE CONDITIONING AND PREDICTION LENGTHS DENOTED IN BLUE, AND RED, RESPECTIVELY. 27
FIGURE 9: EXAMPLE OF THE FLOW OF INFORMATION IN THE KAN NETWORK ARCHITECTURE FOR OUR TRAFFIC FORECASTING TASK. LEARNABLE ACTIVATIONS ARE REPRESENTED INSIDE A SQUARE BOX. 30
FIGURE 10: SATELLITE TRAFFIC OVER THREE DIFFERENT BEAMS WIT FORECASTED VALUES OVER BEAM 1 USING A 4-DEPTH KAN AND A 4-DEPTH MLP. 31
FIGURE 11: SATELLITE TRAFFIC OVER THREE DIFFERENT BEAMS WIT FORECASTED VALUES OVER BEAM 2 USING A 4-DEPTH KAN AND A 4-DEPTH MLP. 31
FIGURE 12: SATELLITE TRAFFIC OVER THREE DIFFERENT BEAMS WIT FORECASTED VALUES OVER BEAM 3 USING A 4-DEPTH KAN AND A 4-DEPTH MLP. 32
FIGURE 13: AVERAGE PRB FORECASTS FOR THE BEAM 36. 35
FIGURE 14: AVERAGE PRB FORECASTS FOR BEAM 37. 35
FIGURE 15: PRB FORECASTS FOR BEAM 38. 36
FIGURE 16: PRB DEMANDS FOR BEAM 39 (HISTORICAL AND PREDICTED). 36
FIGURE 17: ABLATION COMPARISON OF KAN-SPECIFIC PARAMETERS. 38
FIGURE 19: NEAR-RT RIC INTERNAL ARCHITECTURE. 39
FIGURE 20: HIGH-LEVEL ARCHITECTURE OF E2 AGENT IN SRSRAN. 47
FIGURE 21: OFH DATAFLOWS 48
FIGURE 22: INTER-CARRIER INTERFERENCE 50
FIGURE 23: PAYLOAD AND GROUND ARCHITECTURE - SPLIT 2 (DU/RU ON BOARD, CU ON GROUND) 54
FIGURE 24: PAYLOAD AND GROUND ARCHITECTURE - GNB ON BOARD AND UPF/CN ON GROUND 55
FIGURE 25: PAYLOAD AND GROUND ARCHITECTURE - GNB + UPF ON BOARD AND CORE NETWORK ON GROUND 56

LIST OF TABLES

TABLE 1: MODEL CONFIGURATIONS FOR SATELLITE TRAFFIC FORECASTING 32
TABLE 2:RESULTS SUMMARY 34
TABLE 3. PRB PREDICTION RESOURCE SUMMARY 37
TABLE 4: EARTH MOVING BEAMS VS EARTH FIXED BEAMS COMPARISON 52

ABBREVIATIONS

5GC	5G Core network
AAC	Advantage Actor Critic
AF	Application Function
AGA	Adaptive Genetic Algorithm
AR	AutoRegressive
ARIMA	AutoRegressive Integrated Moving Average
ARQ	Automatic Repeat Request
AUC-RC	Area Under the Curve of the Receiver Operating Characteristic
CFO	Carrier Frequency Offset
CM	Configuration Management
CN	Core Network
CNN	Convolutional Neural Network
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CSI	Channel State Information
CU	Central Unit
CUS	Control User Synchronization
DL	Deep Learning
DNN	Deep Neural Network
DQN	Deep-Q-Network
DRL	Deep Reinforcement Learning
DU	Distributed Unit
E2AP	E2 Application Protocol
E2SM	E2 Service Model
eCPRI	Enhanced Common Public Radio Interface
eMBB	enhanced Mobile BroadBand
FCAPS	Fault Configuration Accounting Performance Security

FFT	Fast Fourier transform
FR1	Frequency Range 1
FR2	Frequency Range 2
GEO	Geostationary Earth Orbit
GRU	Gated Recurrent Units
HARQ	Hybrid Automatic Repeat Request
HO	Hand Over
IAB	Integrated Access Backhauling
IQ	In-phase and Quadrature
KAN	Kolmogorov-Arnold Network
KPM	Key Performance Measurement
LAA	License Assisted Access
LBT	Listen-Before-Talk
LDPC	Low Density Parity Check
LEO	Low Earth Orbit
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptrons
MAC	Medium Access Control
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MIH	ML Inference Host
MSE	Mean Squared Error
MTH	ML Training Host
Near-RT	Near-Real-Time
NIB	Network Information Based
Non-RT	Non-Real-Time
OBP	On-Board Processing

O-CU	O-RAN Central Unit
O-CU-CP	O-RAN Central Unit - Control Plane
O-CU-DP	O-RAN Central Unit - Data Plane
O-DU	O-RAN Distributed Unit
OFH	Open FrontHaul Interface
O-RAN	Open Radio Access Network
PAPR	Peak-to-Average-Power Ratio
PCA	Principal Component Analysis
PDCP	Packet Data Convergence Protocol
PHY	Physical Layer
PoC	Proof-of-Concept
PPO	Parametrized Policy Optimization
PRACH	Physical Random Access Channel
PRB	Physical Resource Block
QoS	Quality of Service
QoE	Quality of Experience
RAN	Radio Access Network
RC	Remote Control
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RLC	Radio Link Control
RMSE	Root Mean Squared Error
RRC	Radio Resource Control
RRM	Radio Resource Management
RT	Real-Time
RU	Radio Unit
SARIMA	Seasonal AutoRegressive Integrated Moving Average

SCS	Subcarrier Spacing
SDAP	Service Data Adaptation Protocol
SDL	Shared Data Layer
SEC	Space Edge Computing
SINR	Signal-to-Interference-plus-Noise Ratio
SIR	Signal-to-Interference Ratio
SMO	Service Management and Orchestration
SNR	Signal-to-Noise Ratio
SO	Satellite Operator
TCN	Temporal Convolutional Network
TTI	Transmission Time Unit
UE	User Equipment
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communication
XGBoost	Extreme Gradient Boosting

1 INTRODUCTION

This deliverable D4.4 reports the outcomes of Task 4.3 “Data-enhanced Radio Intelligent Controller Design”, Task 4.4 “Onboard Processing and Communication Capabilities” and Task 4.5 “RAN Softwarisation”. The activities undertaken within the scope of these tasks, pursue the objective to:

- Define, design and analyse data-driven management system components;
- Study, design, and analyse a 5G-based satellite network, implementing onboard processing and storage capabilities;
- Define, design, and analyse a full softwarisation of the end-to-end network architecture.

Towards this end, 5G-STARDUST leverages on:

- The architecture defined in WP3 in [1, 2];
- The datasets described in WP4 and WP5 in [3] and [4], respectively;
- The proof-of-concept (PoC) functional architecture defined in [5].

The data sets have been divided into two parts, i.e., cellular and satellite network data sets. The inputs provided by the satellite operator are related to broadband services for fixed clients (residential broadband) and for mobile communications (maritime communications). The inputs of the cellular data set are related to the rural and the railway environments.

Data-driven RRM, which is one of the pillars of the project, is a remarkable feature that seeks to boost network performance by learning trends and patterns. For instance, by optimizing how the RAN nodes operate over time. To accomplish this goal, the ML framework defined by the O-RAN architecture becomes a powerful tool. More precisely, thanks to the support of RIC-hosted x/rApps, which become instrumental to develop AI engines for RRM tasks. Nonetheless, the specific characteristics of NTN, prevent the existing solutions conceived for TN from being reused. The large coverage area and the short visibility period of RAN nodes, which are inherent to satellite systems, call for new solutions. Across the following sections, RRM enhancements are proposed for NTN:

- Section 2 defines the components and the interfaces of the O-RAN architecture.
- Section 3 provides a view of how the different ML algorithms can be deployed and realized in O-RAN architecture and how to map AI and ML functionalities into the O-RAN control loops.
- Section 4 seeks to apply the framework offered by the O-RAN architecture in an integrated TN and NTN architecture. The section identifies the RRM problems that could be handled by the non-RT RIC and the near-RT RIC. With regard to the design of the algorithms, an innovative AI-based solution is proposed to solve the problems that involve a forecasting task. Numerical results are provided to show that the proposed design clearly outperforms traditional methods in terms of error metrics with less computational resources. The analysis of the application scope highlights that the algorithm described in this section could be used to offload the traffic from TN to NTN. In the context of user-centric beamforming, guidelines are provided to optimize the user scheduling.

- Section 5 describes the general stack modifications that are needed to the PoC implementation. This includes the implementation of the open fronthaul interface (OFH), new functionalities to enable monitoring and control and gNB radio protocol enhancements to support NTN.

- Section 6 analyses the OBP capabilities of regenerative payloads. For different network architectures, the functions and the network elements that are allocated on-board of the satellite are defined.

- Finally, Section 7 concludes this document.

2 O-RAN ARCHITECTURE AND INTERFACES

This section defines the components and the interfaces of the O-RAN architecture.

O-RAN architecture is a virtualized, software-driven, and open radio access network architecture that enables the integration of hardware and software components from multiple vendors. It is designed to be modular, scalable, and flexible, with standardized interfaces that enable interoperability between different RAN components. In this section, we summarize the O-RAN architecture and interfaces.

The O-RAN architecture consists of multiple functional components that can be separated and managed independently. Some of the key components in O-RAN are radio unit (RU), distributed unit (DU), central unit (CU), RIC, and service management and orchestration (SMO). Figure 1 shows these components and the connection among them.

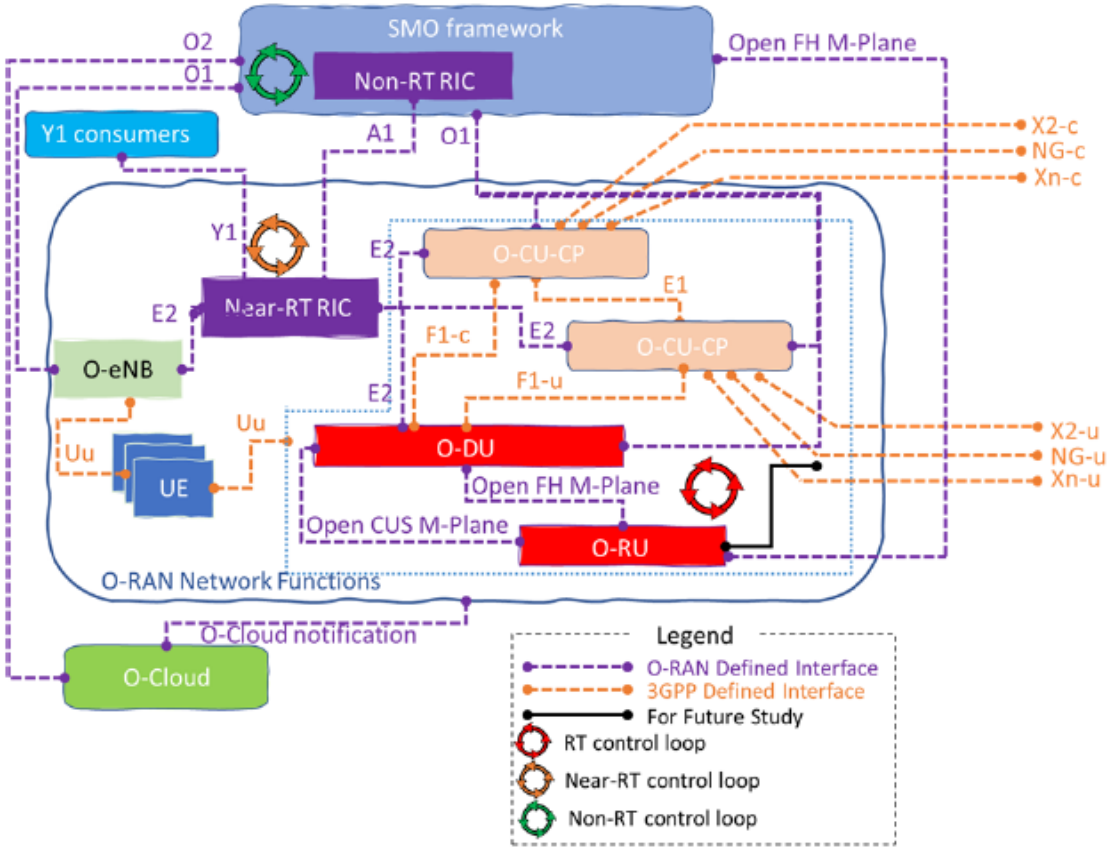

Figure 1. O-RAN Architecture

2.1 O-RAN COMPONENTS

- Service management and orchestration (SMO): the key capabilities of the SMO that provide RAN support in O-RAN are the fault, configuration, accounting, performance, security (FCAPS) interfaces to O-RAN functions, non-real-time (non-RT) RIC for RAN optimization, orchestrator and cloud management. The SMO performs these services through A1/O1/O2 interfaces.

- Non-RT RIC and rApps: this logical function resides within the SMO and provides the A1 interface to the Near-RT RIC. Its main goal is to support large timescale RAN optimization (seconds or minutes), including policy computation, ML model management (e.g., training), and other radio resource management functions within this timescale. The rApps are software applications designed to run on the Non-RT RIC.

- Near-RT RIC and xApps: near-real-time (near-RT) RIC is a logical function that enables near-RT optimization and control and data monitoring of O-RAN central unit (O-CU) and O-RAN distributed unit (O-DU) nodes in near-RT timescales (between 10 ms and 1 s). To this end, near-RT RIC control is steered by the policies and assisted by models computed/trained by the non-RT RIC. The xApps are applications designed to run on the near-RT RIC.

- O-CU - control plane (O-CU-CP): a logical node hosting the radio resource control (RRC) and the control plane part of the packet data convergence control (PDCP) protocol

- O-CU - user plane (O-CU-DP): a logical node hosting the user plane part of the PDCP protocol and the service data adaptation protocol (SDAP).

- O-DU: a logical node hosting radio link control (RLC), medium access control (MAC), high-physical (PHY) layers based on the 7-2x fronthaul split defined by O-RAN

- O-RAN RU (O-RU): a logical node hosting low-PHY layer and radio frequency (RF) processing based on the 7-2x fronthaul split defined by O-RAN.

- O-eNB: an eNB or ng-eNB that supports E2 interface.

- Cloud computing platform: O-Cloud comprising a collection of physical infrastructure nodes that meet O-RAN requirements to host the relevant O-RAN functions (such as Near-RT RIC, O-CU-CP, O-CU-UP, and O-DU), the supporting software components (such as operating system, virtual machine monitor, container runtime, etc.) and the appropriate management and orchestration functions.

2.2 O-RAN INTERFACES

- A1 Interface between Non-RT RIC and Near-RT RIC to enable policy-driven guidance of Near-RT RIC applications/functions, and support AI/ML workflow.

- O1 Interface connecting the SMO to the Near-RT RIC, one or more O-CU-CPs, one or more O-CU-UPs, and one or more O-DUs.

- O2 Interface between the SMO and the O-Cloud

- E2 Interface connecting the Near-RT RIC and one or more O-CU-CPs, one or more O-CU-UPs, one or more O-DUs, and one or more O-eNBs.

- Open fronthaul control user synchronization (CUS) plane interface between O-RU and O-DU

- Open fronthaul M-Plane Interface between O-RU and O-DU as well as in between O-RU and SMO

3 ML AND ORAN

This section provides a view of how the different ML algorithms can be deployed and realized in O-RAN architecture and how to map AI/ML functionalities into O-RAN control loops.

3.1 ML ALGORITHMS IN O-RAN ARCHITECTURE

One of the key aspects of the O-RAN is to natively embed intelligence into the RAN. To this end, AI/ML plays a crucial role in the process. Figure 2 below shows a simplified ML framework and a general procedure for the ML framework operation within O-RAN.

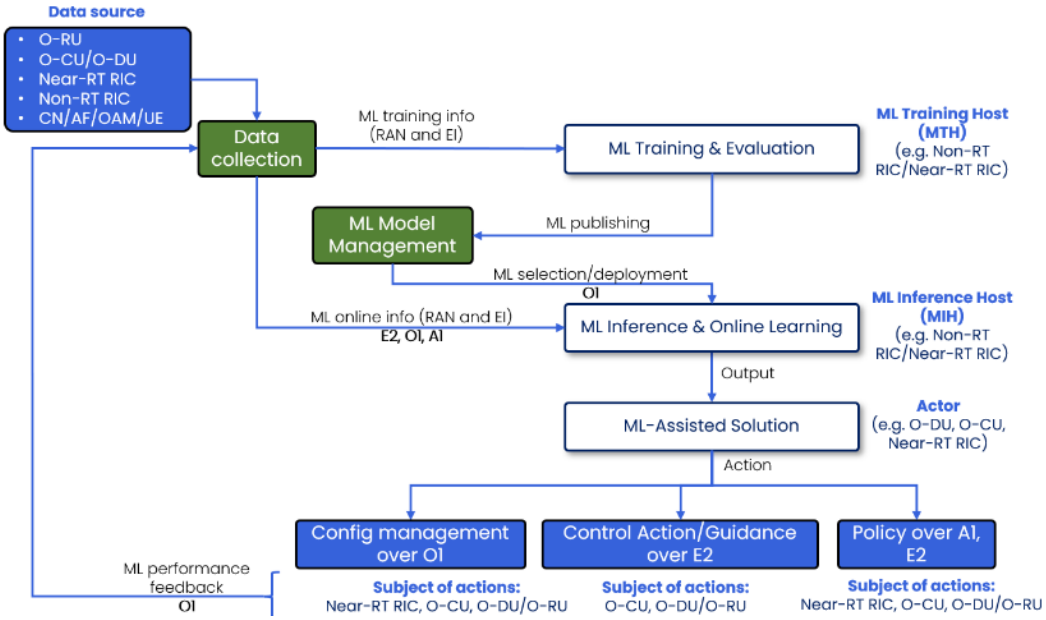

Figure 2 ML Framework in O-RAN – General Procedure

To start with, data is collected through O-RAN interfaces (like O1, E2, or A1) from all of the O-RAN entities, including the O-RU, O-DU, O-CU, near- and non-RT RIC, but also can come from a UE, core network (CN) or application functions (AF). Then the data is used by ML training and inference functions:

- ML training host (MTH), is a network function hosting the online and offline training of the model (typically non-RT RIC is used for this purpose, but also near-RT RIC in some scenarios).

- ML inference host (MIH), is also a network function hosting the ML model during inference mode including model execution and online learning (non- or near-RT RIC can be utilized here).

The inference host provides output to an actor (i.e., an entity hosting an ML-assisted solution. In this case, it can be O-DU, O-CU, non/near-RT RIC). The actor utilizes the results of ML inference for the purpose of RAN performance optimization. Based on the decision, an action is taken on a subject (i.e., an entity or function configured, controlled, or informed by the action). After the action is taken, subjects provide feedback serving as data sources for the next iteration.

Based on the output of the ML model, an ML-assisted solution (i.e., a solution that addresses a specific use case using ML algorithms during operation) informs the actor to take the necessary actions toward the subject. These could include configuration management (CM) changes over O1, policy management over A1, or control actions or policies over E2, depending on the location of the ML inference host and actor.

The location of the ML model components, i.e., ML training and the ML inference for a use case mostly depends on the trade-off between communication delay and computational capabilities of near-RT RIC. Moreover, the availability and quantity of data, available through different O-RAN interfaces should also be taken into account. A detailed discussion for the three types of ML algorithms can be found below.

3.1.1 Supervised learning

Input data is called training data and has a known label or result. Supervised learning is a ML task that aims to learn a mapping function from the input to the output, given a labelled data set. These algorithms include:

- Regression: linear regression, logistic regression;
- Instance-based Algorithms: k-nearest neighbour;
- Decision Tree Algorithms: classification and regression tree;
- Support Vector Machines;
- Bayesian Algorithms: naive Bayes;
- Ensemble Algorithms: extreme gradient boosting (XGBoost), bagging, random forest.

Supervised learning can be further grouped into regression and classification problems. Classification is about predicting a label whereas regression is about predicting a quantity.

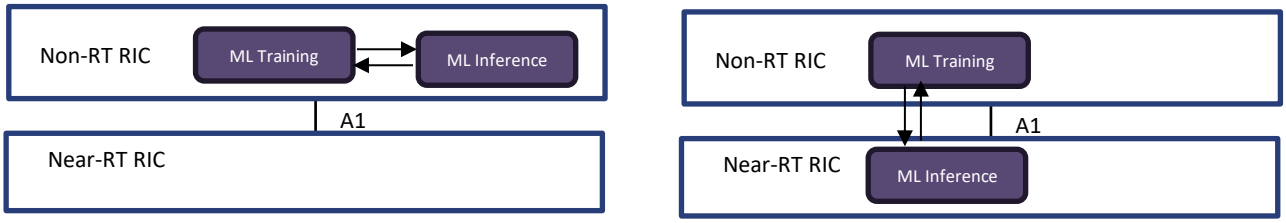

Figure 3 Supervised learning model training and actor locations

In supervised learning (see Figure 3), Non-RT RIC is part of the SMO and thus is part of the management layer. ML training host is part of the Non-RT RIC, the ML inference host can be part of Non-RT RIC or Near-RT RIC.

3.1.2 Unsupervised learning

Input data is not labelled and does not have a known result. Unsupervised learning is a ML task that aims to learn a function to describe a hidden structure from unlabelled data. Some examples of unsupervised learning are K-means clustering and principal component analysis (PCA).

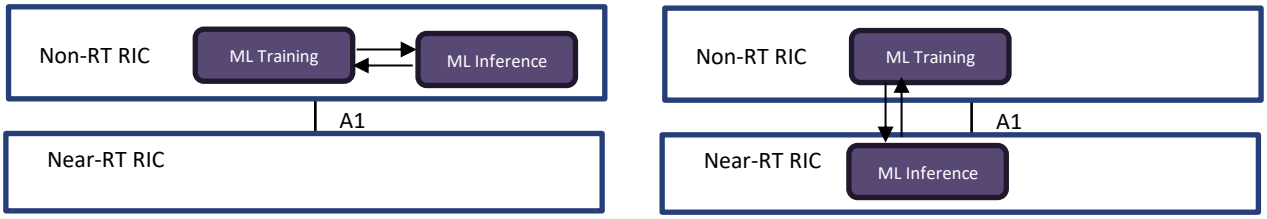

Figure 4 Unsupervised learning model training and actor locations

In unsupervised learning (see Figure 4), ML training host is part of the Non-RT RIC, the ML inference host can be part of Non-RT RIC or Near-RT RIC.

3.1.3 Reinforcement learning

A goal-oriented learning based on interaction with environment. In reinforcement learning (RL), the agent aims to optimize a long-term objective by interacting with the environment based on a trial-and-error process. There are several RL algorithms, e.g.:

- Q-learning
- Multi-armed bandit learning
- Deep RL

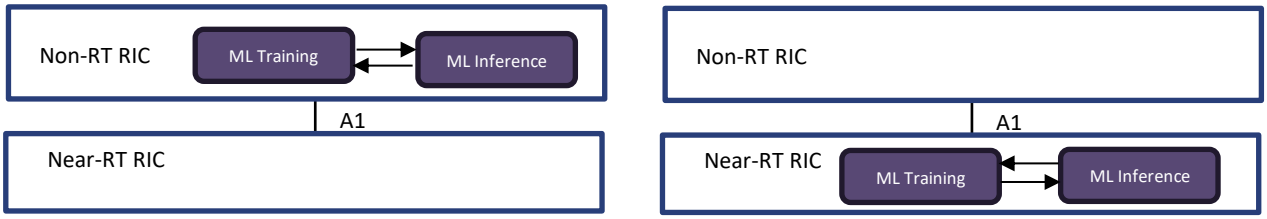

Figure 5 Reinforcement learning model training and actor locations

In reinforcement learning (see
Figure 5), ML training host and ML inference host shall be co-located as part of Non-RT RIC or Near-RT RIC.

3.2 MAPPING AI/ML FUNCTIONALITIES INTO O-RAN CONTROL LOOPS

There are three types of control loops defined in O-RAN. ML assisted solutions fall into the three control loops. Time scale of O-RAN control loops depend on what is being controlled, e.g., system parameters, RRM algorithm parameters. For example, if O-RAN control loop adapts the parameters of RRM algorithms, its time scale is slower than that of the RRM algorithm.

Loop 1 deals with per transmission time interval (TTI) msec level scheduling and operates at a time scale of the TTI or above. Loop 2 operates in the near RT RIC operating within the range of 10-500 msec and above (resource optimization). Loop 3 operates in the non-RT RIC at greater than 500 msec (policies, orchestration). It is not expected that these loops are hierarchical but can instead run in parallel.

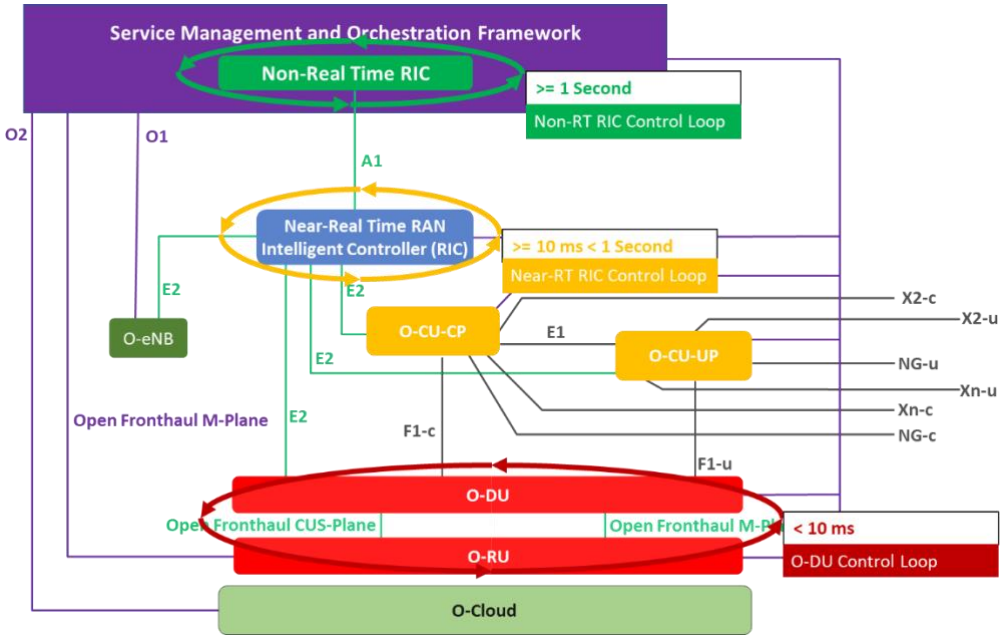

Figure 6 Control loops in O-RAN

Figure 6 shows the three control loops in O-RAN architecture. AI/ML related functionalities can be mapped into the three loops. The location of the ML training host and the ML inference host for a use case depends on the computation complexity, on the availability and the quantity of data to be exchanged, on the response time requirements and on the type of ML model. For example, online ML model for configuring RRM algorithms operating at the TTI time scale could run in O-DU, while the configuration of system parameters such as beamforming configurations requiring a large amount of data with no response time constraints can be performed in the Non-RT RIC and SMO layer where intensive computation means can be made available.

3.3 AI FLOWS IN O-RAN ARCHITECTURE

In the first phase of O-RAN, ML model training will be considered in the non-RT RIC and ML model inference will be considered in loops 2 and 3. For loop2, the ML inference is typically running in near-RT RIC. For Loop 1, the ML model inference is typically running in an O-DU. ML workflow on loop 1 is for further specification. While ML model implementation in O-RU could be envisaged, it is presently not supported in O-RAN.

The section provides an example of ML model lifecycle implementation within the O-RAN architecture. Figure 7 below provides a high-level overview of the typical steps of AI/ML-based use case applications within O-RAN architecture considering supervised learning/unsupervised learning ML models.

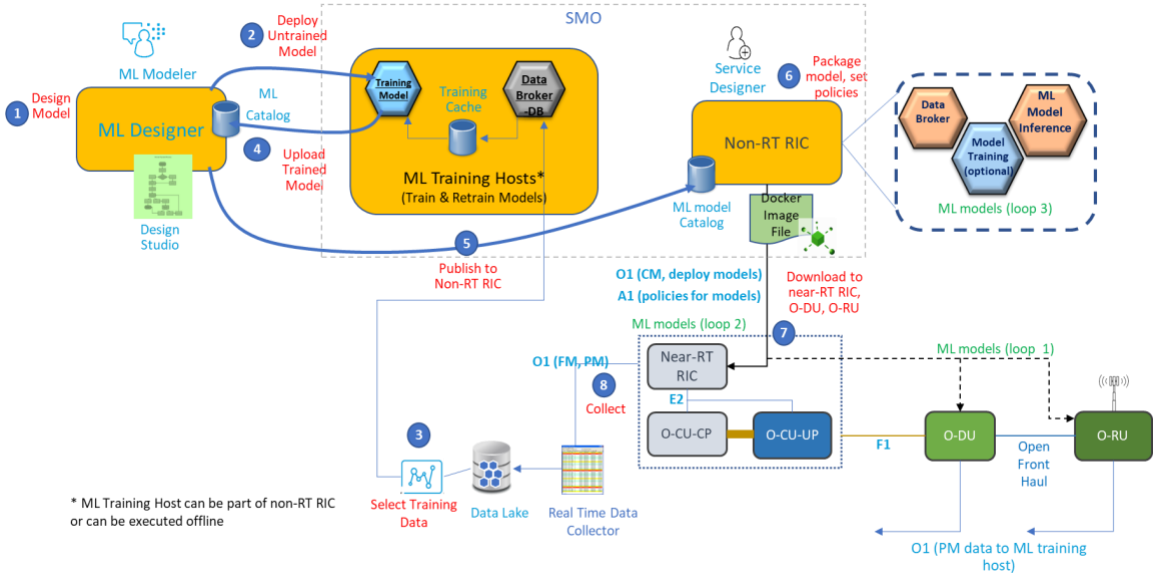

Figure 7: ML Model Lifecycle Example

The steps for reinforcement model could vary with respect to ML training host and the related interaction flows. Nonetheless, the typical steps include:

1. ML modeler uses a designer environment that creates the initial ML model.
2. The initial model is sent to training hosts for training.
3. The appropriate data sets are collected from the near-RT RIC, O-CU and O-DU to a data lake and passed to the ML training hosts.
4. The trained model/sub models are uploaded to the ML designer catalog. The final ML model is composed.
5. The ML model is published to non-RT RIC along with the associated license and metadata.
6. Non-RT RIC deploys the ML application to the near-RT RIC, O-DU and O-RU using the O1 interface. Policies are also set using the A1 interface.
7. Performance measurement data is sent back to ML training hosts from near-RT RIC, O-DU and O-RU for retraining.

4 AI-BASED RADIO RESOURCE MANAGEMENT

This section seeks to apply the framework described in Sections 2 and 3 to the reference architecture defined in [1, 2]. The proposed solution integrates a terrestrial 5G system and a 5G NTN. The NTN component is based on a low Earth orbit (LEO) constellation that can be complemented by geostationary Earth orbit (GEO) satellite. Depending on the OBP capabilities and the functional split, the space segment may host different radio and network functions. Remarkably, user equipment (UE) can connect directly to base stations through TN and NTN links or indirectly using integrated access backhauling (IAB).

This section identifies the RRM problems that can be handled by both the non-RT RIC and the near-RT RIC. Regarding the design of the algorithms, an innovative solution is proposed to solve the problems that involve a forecasting task. The specific case of traffic prediction has been investigated with real satellite data for large timescales. For smaller timescales, guidelines are provided to tackle the the user scheduling RRM algorithm in the context of user-centric beamforming.

4.1 NON-REAL TIME RIC

4.1.1 Non-real time RIC description

The non-RT RIC is a fundamental component of the Open RAN architecture, designed for managing RAN operations with control loops exceeding one second. Unlike the near-RT RIC, it handles operations on longer timescales and supports third-party applications, known as rApps, which aid in optimizing and managing RAN functions. These applications enhance the RAN by providing services such as policy guidance, configuration management, and data analytics. The non-RT RIC is part of the SMO framework within the O-RAN architecture. It is connected to other network elements in the architecture by means of A1, O1 and O2 interfaces. On the one hand, A1 interface connects the non-RT RIC with near-RT RICs to exchange policies and control data. On the other hand, O1 and O2 interfaces are used for operations, administration and management functions. Non-RT RIC enables data management and exposure services. It performs the data management and exposure to services and can cover all the aspects of the AI/ML development, including the collection of data, the training of the AI models and the subsequent validation of them, and, finally, their deployment and execution. The primary responsibilities of non-RT RIC include:

POLICY MANAGEMENT

- Definition and distribution: The non-RT RIC defines policies and guidelines for RAN behaviour. These policies are then distributed to the near-RT RIC and other RAN elements.
- Optimization: It continually refines policies based on long-term data analysis to optimize network performance and resource utilization.

MACHINE LEARNING AND ANALYTICS

- Data collection and analysis: It collects vast amounts of data from the RAN elements and uses this data for advanced analytics.

- Model training: The non-RT RIC trains ML models that can predict and adapt to network conditions. These models can be then deployed to the near-RT RIC to assist in real-time decision-making.

RAN OPTIMIZATION

- Performance optimization: By analyzing historical data and trends, the non-RT RIC can identify performance bottlenecks and suggest optimizations.
- Load Balancing and resource management: It helps in strategic planning for load balancing and resource allocation based on long-term usage patterns.

4.1.2 Radio resource management problems to be handled

TRAFFIC OFFLOADING

It is known that RAN consumes most of the energy in cellular networks and that the traffic load in some regions is usually low at night hours, e.g., rural areas or technopoles. Under these premises, NTN nodes could be used to offload TN traffic during nighttime. Consequently, some rural TN cells or carriers could be switched off, which is an effective mechanism to save energy when the load is low. Then, on the basis that that users can access the network either through TN or NTN links, AI-based traffic offloading techniques could be used to decide when NTN comes into play to provide coverage. For instance, when the traffic forecast is too low. This is aligned with the use case 1.2 described in [6]. The scenario is represented in Figure 8.

Interestingly, the complementary scenario (i.e., traffic offload from terrestrial networks of non-time sensitive data at peak hours) is another application, which is already considered as a use case by 3GPP for NTN. In such a case, the traffic should be offloaded during peak hours when the traffic forecast is higher than a given threshold.

It is important to remark that traffic forecasting is the main enabler to realize AI-based traffic offloading techniques. Based on the timescale of the predictions, configuration changes of TN cells could be triggered by non-RT RIC rApps. Switch on/off decisions will be provided over the O1 and E2 interfaces.

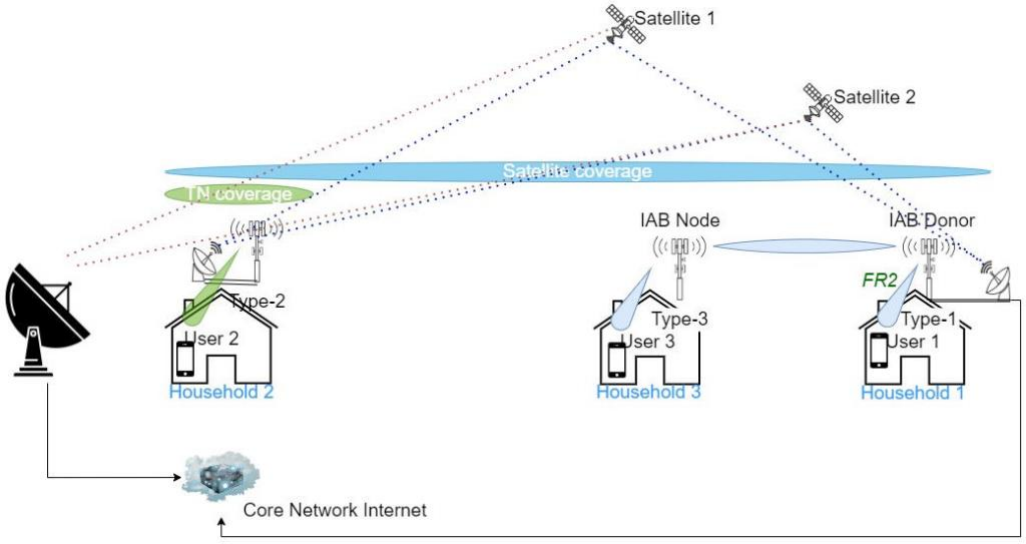

Figure 8: Residential broadband scenario

BANDWIDTH ALLOCATION

Data-driven bandwidth allocation for the aggregated traffic is a RRM task that can be handled by the non-RT RIC. The objective is not to decide the number of physical resource blocks (PRBs) that shall be allocated to UEs. By contrast, the aim is to decide the number of PRBs that are need among multiple NTN beams to serve all the users. IN such a case, non-RT RIC rApps can pre-allocate PRBs on a large timescale based on the traffic demand. Analogously to the traffic offloading problem, traffic forecasting is the core task. Optimizing the bandwidth allocation is essential to make an efficient use of the satellite resources.

4.1.3 AI-techniques for non-real time RRM

This section describes an AI/ML prediction engine that is based on the application of KANs to time-series forecasting.

4.1.3.1 Kolmogorov-Arnold Networks (KANs) for Time Series Analysis

Inspired by the Kolmogorov-Arnold representation theorem, KANs replace traditional linear weights with spline-parametrized univariate functions, allowing them to learn activation patterns dynamically. In this section it is demonstrated that KANs outperforms conventional multi-layer perceptrons (MLPs) in a real-world satellite traffic forecasting task, providing more accurate results with considerably fewer number of learnable parameters. We also provide an ablation study of KAN-specific parameters impact on performance. The proposed approach opens new avenues for adaptive forecasting models, emphasizing the potential of KANs as a powerful tool in predictive analytics, which is relevant for non-RT RRM.

Time series forecasting plays a key role in a wide range of fields, driving critical decision-making processes in finance, economics, medicine, meteorology, and biology, among others, reflecting the wide applicability and its significance across many domains. It involves predicting future values based on the previously observed data points. With this goal in mind, understanding the dynamics of time-dependent phenomena is essential and requires unveiling the patterns, trends and dependencies hidden with the historical data. While conventional approaches have been traditionally centred on parametric models grounded in domain-specific knowledge, such as autoregressive (AR), exponential smoothing, or structural time series models, contemporary ML techniques offered a pathway to discern temporal patterns solely from data-driven insights.

Non-ML methods traditionally tackle the time series forecasting problem and often rely on statistical methods to predict future values based on previously observed data. One of the most well-known techniques is the Autoregressive integrated moving average (ARIMA) model, which combines auto-regression, integration, and moving averages to forecast data. The authors in [7] detailed this approach, providing a comprehensive methodology foundational for subsequent statistical forecasting methods. Extensions of ARIMA, like seasonal ARIMA (SARIMA), adapt the model to handle seasonality in data series, particularly useful in fields like retail and climatology [8]. Exponential Smoothing techniques constitute another popular set of traditional (non-ML-based) forecasting methods. They are characterized by their simplicity and effectiveness in handling data with trends and seasonality. An exponent of this family of techniques is the so-called Holt-Winters seasonal technique, which adjusts the model parameters in response to changes in trend and seasonality within the time series data [9, 10]. These models have been widely used for their efficiency, interpretability and implementation.

More recently, ML models have significantly impacted the forecasting landscape by handling large datasets and capturing complex nonlinear relationships that traditional methods cannot. In recent years, deep learning (DL)-based forecasting models have gained popularity, motivated by the notable achievements in many fields. For instance, neural networks have been extensively studied due to their flexibility and adaptability. Simple MLPs were among the

first to be applied to forecasting problems, demonstrating significant potential in non-linear data modelling [11].

Built upon these light models, more complex architectures have progressively expanded the capabilities of neural networks in time series forecasting. Typical examples are recurrent neural network architectures such as long short-term memory (LSTM) networks and gated recurrent units (GRUs), which are designed to maintain information in memory for long periods without the risk of vanishing gradients – a common issue in traditional recurrent networks [12]. On a related note, convolutional neural networks (CNNs), which are fundamentally inspired by MLPs, are also extensively employed in time series forecasting. These architectures are particularly efficient at processing temporal sequences due to their strong spatial pattern recognition capabilities. The combination of CNNs with LSTMs has resulted in models that efficiently process both spatial and temporal dependencies, enhancing forecasting accuracy [13]. These models have started to outperform established benchmarks in complex forecasting tasks, motivating a significant shift towards more complex network structures. Unfortunately, since the majority of the models mentioned above are inspired by MLP architecture, they tend to have poor scaling law, i.e., the number of parameters in MLPs networks do not scale linear with the number of layers, and often lack interpretability.

A recent study in reference [14], which was recently proposed, introduces KANs, a novel neural network architecture designed to potentially replace traditional multilayer perceptrons. KANs represent a disruptive paradigm shift, and as a potential game changer have recently attracted the interest of the AI community worldwide. They are inspired by the Kolmogorov-Arnold representation theorem, [15, 16]. Unlike MLPs, which are inspired by the universal approximation theorem, KANs take advantage of this representation theorem to generate a different architecture. They innovate by replacing linear weights with spline-based univariate functions along the edges of the network, which are structured as learnable activation functions. This design not only enhances the accuracy and interpretability of the networks, but also enables them to achieve comparable or superior results with smaller network sizes across various tasks, such as data fitting and solving partial differential equations. While KANs show promise in improving the efficiency and interpretability of neural network architectures, the study acknowledges the necessity for further research into their robustness when applied to diverse datasets and their compatibility with other deep learning architectures. These areas are crucial for understanding the full potential and limitations of KANs.

The content described in this section is based on prospective study presented in [17], which investigates the application of KANs to time series forecasting. To the best of authors' knowledge, not previously explored in the literature. The aim of [17] is to evaluate the practicality of KANs in real-world scenarios, analysing their efficiency in terms of the number of trainable parameters and discussing how the additional degrees of freedom might affect forecasting performance. Herein, it is assessed the performance using real-world satellite traffic data. This exploration seeks to further validate KANs as a versatile tool in advanced neural network design for time series forecasting, although more comprehensive studies are required to optimize their use across broader applications. Finally, we note that due to the early stage of KANs, it is fair to compare it as a potential alternative to MLPs, but further investigation is needed to develop more complex solutions that can compete with advanced architectures such as LSTMs, GRUs and CNNs.

First, this section presents the problem statement, providing fundamental background on Kolmogorov-Arnold representation theorem and KANs. Next, the experimental setup description is presented, which serves as basis to analyse the performance of KANs with real-world datasets.

PROBLEM STATEMENT

The traffic forecasting problem is formulated as a time series at time t represented by y_t. The objective is to predict the future values of the series.

$$\boldsymbol{y}_{t_0:T} = [y_{t_0}, y_{t_0+1}, \ldots, y_{t_0+T}] \quad (1)$$

based solely on its historical values.

$$\boldsymbol{x}_{t_0-c:t_0-1} = [x_{t_0-c}, \ldots, x_{t_0-2}, x_{t_0-1}] \quad (2)$$

where t_0 denotes the starting point from which future values $y_t, t = t_0, \ldots, T$ are to be predicted. We differentiate the historical time range $[t_0 - c, t_0 - 1]$ and the forecast range $[t_0, T]$ as the context and prediction lengths, respectively. This approach focuses on generating point forecasts for each time step in the prediction length, aiming to achieve accurate and reliable forecasts. Figure 9 shows an exemplary time series.

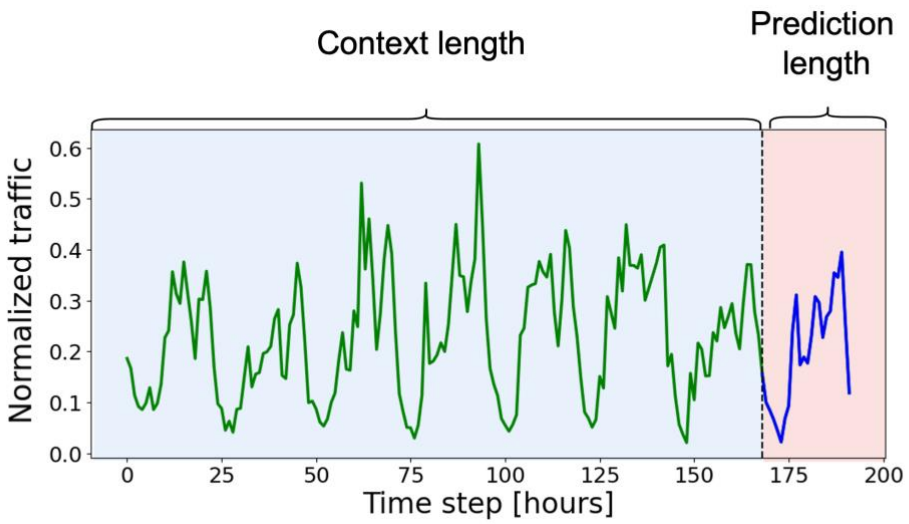

Figure 9: Example of normalized satellite traffic series data with the conditioning and prediction lengths denoted in blue, and red, respectively.

Kolmogorov-Arnold representation background

Contrary to MLPs, which are based on universal approximation theorem, KANs rely on the Kolmogorov-Arnold representation theorem, also known as the Kolmogorov-Arnold superposition theorem. A fundamental result in the theory of dynamical systems and ergodic theory. It was independently formulated by Andrey Kolmogorov and Vladimir Arnold in the mid-20th century.

The theorem states that any multivariate continuous function f, which depends on $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, on a bounded domain, can be represented as the finite composition of simpler continuous functions, involving only one variable. Formally, a real, smooth, and continuous multivariate function $f(\mathbf{x}): [0,1]^n \to \mathbb{R}$ can be represented by the finite superposition of univariate functions:

$$f(\boldsymbol{x}) = \sum_{i=1}^{2n+1} \Phi_i \left(\sum_{j=1}^{n} \phi_{i,j}(x_j) \right), \quad (3)$$

where $\Phi_i: \mathbb{R} \to \mathbb{R}$ and $\phi_{i,j}: [0,1] \to \mathbb{R}$ denote the so-called outer and inner functions, respectively. One might initially perceive this development as highly advantageous for ML. The task of learning a high-dimensional function simplifies to learning a polynomial number of one-dimensional functions. Nevertheless, these 1-dimensional functions can exhibit non-smooth characteristics, rendering them potentially unlearnable in practical contexts. As a result of this problematic behavior, the Kolmogorov-Arnold representation theorem has been traditionally disregarded in machine learning circles, recognized as theoretically solid, but ineffective in practice. Unexpectedly, the theoretical result in [14] has recently emerged as a potential game changer, paving the way for new network architectures, inspired by the Kolmogorov-Arnold theorem.

Kolmogorov-Arnold network background

The authors in [14] mention that equation $f(\boldsymbol{x}) = \sum_{i=1}^{2n+1} \Phi_i \left(\sum_{j=1}^{n} \phi_{i,j}(x_j) \right)$, (3) has two layers of non-linearities, with $2n+1$ terms in the middle layer. Thus, we only need to find the proper functions inner univariate functions $\phi_{i,j}$ and Φ_i that approximate the function. The one-dimensional inner functions $\phi_{i,j}$ can be approximated using B-splines. A spline is a smooth curve defined by a set of control points or knots. Splines are often used to interpolate or approximate data points in a smooth and continuous manner. A spline is defined by the order k ($k = 3$ is a common value), which refers to the degree of the polynomial functions used to interpolate or approximate the curve between control points. The number of intervals, denoted by G, refers to the number of segments or subintervals between adjacent control points. In spline interpolation, the data points are connected by these segments to form a smooth curve (of $G + 1$ grid points). Although splines other than B-splines could also be considered, this is the approach proposed in [14]. Equation $f(\boldsymbol{x}) = \sum_{i=1}^{2n+1} \Phi_i \left(\sum_{j=1}^{n} \phi_{i,j}(x_j) \right)$, (3) can be represented as a 2-layer (or analogous 2-depth) network, with activation functions placed at the edges (instead of at the nodes) and nodes performing a simple summation. Such two-layer network is too simplistic to effectively approximate any arbitrary function with smooth splines. For this reason, reference [14] extends the ideas discussed above by proposing a generalized architecture with wider and deeper KANs.

A KAN layer is defined by a matrix $\boldsymbol{\Phi}$ composed by univariate functions $\{\phi_{i,j}(\cdot)\}$ with $i = 1, \ldots, N_{in}$ and $j = 1, \ldots, N_{out}$, where N_{in} and N_{out} denote the number of inputs and the number of outputs, respectively, and $\phi_{i,j}$ are the trainable spline functions described above. Note according to the previous definition, the Kolmogorov-Arnold representation theorem can be expressed as a two-layer KAN. The inner functions constitute a KAN layer with $N_{in} = n$ and $N_{out} = 2n + 1$, while the external functions constitute another KAN layer with $N_{in} = 2n + 1$ and $N_{out} = 1$.

Let us define the shape of a KAN by $[n_1, \ldots, n_{L+1}]$, where L denotes the number of layers of the KAN. It is worth noting the Kolmogorov-Arnold theorem is defined by a KAN of shape $[n, 2n + 1, 1]$. A generic deeper KAN can be expressed by the composition L layers:

$$\boldsymbol{y} = KAN(\boldsymbol{x}) = (\boldsymbol{\Phi}_L \circ \boldsymbol{\Phi}_{L-1} \circ \ldots \circ \boldsymbol{\Phi}_1)\boldsymbol{x}. \quad (4)$$

Notice that all the operations are differentiable. Consequently, KANs can be trained with backpropagation. Despite their elegant mathematical foundation, KANs are simply combinations of splines and MLPs, which effectively exploit each other's strengths while mitigating their respective weaknesses. Splines stand out for their accuracy on low-dimensional functions and allow transition between various resolutions. Nevertheless, they suffer from a major dimensionality problem due to their inability to effectively exploit compositional structures. In contrast, MLPs experience a lower dimensionality problem, due to their ability to learn features, but exhibit lower accuracy than splines in low dimensions due to their inability to optimize univariate functions effectively. KANs have by their construction 2 levels of degrees of freedom. Consequently, KANs possess the capability not only to acquire features, owing to their external resemblance to MLPs, but also to optimize these acquired features with a high degree of accuracy, facilitated by their internal resemblance to splines. To learn features accurately, KANs can capture compositional structure (external degrees of freedom), but also effectively approximate univariate functions (internal degrees of freedom with the splines). It should be noted that by increasing the number of layers L or the dimension of the grid G, we are increasing the number of parameters and, consequently, the complexity of the network. This approach constitutes an alternative to traditional DL models, which are currently relying on MLP architectures and motivates the extension of this work.

KAN time series forecasting network

The traffic forecasting problem is framed as a supervised learning framework consisting of a training dataset with input-output $\{\mathbf{x}_{t_0-c:t_0-1}, \mathbf{y}_{t_0:T}\}$ in the condition and prediction lengths. The aim is to find f that approximates $\mathbf{y}_{t_0:T}$, i.e., $\mathbf{y}_{t_0:T} \approx f(\mathbf{x}_{t_0-c:t_0-1})$. For ease of notation, the framework is described as a two-layer (2-depth) KAN $[N_i, n, N_o]$ (note that to comply with the original paper notation, the input layer is not accounted as a layer per se). The output and input layers will be comprised of N_o, and N_i nodes corresponding to the total amount of time steps in $\boldsymbol{y}_{t_0:T} = [y_{t_0}, y_{t_0+1}, \ldots, y_{t_0+T}]$ (1)and$\boldsymbol{x}_{t_0-c:t_0-1} = [x_{t_0-c}, \ldots, x_{t_0-2}, x_{t_0-1}]$ (2), while the transformation/hidden layer of n nodes. The inner functions constitute a KAN layer with $N_{in} = N_i$ and $N_{out} = n$, while the external functions constitute another KAN layer with $N_{in} = n$ and $N_{out} = N_o$. Our KAN can be expressed by the composition of 2 layers:

$$\boldsymbol{y} = KAN(\boldsymbol{x}) = (\boldsymbol{\Phi}_2 \circ \boldsymbol{\Phi}_1)\boldsymbol{x}. \quad (5)$$

where the output functions Φ_2 generates the N_o outputs values corresponding to $\boldsymbol{y}_{t_0:T} = [y_{t_0}, y_{t_0+1}, \ldots, y_{t_0+T}]$ (1) by doing the transformation from the previous layers. The proposed network can be used to forecast future traffic data in the prediction length, based solely on the context length. Figure 10 shows a generic representation for any arbitrary number of layers L.

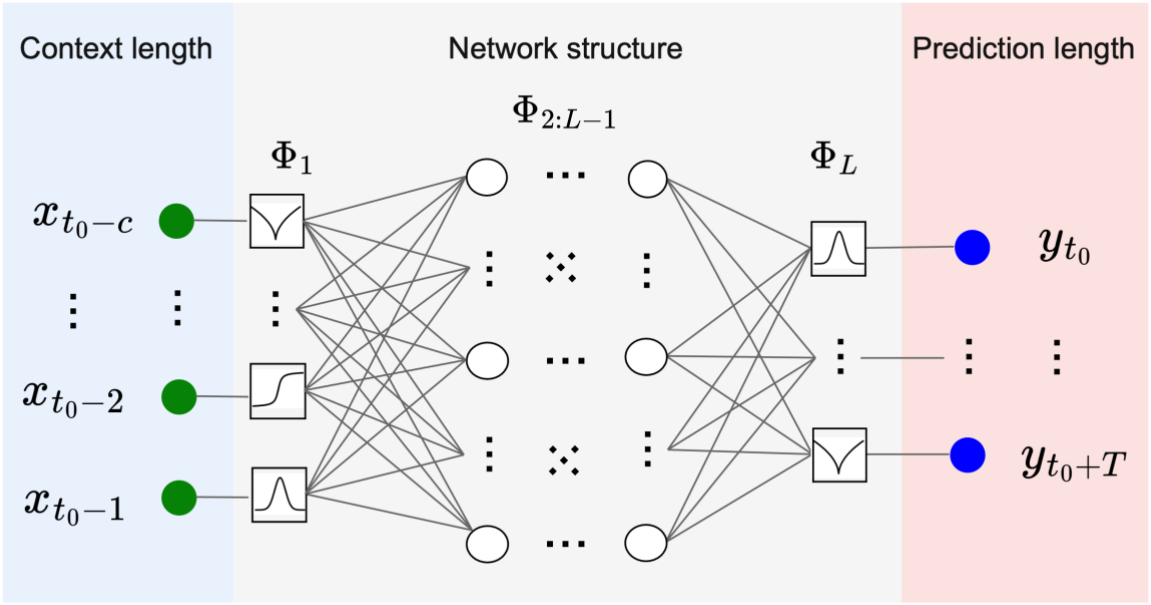

Figure 10: Example of the flow of information in the KAN network architecture for our traffic forecasting task. Learnable activations are represented inside a square box.

EXPERIMENTAL SETUP

The data set has been generated according to [3, 4]. Hence, the inputs are obtained from a satellite operator (SO), as a result of processing real information from a GEO satellite communication system, which provisions broadband services. The dataset is a long time series capturing aggregated traffic data. To preserve privacy, anonymous clients have been defined with more than 500 connected users, and the traffic has been normalized. The measurements are monthly long, and the time granularity is 1 hour.

The traffic has been extracted per satellite beam in Megabits per second (Mbps). Although the data has been collected using a GEO satellite communication system, it is expected that user needs could be used to address LEO systems, as well. It is worth emphasizing that the data collected can be used for AI-driven predictive analysis, to forecast traffic conditions, which is essential to avoid congestion and to make efficient use of satellite resources. Endowing the network with intelligence will be beneficial to meet the different demands of satellite applications.

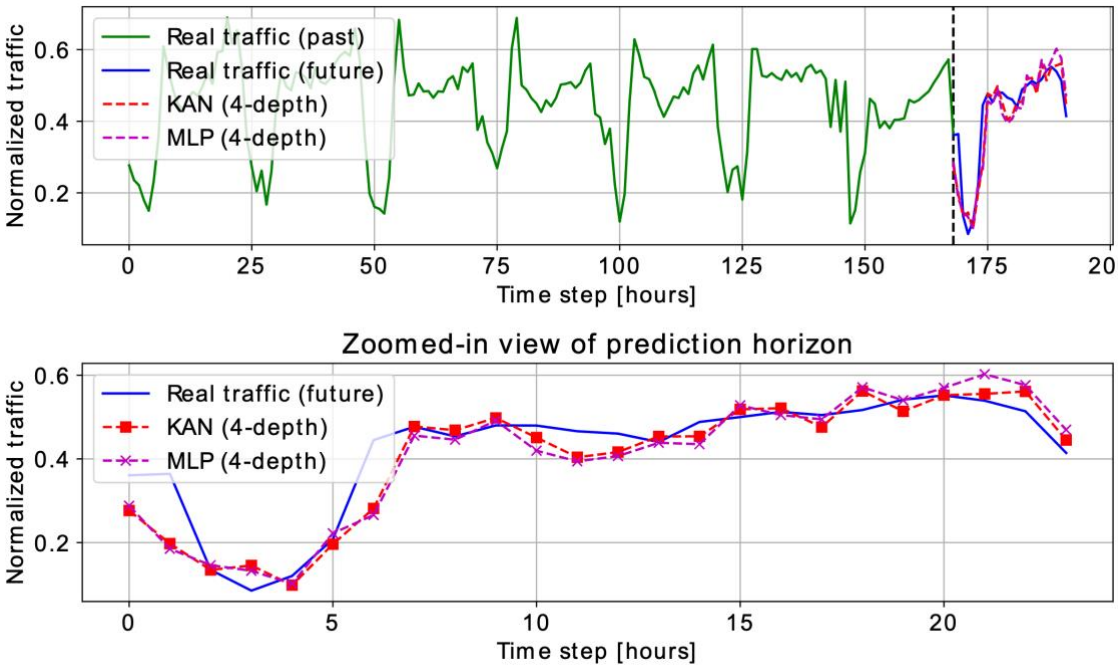

Figure 11: Satellite traffic over three different beams wit forecasted values over beam 1 using a 4-depth KAN and a 4-depth MLP.

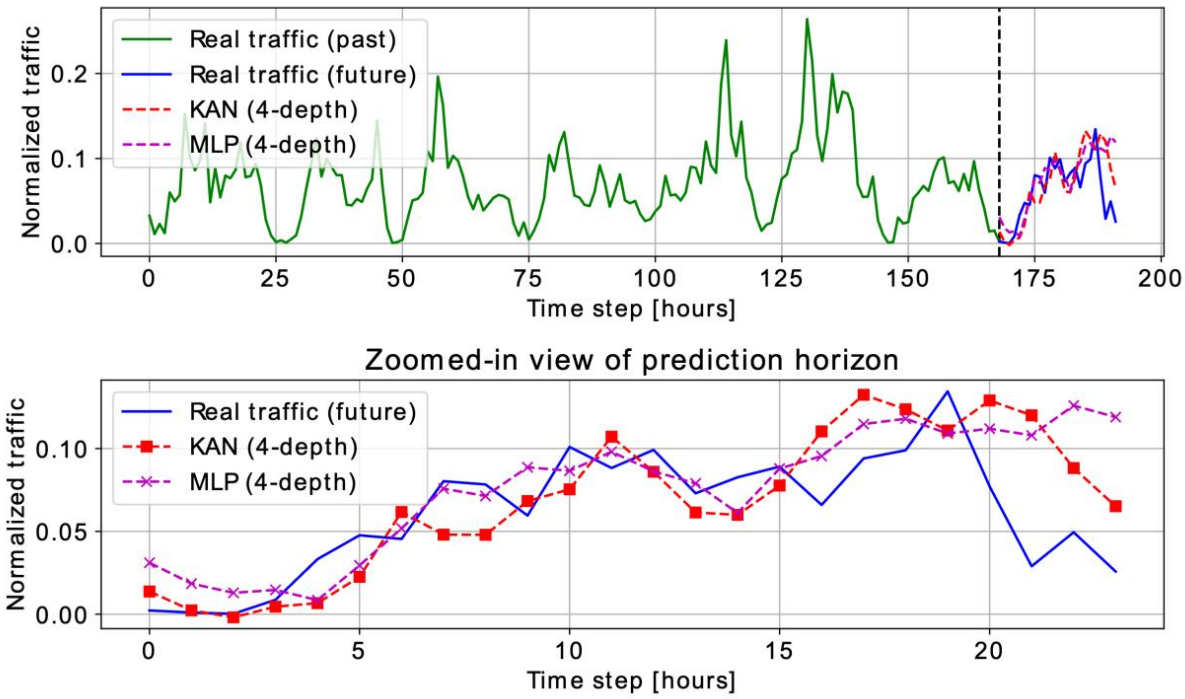

Figure 12: Satellite traffic over three different beams wit forecasted values over beam 2 using a 4-depth KAN and a 4-depth MLP.

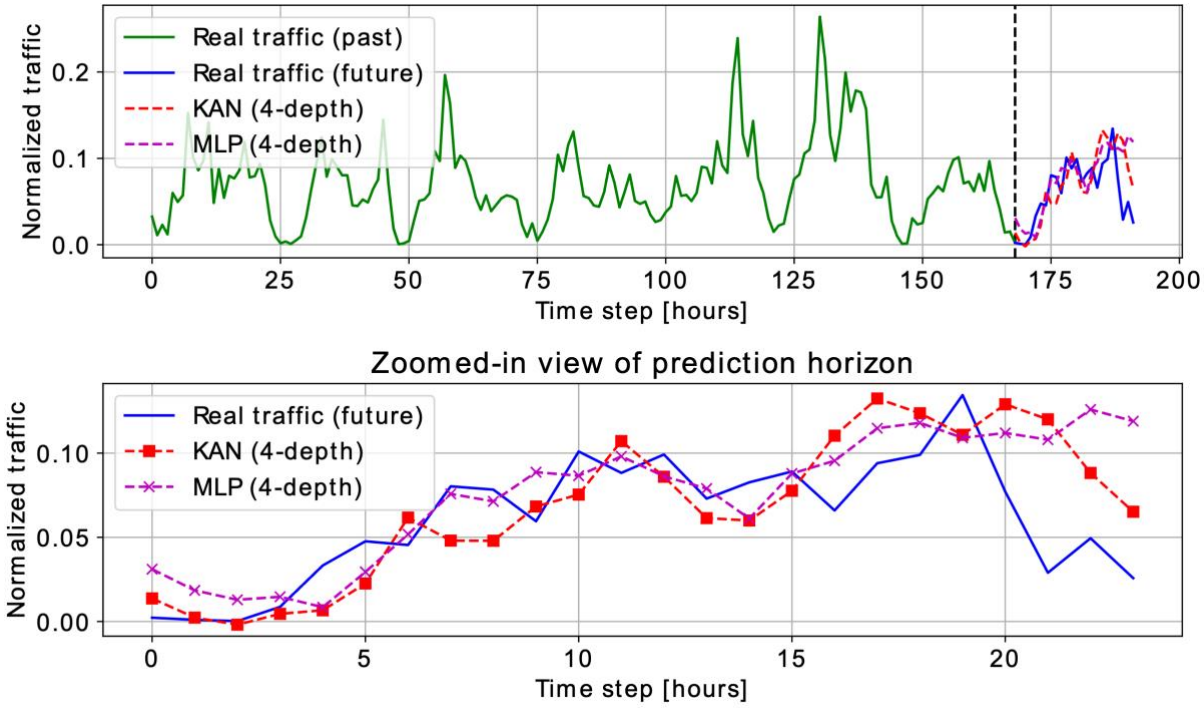

Figure 13: Satellite traffic over three different beams wit forecasted values over beam 3 using a 4-depth KAN and a 4-depth MLP.

SIMULATION RESULTS

This section investigates the forecasting performance of different KAN and MLP architectures for predicting satellite traffic over the six beam areas. Concretely, we have a context length of 168 hours (one week) and a prediction length of 24 hours (one day). This translates to $T = 24$, $c = 168$, where $y_{t_0+T} = 192$ in $\boldsymbol{y}_{t_0:T} = [y_{t_0}, y_{t_0+1}, \ldots, y_{t_0+T}]$ (1) and $\boldsymbol{x}_{t_0-c:t_0-1} = [x_{t_0-c}, \ldots, x_{t_0-2}, x_{t_0-1}]$ (2). The focus is on evaluating the efficacy of KAN models compared to traditional MLPs. We designed our experiments to compare models with similar depths but varying architectures to analyze their impact on forecasting accuracy and parameter efficiency. Table 1 summarizes the parameters selected for this evaluation. We have data for the six beams over one month. We use two weeks + 1 day for training and one week + 1 day for testing for all beams that were not seen by the network. We train all the networks with 500 epochs and Adam optimizer with a learning rate of 0.001. The selected loss function minimizes the mean absolute error (MAE) of the values around the prediction length.

Table 1: Model configurations for satellite traffic forecasting

Model	Configuration	Time horizon (h)	Spline details	Activations
MLP (3-depth)	[168, 300, 300, 300, 24]	Context/Prediction: 168/24	N/A	ReLU (fixed)
MLP (4-depth)	[168, 300, 300, 300, 300, 24]	Context/Prediction: 168/24	N/A	ReLU (fixed)

KAN (3-depth)	[168, 40, 40, 24]	Context/Prediction: 168/24	Type: B-spline, k = 3, G = 5	Learnable
KAN (4-depth)	[168, 40, 40, 40, 24]	Context/Prediction: 168/24	Type: B-spline, k = 3, G = 5	Learnable

Performance analysis for real traffic

We analyze the forecasting performance in the prediction length. Figure 11,

Figure 12 and Figure 13 depict the real traffic value used as input (in green) to the networks, the expected output prediction length (in blue) and the values predicted values using a KAN (in red) and MLP (in purple) of depth 4 both – see Table 1 for details on model configuration. In general, the results show that the predictions obtained using KANs better approximates the real traffic values than the predictions obtained using traditional MLPs. This is particularly evident in Figure 11. Here, KAN accurately matches rapid changes in traffic volume, which the MLP models sometimes moderately over/under-predicted, as the last part of the forecast shows. This capability suggests that KANs are better suited to adapt to sudden shifts in traffic conditions, a critical aspect of effective traffic management.

Additionally, the responsiveness of KANs is particularly noticeable in

Figure 12 during fast changing traffic conditions. KAN shows a rapid adjustment to its forecast that is closely aligned with the actual traffic pattern. This is particularly noticeable in the last 6 hours of the prediction length where MLP exhibits a lag failing to capture these immediate fluctuations, which shows its worse performance to capture dynamic traffic variations. Further analysis is shown in Figure 13, where traffic conditions are more variable and intense, demonstrated the robustness of KAN in maintaining high performance despite the complexity and higher volume. This robustness suggests that KANs can manage different scales and intensities of traffic data more effectively than MLPs, making them more reliable for deployment in varied traffic scenarios.

To further quantify the performance and advantages of using KANs for the satellite traffic forecasting task we show Table 2. It shows a detailed comparison of MLPs and KANs different architectures used for evaluation over all the beams. The table displays the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and the number of trainable parameters for each model. Analysing the error metrics, it becomes clear that KANs outperform MLPs, where the KAN (4-depth) is the best in performance. Its lower values in MSE and RMSE indicates its better ability to predict traffic volumes with lower deviation. Similarly, its lower values in MAE and MAPE suggests that KANs not only provides more accurate predictions but also maintains consistency across different traffic volumes, which is crucial for practical traffic forecasting scenarios.

Furthermore, the parameter count reveals a significant difference in model complexity. KAN models are notably more parameter-efficient, with KAN (4-depth) utilizing only 109k parameters compared to 329k parameters for MLP (4-depth) or 238k for MLP (3-depth). **This reduced complexity suggests that KANs can achieve higher or comparable forecasting accuracy with simpler and potentially faster models**. Such efficiency is especially valuable in scenarios where computational resources are limited or where rapid model deployment is required. The results also show that with an augmentation of 16k parameters in KAN, the

performance can be significantly improved, contrary to MLPs which an increment of 91k parameters does not showcase a significant improvement.

From a technical perspective, KANs leverage a theoretical foundation that provides an intrinsic advantage in modelling complex, non-linear patterns typical in traffic systems. This capability likely contributes to their flexibility and accuracy in traffic forecasting. The consistency in performance across diverse conditions also suggests that KANs have strong generalization capabilities, which is essential for models used in geographically varied locations under different traffic conditions. Moreover, besides obtaining lower error rates, our results also suggest that KANs can do so with considerably smaller number of parameters than traditional MLP networks.

Table 2:Results summary

Model	MSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	MAE ($\times 10^{-2}$)	MAPE	Parameters
MLP (3-depth)	6.34	7.96	5.41	0.64	238k
MLP (4-depth)	6.12	7.82	5.55	1.05	329k
KAN (3-depth)	5.99	7.73	5.51	0.62	93k
KAN (4-depth)	5.08	7.12	5.06	0.52	109k

Prediction of resources to serve the traffic

In 5G NR, the data rate can be computed as follows [18]:

$$\text{data rate (in Mbps)} = 10^{-6} \cdot v_{Layers} \cdot Q_m \cdot f \cdot R \cdot \frac{N_{PRB}^{\mu} \cdot 12}{T_s^{\mu}} (1 - OH)$$

where v_{Layers} is the maximum number of supported layers (for simplicity, here we assume equal to one), Q_m denotes the modulation order, f denotes the so-called scaling factor, μ is the numerology (as defined in [19]), R denotes the code rate, T_s^{μ} is the average OFDM symbol duration in a subframe $T_s^{\mu} = {10^{-3}}/{(14 \cdot 2^{\mu})}$ (Note that normal cyclic prefix is assumed), N_{PRB}^{μ} denotes the number of PRBs, OH is the overhead factor. This last parameter, the OH factor can take different values depending on the frequency range.

For the residential broadband dataset provided in [3, 4], the average signal-to-interference-plus-noise ratio (SINR) at the beam centre and at the edge of the coverage are 11.13 and 9.52 dB, respectively. In order to guarantee a BLER=0.1 at the edge of the coverage, the next values for the aforementioned parameters have been considered: $Q_m = 6$, $f = 1$, R=466/1024, $OH = 0.14$ (FR1). Then, the number of requested PRBs to serve the traffic can be expressed as follows:

$$N_{PRB}^{\mu} = \frac{1024 \cdot \text{data rate (in Mbps)}}{2796 \cdot 12 \cdot 14 \cdot 0.86 \cdot 10^{-3}} = 3.22619 \cdot \text{data rate (in Mbps)}$$

Based on the aforementioned formula, the average requested PRBs are forecasted with KANs and MLPs for the beams 36, 37, 38 and 39, respectively, of the residential broadband dataset are shown in the following figures.

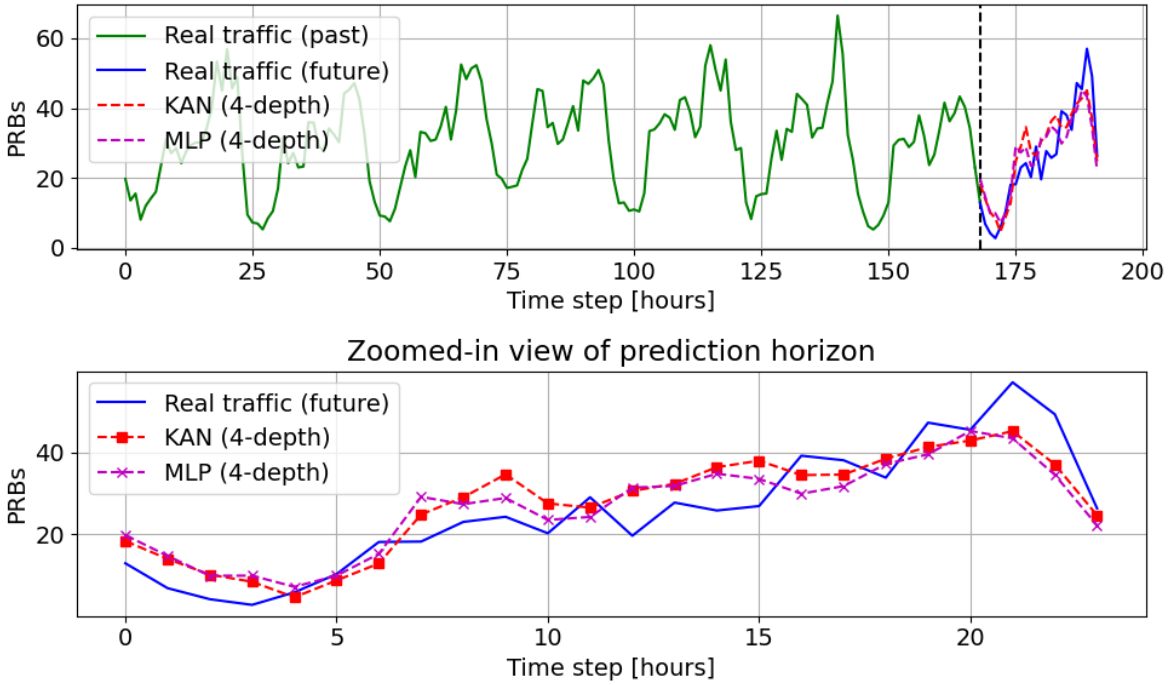

Figure 14: Average PRB forecasts for the beam 36.

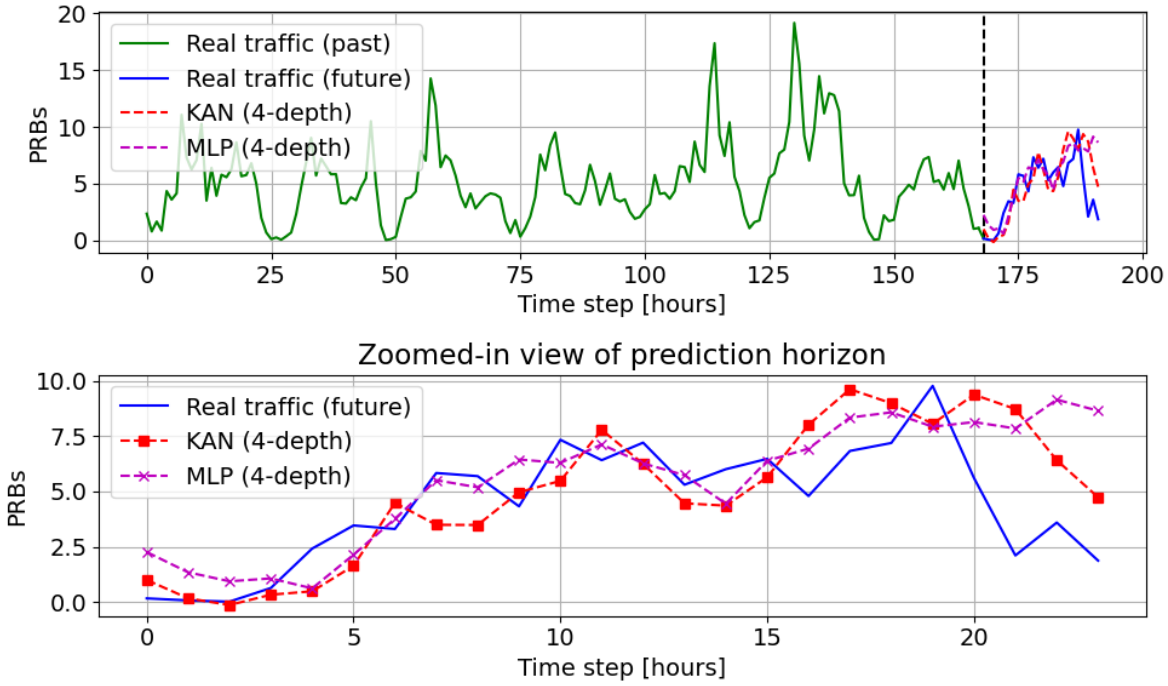

Figure 15: Average PRB forecasts for beam 37.

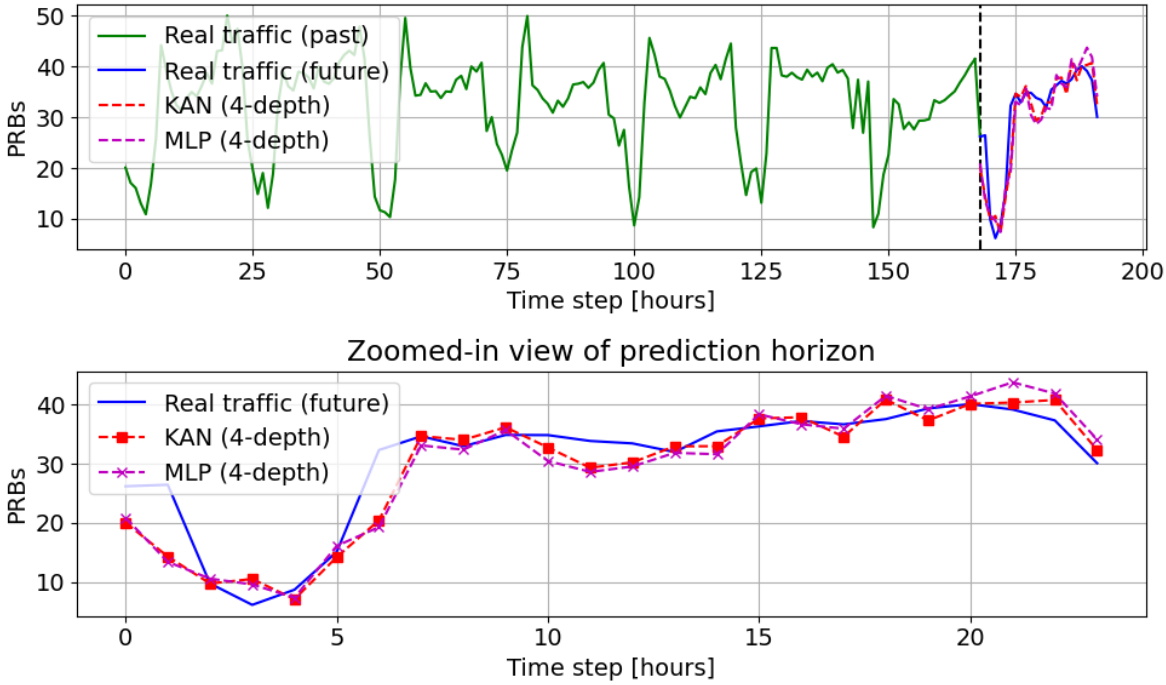

Figure 16: PRB forecasts for beam 38.

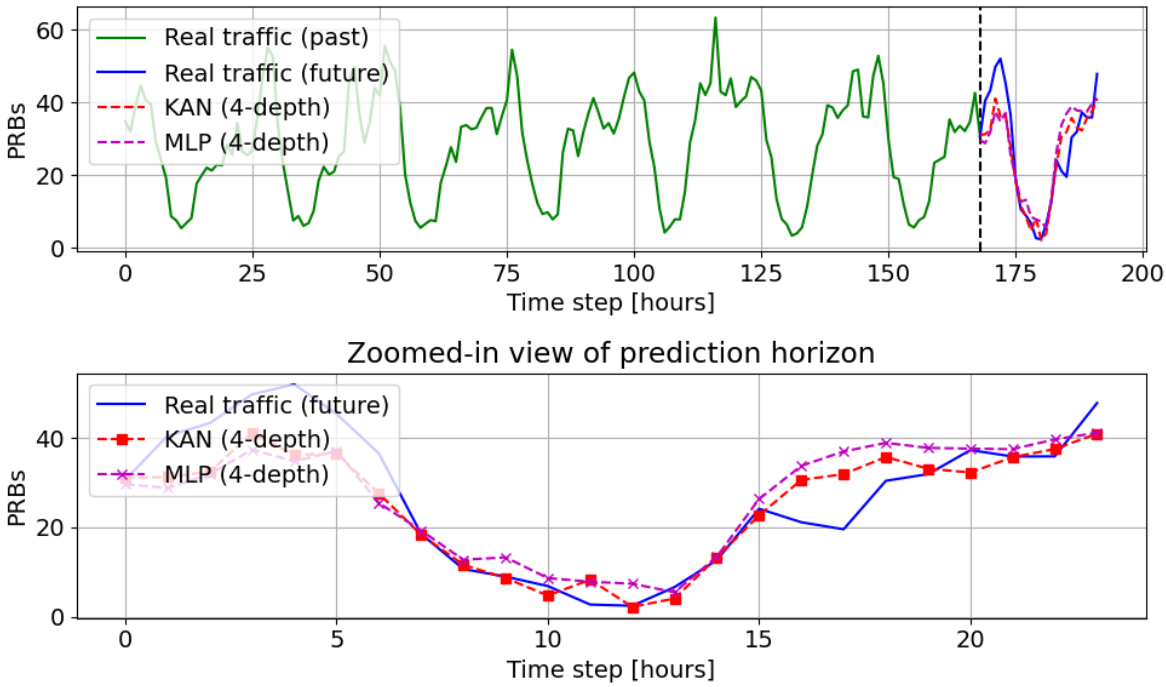

Figure 17: PRB demands for beam 39 (historical and predicted).

Table 3 summarizes the results of PRB forecasting, for the KAN and MLP, for different depths (3-depth and 4-depth), evaluating their performance across several key metrics: MSE, RMSE, MAE, MAPE, and the number of parameters. The analysis reveals that the 4-depth KAN model

significantly outperforms all other models, achieving the lowest MSE (26.79), RMSE (4.81), MAE (3.67), and MAPE (0.52) with only 109k parameters.

Comparatively, the 4-depth MLP model, while improving slightly over its 3-depth counterpart, still lags behind with higher error metrics and substantially more parameters (329k). This indicates that the KAN architecture is not only more accurate but also more parameter-efficient than the MLP, suggesting a better generalization capability and potentially faster inference times. Additionally, increasing the depth from 3 to 4 layers generally enhances the performance for both types of models, but the improvement is notably more pronounced in the KAN models. For instance, the KAN model's MAPE decreases from 0.62 to 0.52 when increasing from 3 to 4 depths, while the MLP model's MAPE counterintuitively increases from 0.64 to 1.05, highlighting potential overfitting issues with deeper MLP structures.
Furthermore, the efficiency of the 4-depth KAN model in terms of parameter usage has significant implications for Open-RAN environments. Open-RAN emphasizes interoperability and flexibility by using open interfaces and modular components, which often operate under constrained computational resources. The reduced parameter count in the KAN model (109k compared to MLP's 329k) translates to lower memory usage and faster processing times. This efficiency can lead to reduced hardware costs and energy consumption, which are critical factors in deploying and scaling Open-RAN solutions. Consequently, the KAN model's superior performance and efficiency support the goals of Open-RAN, promoting more scalable, cost-effective, and sustainable network deployments.

Table 3. PRB prediction resource summary

Model	MSE	RMSE	MAE	MAPE	Parameters
MLP (3-depth)	33.41	5.25	3.92	0.64	238k
MLP (4-depth)	32.26	5.22	4.03	1.05	329k
KAN (3-depth)	31.61	5.19	4	0.62	93k
KAN (4-depth)	26.79	4.81	3.67	0.52	109k

KANs parameter-specific analysis

We provide an insightful analysis of how different configurations of nodes and grid sizes affect the performance of KANs, particularly in the context of traffic forecasting. For this analysis, we designed 3 KANs (2-depth) $[168, n, 24]$ with $n \in \{5,10,20\}$ and varying grids $G \in \{5,10,20\}$ for a $k = 3$ order B-spline. These results are shown during training time.

Figure 18 shows a clear trend, where increasing the number of nodes generally results in lower loss values. This indicates that higher node counts are more effective at capturing the complex patterns in traffic data, thus improving the performance. For instance, configurations with $n = 20$ demonstrate significantly lower losses across all grid sizes compared to those with fewer nodes.

Similarly, the **grid size within the splines of KANs has a notable impact on model performance**. Larger grid sizes, when used with a significant number of nodes ($n \in \{10,20\}$), consistently result in better performance. However, when the number of nodes is low ($n = 5$) the extra complexity of the grid size shows the opposite effect. When having a significant number of nodes larger grids likely provide a more detailed basis for the spline functions,

allowing the model to accommodate better variations in the data, which is crucial for capturing complex temporal traffic patterns.

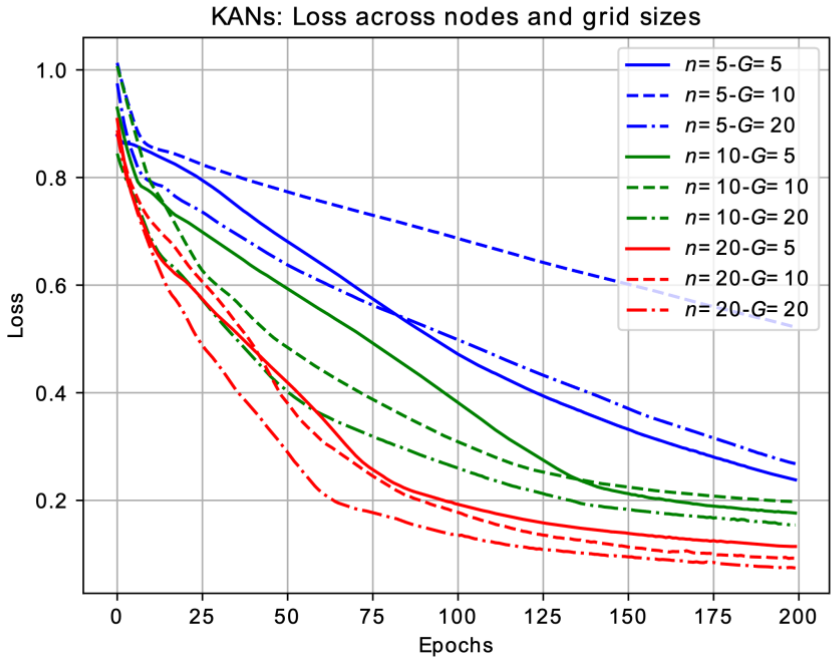

Figure 18: Ablation comparison of KAN-specific parameters.

The best performance is observed in configurations that combine a high node count with a large grid size, such as the $n = 20$, and $G = 20$ setup. This combination likely offers the highest degree of flexibility and learning capacity, making it particularly effective for modelling the intricate dependencies found in traffic data. However, this superior performance comes at the cost of potentially higher computational demands and longer training times, as more trainable parameters are included.

These findings imply that while increasing nodes and grid sizes can significantly enhance the performance of KANs, these benefits must be weighed against the increased computational requirements. For practical applications, particularly in real-time traffic management where timely responses are critical, it is essential to strike a balance. An effective approach could involve starting with moderate settings and gradually adjusting the nodes and grid sizes based on performance assessments and computational constraints. Besides, we want to highlight that for this study continual learning was not assessed, a possibility mentioned in the original paper [14].

Finally, it is worth remarking that KAN are still in its infancy. KANs have been presented recently (just 2 months ago prior to the submission of this deliverable). With its original implementation for the same number of parameters computational complexity seems to be higher. Nevertheless, due to the learnable activation functions at the nodes, the same performance of MLPs can be obtained with a lower number of parameters. Furthermore, incremental learning can be considered to reduce the complexity [14]. Adaptive grid search training enables reduced computational complexity. Besides, efficient implementations are currently in the scope of many research group in AI worldwide. Recent results in [20] propose a new architecture for implementing KAN achieving 20x faster computation.

CONCLUSION

In this section, we have performed an analysis of KANs and MLPs for satellite traffic forecasting. The results highlighted several benefits of KANs, including superior forecasting performance and greater parameter efficiency. In our analysis, we showed that KANs consistently outperformed MLPs in terms of lower error metrics and were able to achieve better results with lower computational resources. Additionally, we explored specific KAN parameters impact on performance. This study highlights the importance of optimizing node counts and grid sizes to enhance model performance. Given their effectiveness and efficiency, KANs appear to be a reasonable alternative to traditional MLPs in traffic management.

4.2 NEAR-REAL TIME CONTROLLER

4.2.1 Near-real time RIC description

The Near-RT RIC is deployed at the edge of the network to operate control-loops over the CUs and DUs in the RAN, as well as over O-RAN compliant eNBs. Usually, the near-RT RIC controls multiple RAN nodes, so its closed-loop control function is associated with the UEs of several cells. It hosts the termination of the O1, A1, and E2 interfaces, the xApps, and the components required to manage and execute the xApps, [21].The xApp is a plug-and-play application deployed inside the RIC that support custom logic. The xApps receive key performance indicators (KPIs) data from the RAN at all different layers, i.e., user, cell, or slice, and computes and applies control policies. As described on [22], the xApps are defined by the software image and by a descriptor, that includes information on parameters needed to manage the xApp. Near-RT RIC shall consist of multiple xApps and a set of platform functions that are commonly used to support the specific functions hosted by xApps.

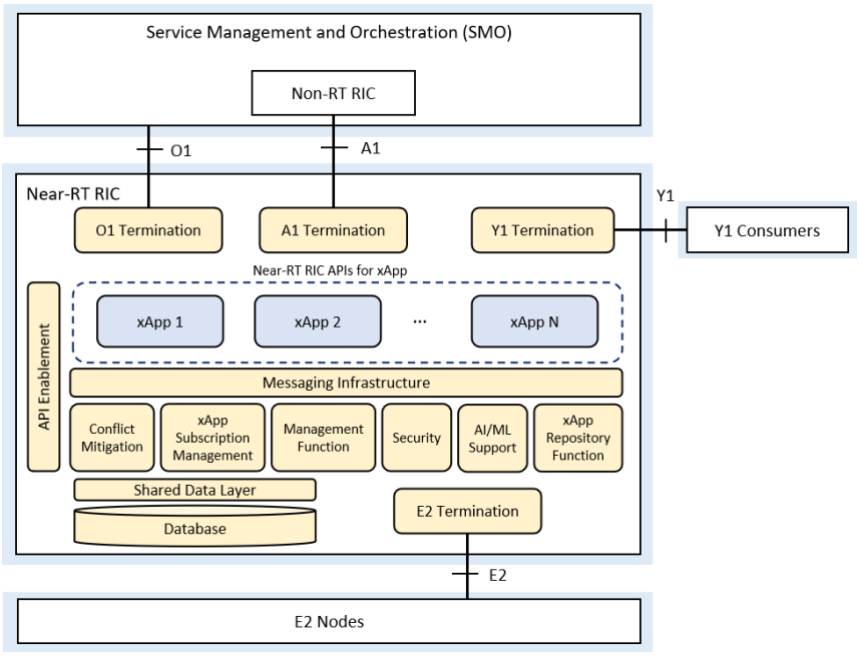

Figure 19: Near-RT RIC Internal Architecture.

An overview of the architecture of the O-RAN standardized near-RT RIC is provided in Figure 19.

The architecture includes:

- **Internal messaging infrastructure**: it offers a low-latency message delivery service among near-RT RIC internal endpoints. It is required to support the following functionalities: (i) registration messages, enabling endpoints to register themselves with the messaging infrastructure; (ii) discovery messages, facilitating the initial discovery and registration of endpoints by the messaging infrastructure; and (iii) deletion messages, allowing for the removal of endpoints that are no longer in use. Additionally, this infrastructure provides APIs for sending and receiving messages from xApps. These APIs can operate based on either point-to-point communications or publish/subscribe mechanisms.

- **Conflict mitigation**: in the context of near-RT RIC, conflict mitigation involves addressing conflicting interactions between different xApps. Typically, an application will modify one or more parameters to optimize a specific metric. Conflict mitigation is necessary because the objectives of xApps may be configured in ways that lead to conflicting actions. The control target of radio resource management can include elements such as a cell, a UE, or a bearer, among others. The control contents of radio resource management encompass access control, bearer control, handover control, QoS control, resource assignment, and more. The control time span refers to the valid duration of control as expected by the control request. Conflicts in control can be categorized as follows: (i) Direct conflicts, which are observable directly by the conflict mitigation process; and (ii) Indirect conflicts, which are not directly observable but may be inferred through dependencies among the parameters and resources targeted by the xApps. Conflict mitigation may involve anticipating potential conflicts and implementing measures to mitigate them.

- **Subscription manager**: the subscription management functionality oversees subscriptions from xApps to E2 Nodes and enforces the authorization of policies that control xApp access to messages. Additionally, it enables the merging of identical subscriptions from different xApps into a single subscription directed toward an E2 Node.

- **Security**: to prevent malicious xApps from leaking sensitive RAN data or from affecting the RAN performance. The details of this component are still left for further studies;

- **Network Information Based (NIB) Database and Shared Data Layer API**: the RAN NIB contains information about the E2 nodes, while the UE-NIB includes the identification and entries of the UEs. The shared data layer (SDL) is utilized by xApps to subscribe to database notification services and to read, write, and modify information stored in the database. UE-NIB, R-NIB, and other use case-specific information may be accessed using SDL services.

- **xApp management**: this service features automated lifecycle management of xApps, encompassing onboarding, deployment, termination, and the tracing and logging of FCAPS.

- **AI/ML support**: The AI/ML data pipeline in near-RT RIC facilitates data ingestion and preparation for xApps. The input to this pipeline may include E2 node data collected via the E2 interface, enrichment information from the A1 interface, information from applications, and data retrieved from the Near-RT RIC database through the messaging infrastructure. The output of the AI/ML data pipeline may be supplied to the AI/ML training capability in near-RT RIC. The AI/ML training in near-RT RIC enables the training of xApps within the system, offering generic and use case-independent capabilities to AI/ML-based applications, potentially benefiting multiple use cases.

4.2.2 Radio resource management problems to be handled

Previous works in the near-real time RIC literature show that AI techniques can be effectively implemented to tackle several RRM tasks. While the models can be trained in non-RT, at the non-real time RIC, they can be forwarded to the near-real time RIC for inference. To the best of our knowledge, there are no previous works evaluating AI techniques for RRM in the near-real time RIC in NTNs. Thus, this section provides an overview of such literature from the TN perspective.

LOAD PREDICTION

In order to ensure that the radio resource allocation can cope with upcoming traffic requests, while also minimizing the number of unutilized resources, the radio resource allocation should be performed taking into account historical traffic data, e.g., time series of the cell load. For this reason, load prediction techniques can be implemented to extract the forecasted load in a cell, or in the network, based on historical statistics. Such predictions can be inferred within the near-real time RIC and incorporated in the radio resource allocation strategy, resulting in a potentially more optimal channel allocation, power allocation, network slicing, etc. In particular, [23] assessed the load prediction performance of 3 techniques, namely the linear regression, the Recurrent Neural Network (RNN) AI model, and the long-short term memory (LSTM) RNN. The work considers a network of gNBs with several mobile users of different time, each generating traffic of various size, with each device being characterized by a mean handover ratio based on the time of the day and the type of device (e.g., IoT device, cell phone). The models were trained to forecast the data load in an upcoming time slot based on the average data rate in the last 2 time slots, and the percentage of change in data rate between the last time slot and the second, third, and fourth to last time slots. After training, the LSTM and the RNN models scored comparably in the load prediction task, achieving a MAPE of 0.544 and 0.560, respectively, while the linear regression model scored a MAPE of 0.615. The authors highlighted that LSTM resulted being more successful in predicting steady loads, while RRM achieved better performance in the prediction of load peaks. An LSTM model was also trained in [24] to predict the cell load within the following 15 minutes based on historical cell load data with 15 minutes resolution. The optimal time window for the input time series was empirically observed to be 1 day, scoring an MSE of $1.1 \cdot 10^{-3}$Gbps2. The authors in [25] also chose an LSTM model to predict the traffic in terms of downlink PRBs utilization and average user-perceived IP throughput, forecasting such metrics in the following hour for each gNB in the network based on a time series of 12 hours. The predictions, which are obtained with an accuracy of 92.64%, are then used to split the congested cells, e.g., by turning on femtocells. Load prediction has been carried out to provide aid to network slicing, too. [26] analysed the performance of LSTM, a Sequence To Sequence (Seq2Seq), and a temporal convolutional network (TCN), on the prediction of packet data convergence protocol (PDCP) traffic volume. Depending on the available data sampling rate, ranging from 5 minutes to 30 minutes, a different model provides the best RMSE. At the extremes, the Seq2Seq model achieves the best performance, scoring an RMSE of 76.13 Mbps and 56.48 Mbps for the sample rates of 5 minutes and 30 minutes, respectively. The predictions are then used to allocate PRBs to 2 different network slices with a traditional, i.e., non-AI, algorithm. Similarly, in [27] a proactive network slicing xApp was developed to allocate physical resources to RAN slices. The allocation takes as input the LSTM-predicted Medium Access Control packet sizes in the upcoming 100 TTIs, which are obtained from the time series during the most recent 1000 TTIs. The prediction model scores and accuracy of 92.45%.

TRAFFIC ANOMALY DETECTION

High data rate peaks are typically related to specific hours of the day and geographical locations. However, specific events, e.g., a sport match or a parade, may lead to unexpected traffic requests. For this reason, the radio resource allocation may integrate the detection of traffic anomalies to improve the resource utilization. Under this framework, the authors in [23] proposed a logistic-regression-based solution and an XGBoost ML model to identify anomalies in the data rates of 5G cells. The logistic regression and XGBoost algorithms scored an area under the curve of the receiver operating characteristic (AUC-ROC) of 87.0% and 90.7%, respectively.

POWER ALLOCATION

With strong implications on user throughput and energy consumption, power allocation is one of the most important RRM tasks. In [24], a multi-agent deep reinforcement learning (DRL) framework is developed to determine the optimal power allocation of RUs based on UE measurement reports, with the objective of maximizing the network energy efficiency under station-specific power and user-specific throughput constraints. With respect to the initial conditions, set to an equal power distribution over the PRBs of each RU, the DRL model achieves a 72.3% improvement in energy efficiency, computed as the total data rate over the overall power allocated to RUs. Under the RAN slicing paradigm, a power allocation xApp is implemented in [28], together with a radio resource allocation xApp, with the aim of improving the bitrate of enhanced Mobile BroadBand (eMBB) slices and reducing the delay for ultra-reliable low latency communication (URLLC) slices. In particular, a deep-Q-network (DQN) is deployed to set the power level of gNBs, optimizing the performance of the 2 types of slices and penalizing high power consumptions. Under a federated learning framework, the learnt Q-tables are processed together with the Q-tables obtained in the resource allocation xApp to jointly optimize the policy to be taken. The power allocation problem is also tackled jointly with user association and resource allocation in [29] through the adaptive genetic algorithm (AGA). The power allocation subproblem is formulated as a dual “0/1” multiple knapsack problem, where each knapsack represents a gNB and each item a UE. The objective is the maximization of the user throughput, considering power consumption as item weight, user throughput as item value, and power capacity as knapsack capacity. The joint approach to power allocation and resource allocation is also taken in [30], where a DQN is first used to choose a primary power based on the transmission rate, the initial transmission power, the channel state information (CSI) and the length of queued data in the buffer. Based on the set transmission power values, a resource allocation DQN performs the PRB allocation between gNBs and UEs. Finally, a finer power allocation is carried out by a dedicated DQN, considering the chosen resource allocation.

RADIO RESOURCE ALLOCATION

The allocation of radio resources, e.g., of PRBs to UEs, is the core RRM task. As the time and frequency resources are scarce, an algorithm should be implemented to determine what is the optimal use of such resources, i.e., which UEs should be allocated to the available PRBs to maximize identified metrics. An xApp for the optimization of the resource block groups to UEs allocation among multiple cells has been tackled in [31], comparing the performance of the advantage actor critic (A2C) and parametrized policy optimization (PPO) algorithms. Both the algorithms aim at maximizing the quality of experience (QoE) within the network considering the channel request, the channel quality indicators (CQIs) from the CSIs, the data rate, and the UE fairness for each UE. The work has shown that PPO converges to the maximum QoE faster and in with a steadier behaviour in time compared to A2C. In [28], the resource allocation is carried out jointly with a power allocation xApp, aiming at optimizing the resource usage among different network slices using a DQN. Resources allocated to eMBB slices provide higher rewards if they provide a large increase in throughput, while rewards associated to resources allocated to URLLC slices are dependent on the resulting

communication delay. Through the joint optimization of the power and resource allocation, the authors showed that the federated technique achieve 11% higher throughput for eMBB slices and 33% lower delay for URLLC slices with respect to the independent optimization. In [30], after a first round of power allocation with a dedicated algorithm, a DQN is implemented to perform the allocation of PRBs to UEs at each base station. The algorithm considers as state information the transmission rate, the allocated transmission power, the CSI, and the length of the queued data in the buffer, aiming at maximizing the total throughput of each base station. A finer second round of power allocation is then carried out based on the decided resource allocation. With each agent (gNB) sharing its state information with other agents, the resulting team-learning-based DQNs achieve up to 8.8% higher throughput and 64.8% lower packet drop ratio than the same models running on local information only.

TRAFFIC STEERING

To manage the UE mobility from one cell to another, the RAN must implement techniques to handle handovers and dual connectivity. Through traffic steering, the UE connectivity can be managed by selecting primary and secondary cells, associating users to cells, and triggering handovers between cells. In [29] t, the user association task is solved together with the user association task in a dual “0/1” multiple knapsack problem, using AGA as solver. The objective is to find how to associate users (items) to gNBs (knapsacks) given the user demands (weights) in order to maximize the total available throughput (knapsacks’ total value), under the given maximum gNB capacity constraints. Under video streaming and voice over IP traffic requests, the proposed algorithm achieves an increased QoE with respect to the default genetic algorithm and the simple allocation to a single base station. The authors in [32] developed a traffic steering xApp to manage the user association to target Primary cells of the secondary node. The task is formulated as a Markov decision problem, which is solved with the use of a CNN-based DQN. The solution aims at maximizing the weighted cumulative sum of the logarithmic throughput of all the UEs across time, introducing a penalization of the reward based on the number of experienced handovers. The authors showed that the proposed algorithm is capable of increasing the overall spectral efficiency with respect to multiple benchmark algorithms under different types of traffic (video streaming, web browsing, and instant messaging).

Clearly, the literature has proven that the near-RT RIC is able to tackle several RRM tasks in TNs through the implementation of AI algorithms. However, such algorithms should be evaluated in the context of NTNs to assess the achievable gains of the techniques in satellite-based communications. Due to the extensive coverage area of communication satellites, as well as the short visibility windows due to the satellite’s movement, RRM is particularly crucial in NTN. For example, as reported in [33], in the context of user-centric beamforming in LEO-based NTNs, the beamforming performance are strongly dependent on the channel matrix rank, where the channel matrix is the matrix of CSI vectors (*i.e.*, the vector of propagation channel coefficients between a UE and each radiating feed on board of the satellite) of the UEs scheduled for transmission within a considered time slot. Hence, resource allocation, with an emphasis on user scheduling, is critical for the maximization of the user-centric beamforming performance. Furthermore, in the case of CSI-based user-centric beamforming, the beamforming vectors are computed on feedback CSI relayed by the UEs, which are subject to channel aging. Depending on the user-centric beamforming architecture, the aging interval can assume different durations and introduce strong degradation in the CSI accuracy, resulting in large inter-beam interference [33]. For this reason, AI-based channel prediction techniques may be implemented to counteract this effect, *e.g.*, [34], [35].

4.2.3 AI-techniques for near-real time RRM

In the context of user-centric beamforming, the user scheduling RRM algorithm can be formulated as an optimization problem, aiming at maximizing or minimizing an objective function $F(\cdot)$, *e.g.*, system capacity, per-user throughput, SINR, user fairness, or a combination of multiple identified KPIs. The scheduling task can be subject to several constraints, which can limit the search space for the optimal RRM solution. Per-UE throughput requirements may be set for all the K users to ensure that a minimum performance level is provided within the coverage area. If priority classes are introduced, more stringent throughput requirements can be set for high priority users. Furthermore, based on the set scheduling window, a maximum number of time slots T_{max} for the user allocation can be set as an additional constraint. Finally, constraints can be associated to the variable to be optimized, *e.g.*, a $K \times T_{max}$ binary matrix A representing the association of K users and T_{max} time slots. Based on the available user information I (*i.e.*, the channel coefficient of correlation matrix for CSI-based scheduling or the inter-users great circle distance matrix for location-based scheduling [36]), the optimization function $F(A, I)$, and the set of M constraints $\mathcal{C} = \{C_m(A, I)\}_{m=1,\dots,M}$, the optimization problem can be formulated as:

$$\min_{A} F(A, I)$$
$$\text{subject to } \mathcal{C}$$

Due to the non-linearities present in typical optimization functions and constraints, AI algorithms, with emphasis on unsupervised ML techniques, can be implemented to learn patterns from a training dataset. In particular, DRL algorithms, like DQN, can be trained to determine the optimal policies, *i.e.*, the optimal user scheduling. Q-Learning introduces the concept of state-action value function (or Q-function) of a policy π, $Q^{\pi}(s, a)$, which represents the expected reward obtained by taking action a from a certain state s and following policy π for the next steps. The Q-function that follows the optimal policy is represented by $Q^*(s, a)$ and can be approximated by iteratively using the Bellman optimality equation, *i.e.*, by computing $Q_{i+1}(s, a) \leftarrow \mathbb{E}\left[r + \gamma \max_{a'} Q_i(s', a')\right]$ at time step i [37], where r represents the immediate reward achieved by taking action a from state s, (s', a') are the generic future state and action, and the maximum reward obtainable at state s', $\max_{a'} Q_i(s', a')$, is reduced through a discounting factor γ. In DQN, a deep neural network (DNN) is used as a function approximator to learn the optimal Q-function, *i.e.*, to obtain $Q^*(s, a; \theta_i) \approx Q^*(s, a)$ given the DNN parameters θ_i. In particular, the DNN is trained to minimize at each step i the temporal difference error:

$$L_i(\theta_i) = \mathbb{E}[(y_i - Q(s', a'; \theta_i))^2],$$

$$y_i = r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}).$$

Taking into account user scheduling, an episode can correspond to one scheduling window, and each step within an episode can correspond to a scheduling time slot. To perform user scheduling, a scheduler requires the available user information I, which should then be contained in the state s. The action a can be represented by the K-elements-long binary scheduling vector, in which a 1 in position k implies that user k is scheduled within the current time step. The reward r should consider one or more KPIs resulting from the user-centric transmission towards the scheduled users, depending on possible additional state information. As an example, r can coincide with the average SINR among the scheduled users during the current time slot, assessed through the computation of the corresponding beamforming matrix. By including in s the number of times each user has already been scheduled within the current episode, fairness metrics can be included within the reward (*e.g.*, introducing a penalization for each user that has not been scheduled by the end of the scheduling window, or a reward based on the increase of a fairness KPI, such as the Jain index). Constraints on the SINR,

e.g., minimum SINR levels for users with high priority, may be mapped to a reduction in reward if such requirements are not satisfied. Similarly, different types of services may be considered by including each user's traffic demand in the state s and introducing the corresponding reward penalizations in r [38].

5 RAN SOFTWARIZATION

5.1 GENERAL STACK MODIFICATIONS TO ENABLE POC

5.1.1 RIC – RAN interface to enable monitoring and control

The complex nature of the PoC means that the ability to monitor and control different functions and procedures at various levels of the RAN stack was essential.

To that end, a critical part of the RAN software extension was to enable such functionality. Instead of designing and implementing a bespoke system to achieve such functionality, it was decided to leverage the abilities of the O-RAN E2 interface to provide this functionality.

O-RAN E2 interface facilitates communication between the RIC and other RAN elements, specifically the DUs and CUs. It enables the RIC to exert control over the RAN, allowing for the transmission of data and control messages. This capability supports the management of radio resources, monitoring of network performance, and implementation of network policies.

The functionality of the E2 is split into two aspects, the E2 application protocol (E2AP) and the E2 service models (E2SM):

- E2AP is the protocol framework used for communication between the RIC and E2 nodes, such as DUs and CUs. It manages the setup, modification, and release of E2 connections and ensures the reliable delivery of control messages.
- E2SM, on the other hand, defines specific functions and data structures for particular RAN management tasks, such as traffic steering, quality of service (QoS) management, and anomaly detection.

The implementation of specific service models is what provides the capabilities of monitoring and control that are of interest to our PoC implementation, specifically the key performance measurement (KPM) service model and the remote control (RC) Service Model.

The KPM service model defines the framework for collecting, reporting, and analysing KPIs to monitor and optimize RAN performance and the RC Service Model specifies the mechanisms for dynamic control and configuration of radio network parameters to optimize performance and resource utilization in the RAN.
In the srsRAN, the high-level architecture of the E2 agent is depicted in Figure 20. We have the following top-level components:

- ***E2 Setup Procedure:*** This component manages the establishment of the connection between the E2 agent at the RAN and the RIC. It will communicate the supported E2 capabilities in this particular RAN, known as the RAN functions. This block will establish what RAN functions will be active during the session.
- ***E2 Subscription Procedure****:* This component manages the establishment of subscriptions between the RAN and the RIC. The subscription procedure will communicate details of which metrics are being requested by the RIC and in what way they are to be reported. The procedure will then work in cooperation with the subscription manager to check if the request is compatible with what is supported in the RAN and set up the flow of data accordingly.
- ***Subscription Manager:*** The subscription manager is charged with handling all tasks related to the subscriptions. It will check if the requested metrics are supported, it will start and handle the indication message procedure and it will manage the interaction with the RAN stack. Once the subscription is finished it will perform the cleanup.

- ***RIC Control Procedures****:* The RIC control procedure handles the process by which parameters within the RAN can be changed by the RIC. This complex process will inject the request into the RAN, manage the reconfiguration and return the result to the RIC.
- ***E2SM Manager****:* The E2SM manager handles all the data structures related to the different service models supported by the RAN. It enables the adding and removing of services and provides the interfaces to the different service objects to the rest of the components.

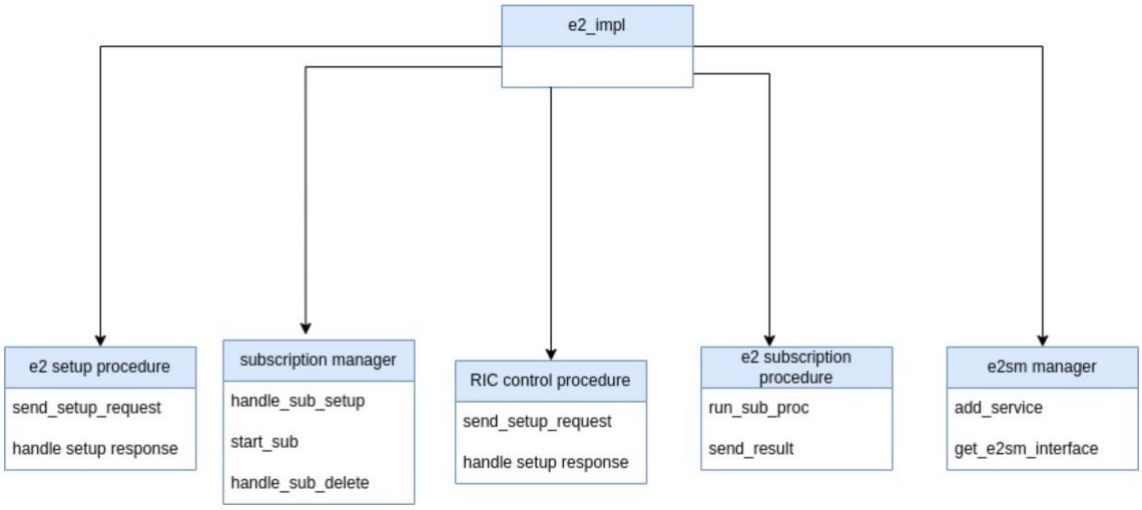

Figure 20: High-level architecture of E2 agent in srsRAN.

5.1.2 API creation for RU integration at desired split level

In O-RAN architecture, the distribution of radio stack functions between the DU and RU can follow several different split options, each with its own set of benefits and trade-offs. Here are some of the most commonly used DU-RU split options:

- Split 8
- Split 7.2x
- Split 6

Split 8 places the whole PHY at the DU side and only sends time-domain in-phase and quadrature (IQ) samples over the DU-RU interface, this is the simplest option but requires a very high data rate, for large bandwidths (100MHz+) this becomes impractical.

Split 7.2x allows for lower fronthaul bandwidth requirements by pushing some of the PHY processing to the RU but requires higher complexity at the RU, additionally it still has stringent latency requirements for the fronthaul as with Split 8.

Split 6 places the entire PHY at the RU which significantly relaxes the latency requirements of the fronthaul, however it significantly increases the complexity of the RU.

The 7.2 split has emerged as the favoured ORAN split because of its lower complexity RU design, lower bandwidth requirements and simple interface allowing for easy inter-vendor integration. This is the split that will be utilized in 5G-STARDUST.

The OFH is the communication interface used to link a DU with one or multiple RUs. It implements split **7.2x** which balances the trade-offs between keeping the RU as simple as possible and reduce the interface throughput. It is internally split into the following logical planes:

- CUS plane: control, user and synchronization.
- M plane: management.

The CUS plane is further split into different data flows to exchange data between the DU and the RU.

- User plane:
    - Data flow 1a: flow of IQ data in fast Fourier transform (FFT) frequency domain on DL
    - Data flow 1b: flow of IQ data in FFT frequency domain on UL
    - Data flow 1c: flow of physical random access channel (PRACH) IQ data in FFT frequency domain
- Control plane:
    - Data flow 2a: Scheduling commands (DL and UL) & beamforming commands
    - Data flow 2b: License assisted access (LAA) listen-before-talk (LBT) configuration commands and requests
    - Data flow 2c: LAA LBT status and response messages
- Synchronization plane:
    - Data flow S: timing and synchronization data

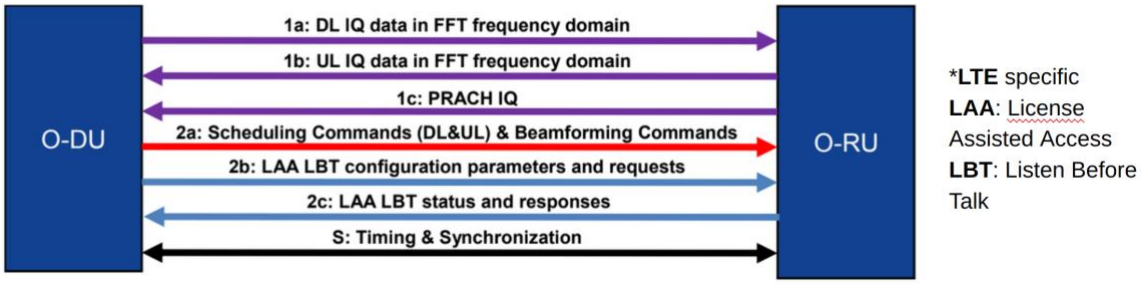

Figure 21: OFH dataflows

Each OFH message belongs to a specific data flow that is encapsulated in an enhanced common public radio interface (eCPRI) message which is transported over an Ethernet frame.

The SRS OFH implementation has been split into different components to make it modular, easy to extend, easy to maintain and easy to test.

It is mainly split into a transmitter component and a receiver component: each of which contains the corresponding data flows previously mentioned. Individual libraries have been developed for different compression algorithms, eCPRI packing and unpacking and raw Ethernet frame management.

Additionally, for performance reasons, both the transmitter and the receiver are executed concurrently in different threads to minimize the latency required to process the IQ samples. Our implementation leverages the 7.2x split within the ORAN architecture, utilizing the OFH to allow for simplicity in the RU while ensuring robust performance and modular design in the DU, ultimately enhancing interoperability and minimizing latency through concurrent processing.

5.1.3 RAN modifications for NTN

The implementation detail of the RAN modifications made for NTN have been covered in [39], here we provide a brief summary of that work to provide appropriate context for this deliverable.

This part of the RAN softwarization task involved extensive modifications to the gNB-A components to enhance functionality for NTNs. The gNB-A is divided into the CU and the DU. The CU includes the PDCP, SDAP, and RRC, while the DU is split into DU-high (containing RLC and MAC) and DU-low (containing PHY). Key modifications focus on the DU-high, particularly on the scheduling offset, hybrid automatic repeat request (HARQ) extensions, and RLC enhancements. For the scheduling component, adjustments are required to accommodate increased delay (*ntn_k_offset*) for uplink scheduling, impacting queues for various messages, including scheduling requests and HARQ acknowledgments. The HARQ procedure enhancements involve either disabling HARQ and relying on automatic repeat request (ARQ) in RLC or extending HARQ processes to 32, necessitating changes in data handling and process management. The RLC extensions involve increasing retransmit times and buffer sizes, requiring additional memory and ASN1 adjustments for communication. The RRC modifications in the CU involve generating and broadcasting the SIB19 message, integrating ASN1 packing mechanisms, and updating fields with satellite orbital information.

These modifications we performed on top of the SRS 5G RAN and are being integrated with the other enhancements made to the RAN as part of this work package for use in the PoC.

5.1.4 Development of 60KHz SCS to enable FR2

In the 3GPP Release 17, NTN bands (n255 & n256) have been established in the S-band, which is frequency range 1 (FR1). However, in later releases, it is expected that new NTN bands will be established in Ka/Ku which will be classified as frequency range 2 (FR2) bands.

Establishing a multi-carrier link over Ka/Ku bands present a different challenge to that of S-band. The main reason being the nature of the Doppler shift involved at these higher frequencies. At S-band frequencies, the maximum doppler shift is ~40kHz whereas in Ku/Ka bands doppler shifts can be as high as 400kHz. While the NTN procedures in the UE and gNB are designed to compensate this effect, there will always be a certain amount of residual doppler. For S-band this is easily handled by carrier frequency offset (CFO) compensation techniques that are common in multi-carrier cellular systems but in Ka/Ku bands this residual doppler has the potential to grow to a level that is not easily compensated CFO compensation techniques. The ability to effectively compensate CFO in a multi-carrier system is a function of (among other things) the subcarrier spacing (SCS) of the waveform. In Figure 22 we can see this fact represented graphically. The amount of intercarrier interference that takes place as a function of Doppler (normalized to SCS) shows us that a doppler shift of 30% of the SCS will result in a signal-to-interference ratio (SIR) of 8dB. This is probably the limit of what is reasonable to a system to function well, meaning that for 30kHz, a maximum of 9kHz residual doppler is tolerable, only 2.25% of the total possible doppler. Given that there is likely to be some unavoidable error in the doppler correction, 2.25% is not a large margin. It is for this reason that higher SCS is considered for FR2 cases.

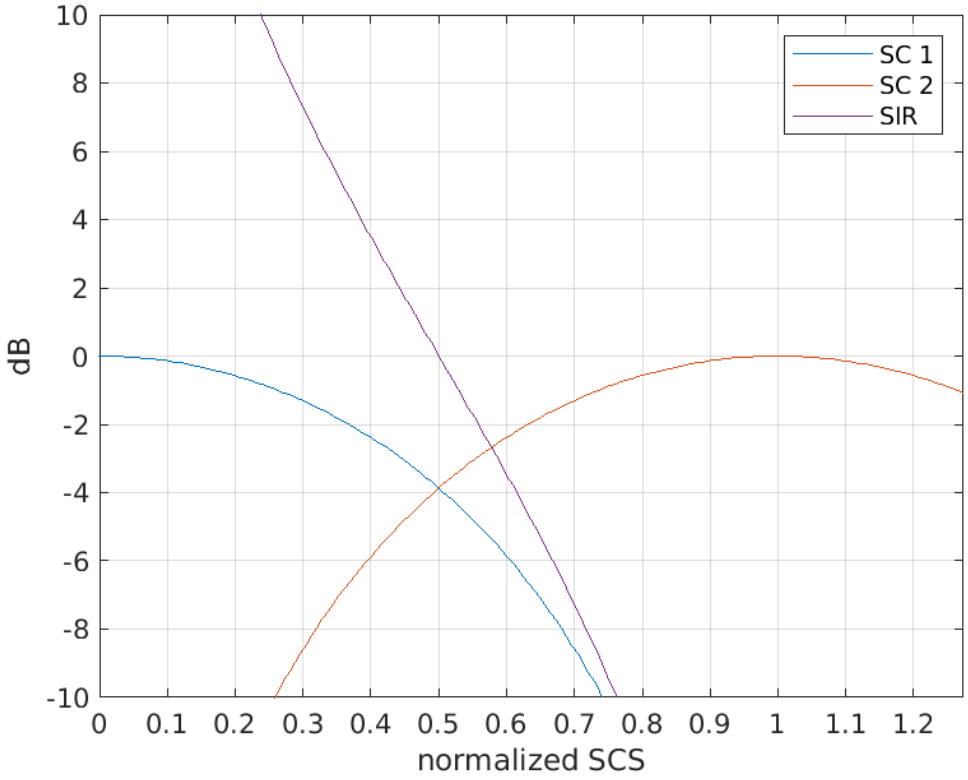

Figure 22: Inter-carrier interference

6 ONBOARD PROCESSING

6.1 INTRODUCTION, CONTEXT & DRIVERS

The development of 5G, and long-term 6G NTN systems makes on-board processing with fully regenerative payload a major trend that tends to be generalized, especially for telecom constellation systems.

When it comes to design processing capabilities, two sides shall be jointly considered:

- Pros and opportunities brought by processing: performances (e.g. gain on signal-to-noise ratio (SNR), system capacity, etc.) and provided features (internal system features that help system design, or value-added service directly perceived by users and customers)

- Constraints and impacts: such as requirement related to processing, CPU, memory, power consumption, implementation, mass and sizing impact, resistance to radiations, and maybe thermal impacts. From these perspectives, any processing resources are always sized at the right level in operational systems and processing margins are sized to be limited for this reason. At the same time, in system with 5-10 years of satellite lifetime, the exact need in terms of telecom service may be hard to anticipate and size, particularly when thinking to the generalisation of software processing related to the processing of telecom services. Ideally, expectations from operation and service providers would be to integrate satellite constellations in Cloud infrastructures with the same level of flexibility and number of resources as found in ground.

Such trade-off is hard to achieve, and generally speaking the exact mission and system objectives will allow to determine what are the limits to consider - hence on a per system basis. In short-term application, and first generation of 5G constellation system, the target cost to consider for wide-scale constellation is in Billions of dollars. **System offered capacity** (total Gbps) and **achieved bit rate** per users (peak/average) are among the first KPIs of interest. The offered Gbps capacity per satellite can also be derived.
It means that this amount shall be:

- Supported in every external communication interface

    - Feeder link (satellite-to-Gateway), possibly with multiple Gateways at the same time
    - On inter-satellite link (ISL), if the system is equipped
    - On the user link. This represents on the user link ~50 to 500 MHz of spectrum band to process, and frequency reuse may apply among different spot beams raising this even higher.

- Supported by the OBP router / switch (sum of all data rates)

- Supported on all internal communication interfaces (for example between radio access layer and infrastructure-level network layer)

Other drivers can be cited:

- The development of 5G/6G NTN standard remains without any large-scale deployment today, letting aside non-standard 4G based solutions, and/or in-orbit experimentations. The 5G/6G found its origin in terrestrial networks where the resources are “unlimited”. Do

the requirements and 3GPP specifications (at least in short-term) will actually be feasible in operational system where satellites are used as (partial) 5G base stations? For processing requirements, the short-term critical questions relate to:

  - The question of high frequency update (more CPU) demanded by 3GPP process (user plane, control plane) and by the supporting means such as active antenna for near real-time beamforming
  - The memory requirement can be significant (upon the exact role of the satellite and split of on-board function; could be several Gbytes to tens or hundreds of Gbytes)

- Application on the virtualization concepts and technologies on the execution platforms:

  - Strong incentives to support and segregate distinct missions, and/or services from distinct operators; they can manage services on their own, on the other hand in a context of limited resource it is difficult to ensure an optimal allocation in particular taking into account variations (*dynamic and elastic partitioning*) with generally some pre-empting usages.

- Achieved QoS and QoE:

  - In addition to data rates 5G/6G users will expect low and stable end-to-end delay, while access to the majority of 5G services known in TN.
  - Continuity of service: transparent handover, with no/few packet losses.

Last, impact on 3GPP standard may also have to be considered. For example, some implementation feasibility could lead to introduce features in the scope of desired Release. Even if deviations can always be implemented on a per system basis, they should be minimized to guarantee the long-term support of the feature.

BEAM MANAGEMENT

When it comes to digital processing on-board topic, the antenna beamforming strategy is among first topic to consider for the antenna beam laws computation and application. At system level, the question of Earth moving beams vs Earth fixed beams require first a trade-off analysis. Beam size determination can also be jointly considered, if not already known in customer specifications.

Table 4: Earth moving beams vs Earth fixed beams comparison

	PROS	CONS
Earth moving cell	Simpler satellite beam management (for antenna and its Beam Forming Network) & Radio Parameters configuration	**QoS and Performance:** • Frequent radio HOs. • Impact on UE Throughput (overhead due to heavy HO singnalling)
Earth-fixed cell	**Smooth impact on 5G 3GPP standard and against exisitng Terrestrial Network** **QoS and Performance:** • Less Frequent radio Hos	More complex satellite beam generation. Potentially requires additional change/adaptation with respect to radio parameters configuration.

	• Less Impact on UE Throughput	

More in detail:

- From QoS and performance perspective, moving cells may suffer from recurrent handovers (HOs): In these cells, the signalling overhead due to HOs may range from tens of kbps at the nadir to hundreds of kbps at satellite coverage border.
- For earth fixed cells the HOs happen in bursts. The peak in number of handovers can be lowered by extending the overlapping period (this is implementation dependent, e.g. constellation design).
- Due to the asynchronous nature of the NR Release 15 handover functionality with random access at every cell change, there is an undesirable temporary data interruption gap at every handover. This is impacting mainly the moving cell, the connected UE is non-stop handed over between cells. This may degrade the end-user QoE.
- Another main KPI to be considered when comparing fixed and moving cells is the handover success rate. This KPI is likely to be degraded on moving cell due to fast mobility and high volume of Handovers (more than 250000 HOs/h in case of moving cell of 60km size). Furthermore, in case of a handover failure, the UE cannot return to old cell (due to the motion of the source moving cell) which means the call will be simply dropped. Because the moving cell is suffering largely from the fast mobility and high volume of handovers to be handled, we can expect that: the overall HO success rate for the moving cells will be lower compared to fixed cells, while the call drop rate will be greater compared to fixed cells.
- When it comes to tracking area management, for Earth fixed cell deployment: Earth Fixed timing advance is used. Thus, no enhancement will be needed in the standard. For moving cell: Earth Fixed timing advance or moving timing advance can be used. This will need some modification in the 3GPP standard.

Overall, the earth fixed cells are the preferred solution with respect to QoS and performance and smooth impact of 3GPP standard.

In the following we consider 3 incremental on-board architectures, each requiring more demanding resources, but also offering more features.

We could associate:

- CU/DU splitting (split 2) more relevant for the short-term application (current constellation design)
- Full gNB on-board to mid-term (3-5 next years)
- Full gNB + user plane function (UPF) for even long-term (>5 years)

6.2 CU/DU SPLITTING

The first option is to consider gNB CU/DU split, and according to the on-board and ground architecture presented in Figure 23.

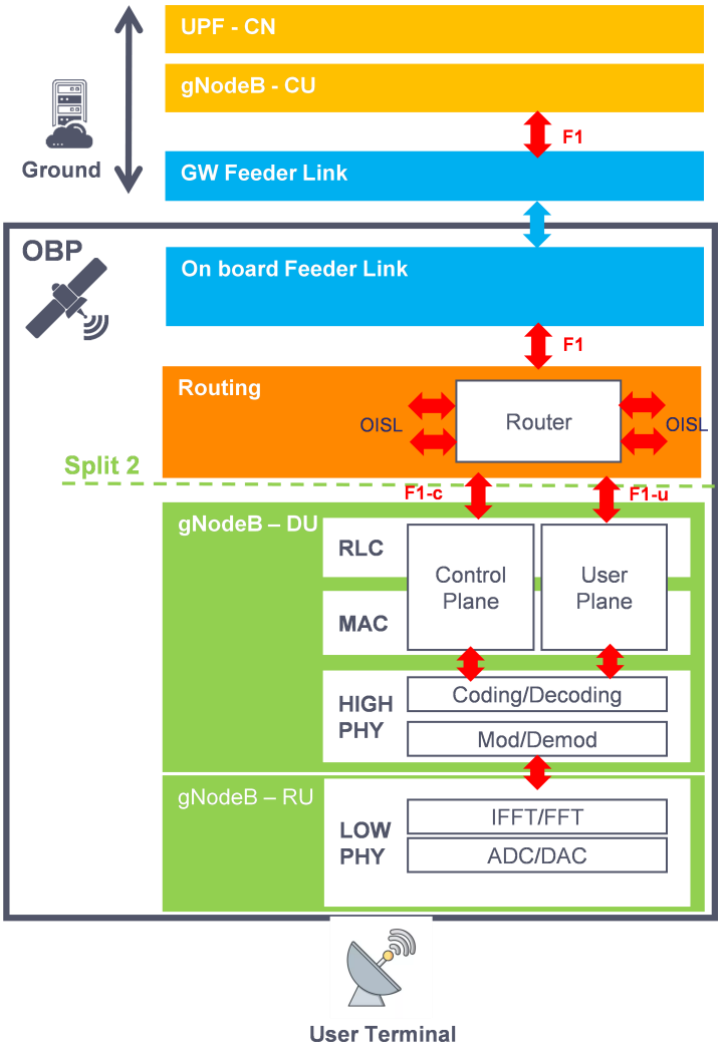

Figure 23: Payload and ground architecture - split 2 (DU/RU on board, CU on ground)

The physical layer is responsible for coding, modulation, beamforming, and mapping of the signal to the appropriate physical time-frequency resources. It provides services to the MAC layer in the form of transport channels and the overall transport channel processing is mostly the same in UL and DL.

A cyclic redundancy check (CRC) for error-detecting purposes is added to each transport block, followed by error-correcting coding using low density parity check (LDPC) codes. Rate matching adapts the number of coded bits to the scheduled resources, the code bits are scrambled and fed to a modulator, and finally the symbols are mapped to the physical resources, including the spatial domain. For the UL there is also a possibility of an FFT-precoding to reduce the peak-to-average-power ratio (PAPR) of the signal at the amplifier input.

One key aspect of this processing chain is the multi-antenna precoding, i.e. beamforming. The purpose of the precoding is to map the different transmission layers to a set of antenna ports using a precoder matrix to provide directivity and focus the overall transmitted power in a certain direction. In the context of a digital beam forming network, this operation may present a challenge in terms of implementation complexity, especially in a fixed beam scenario where the beamforming laws must be updated frequently.

At access layer, one important function to perform is the real-time L2 scheduling. Implementation of the scheduler shall be carefully investigated: firstly, its performance is detrimental to delivered QoS, and shall be accommodated with the relevant transmission channels (FR1 or FR2, according to the type of satellite channel). The typical dynamicity (few ms) to consider is different on the satellite (even LEO) case than for terrestrial case and could be relaxed on a ~10 ms basis, or more with Gaussian channels (FR2 case in particular). A positive impact for alleviating processing, if a starting implementation was used from a TN software stack would be expected.

6.3 FULL GNB ON-BOARD

In a second architecture, the full gNB is envisaged on-board (see Figure 24).

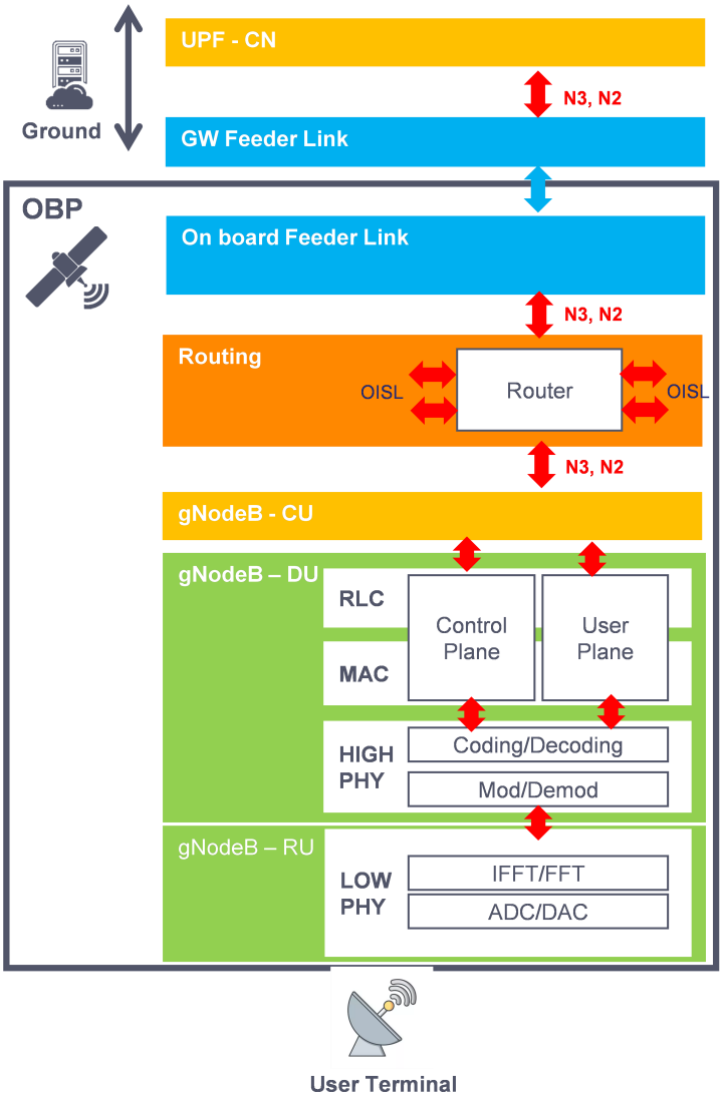

Figure 24: Payload and ground architecture - gNB on board and UPF/CN on ground

Note that split can have various impact on system levels implication, but this analysis exceeds the scope of this discussion.

The full gNB implementation will add processing and network load to consider, in particular for the short-term.

Architecture with full gNB on-board makes easy the implementation of RAN-Core interconnection conceptually, while validation of communication interface shall still be carefully investigated in the context of NTN. Industrially speaking, it is also of interest. However, no application layer nor caching functions can be implemented on board in this architecture as the N3 interface tunnels the traffic between the gNB and the UPF. Full gNB can be an intermediary step to the last architecture.

6.4 FULL GNB ON-BOARD + UPF

The third architecture envisages the full gNB on-board with UPF functions (see Figure 25).

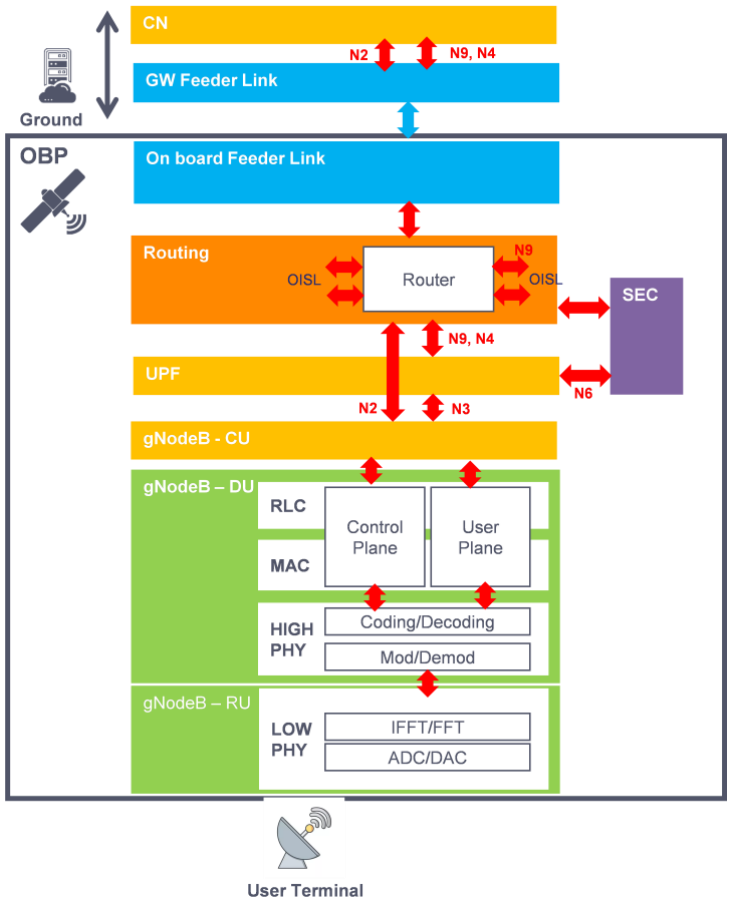

Figure 25: Payload and ground architecture - gNB + UPF on board and core network on ground

This third architecture is compatible with mesh connectivity through the N9 interface at the OISLs. It can provide the ability for a user terminal to reach back without the need for anchor relay. The consideration of limited power and computing resources further highlight the need for efficient processing of traffic at gNB + UPF side, and algorithms that optimize resource usage while minimizing energy consumption. Potentially only a subset of the traffic could be fully processed on-board enabling Space Edge Computing (SEC) and Mesh only for this part of the traffic.

This architecture is also compatible with SEC as the N6 interface of the UPF enables accessing to the PDU layer which makes it then compatible with application layer and caching functions.

SEC[1] refers to the deployment of computational resources and data processing capabilities closer to the data source—in this case, on satellites payload. This approach aims to reduce latency, optimize bandwidth usage, and enhance the efficiency of satellite communications by processing data at the "edge" of the network rather than transmitting it back to centralized data centers on Earth.
However, to provide these functions the satellite needs to be equipped with additional OBP units, such as CPUs, GPUs, or specialized accelerators, or mass storage. These nodes can perform data analysis, filtering, caching and decision-making tasks in orbit.

This is one of the main challenges of SEC as satellite have limited power resources, size and mass, and the processing of the traffic is already demanding which may leave few processing units for other tasks or value-added application. It's a tradeoff (economical and technical) that should be done between the processed traffic and the capacity to perform application layer or caching function that can provide added value.

In some use cases processing data locally on the satellite can help that only relevant or summarized information is transmitted back to Earth, to significantly reduce the volume of data that needs to be sent over potentially limited bandwidth links. Edge computing can also minimize the time it takes to process and respond to application layer request, as it avoids the delays associated with transmitting data to and from ground stations.

However, one of the big challenges in SEC for LEO constellations is the fact that the satellite is moving and then the edge for the UE is frequently changing, making needs for SEC HO and context exchanges.

Further, the SEC approach will be a concrete case to instantiate concepts of payload (or OBP) resources. OS virtualization (Hypervisor type 1 or 2) or container-based approach (Linux Docker) are very popular solutions. Not speaking about the services (e.g., data analysis, filtering, caching and decision-making, etc.) the top-level implementation issues to consider in the design choices include:

- How the computing resource arbitration is managed among the service (soft / elastic partitioning)? or hard partitioning? is the ground involved in the decision process, what is the dynamicity to be expected?

- What about segregation / isolation constraints between the OS/containers from the Cyber-security perspectives? is the qualification level of cyber-security of the hypervisor relevant to the specific services and missions envisaged?

[1] Space edge computing is also addressed as part of T5.5 in WP5, in the context of 'Onboard networking capabilities' and later documented in deliverable D5.5 due at M30 of the project.

7 CONCLUSIONS AND NEXT STEPS

This concluding section provides the main findings, draws the conclusion and suggests a way forward in the next iteration of this deliverable.

NON-RT RIC

The non-RT RIC is a key component in the O-RAN architecture designed to RAN operations with control loops that exceed one second. This controller supports the development and deployment of AI/ML models that are essential for optimizing RAN functions over longer timescales. It is integrated into the SMO framework of the O-RAN architecture and communicates with other network elements via the A1, O1, and O2 interfaces.

The non-RT RIC plays a significant role in data management, policy definition, and machine learning model training and validation. It can collect extensive data from RAN elements, enabling advanced analytics to refine network performance and optimize resource utilization. The primary responsibilities of the non-RT RIC include defining and distributing policies that guide RAN behaviour and continuously refining these policies based on long-term data analysis. It also involves the collection and analysis of vast amounts of RAN data to train ML models that can predict and adapt to network conditions, identifying performance bottlenecks, and providing strategic planning for load balancing and resource management based on long-term usage patterns.

In addition to these core functions, the non-RT RIC addresses several specific challenges and explores various AI techniques to optimize network performance. One significant problem is traffic offloading, where AI-based techniques determine when to offload traffic to NTNs during low traffic periods, such as night-time in rural areas. This approach can lead to considerable energy savings by switching off some terrestrial network cells or carriers during low load times. Another critical issue is bandwidth allocation, where AI techniques pre-allocate PRBs on a large timescale based on traffic demand forecasts, thus ensuring efficient use of satellite resources and meeting user demands efficiently. These aspects will be further explored in Deliverable 4.7.

Load/traffic prediction is another essential aspect, where accurate forecasts ensure optimal radio resource allocation. AI models are employed to predict traffic loads, enabling dynamic adjustments to network resources to handle varying traffic patterns effectively. Furthermore, the non-RT RIC role in performance optimization involves analysing historical data and trends to identify bottlenecks and suggest improvements, refining resource allocation strategies to enhance overall network performance.

To tackle these challenges, various AI techniques are applied. In this document, techniques like KANs and MLP for time series forecasting or for load prediction exemplify how AI can be leveraged to improve network operations.

The next steps for non-RT RIC development should focus on addressing additional potential problems and exploring other AI techniques to further enhance network optimization. Developing AI models is useful to enable proactive radio resource and enhancing network performance and preventing service disruptions.

Probabilistic forecasting AI techniques offer a powerful approach for predicting various aspects of network performance in the context of non-RT RIC. Unlike traditional forecasting methods that provide a single-point estimate, probabilistic forecasting generates a range of possible outcomes with associated probabilities, enabling more robust and reliable decision-making. Probabilistic forecasting techniques based on AI can be employed to capture the inherent

uncertainties in network conditions and traffic patterns. These methods allow for more accurate prediction of load variations, potential bottlenecks, and resource demands, thereby enhancing the ability of the non-RT RIC to optimize resource allocation, plan for future capacity needs, and mitigate potential issues before they impact network performance. By incorporating probabilistic forecasts, the non-RT RIC can better manage long-term network planning and operational strategies.

By focusing on these potential problems and exploring these advanced AI techniques, the non-RT RIC can continue to evolve and significantly enhance the performance and reliability of 5G networks.

NEAR-RT RIC

This deliverable presents an overview of the O-RAN-standardized near-RT RIC architecture, together with the State of the Art of the AI-based RRM algorithms that can be implemented in it. The State of the Art includes a list of approaches, i.e., AI techniques and algorithms, that have been proposed in the literature to tackle specific RRM functions in the near-R RIC, focusing on TNs. Indeed, the application of AI algorithms for RRM in the near-RT RIC in NTN systems has not been investigated yet in the literature. As a first step, this deliverable presents the user scheduling RRM task in the context of user-centric beamforming in NTN from the point of view of DRL. The next iteration of this Deliverable, from the near-RT RIC side, will assess the performance of AI for RRM in NTN, considering the information presented in this Deliverable on user scheduling for user-centric beamforming as a starting point. For the evaluation, which will be carried out by means of simulation tools, the usage of datasets will be considered. In the context of user-centric beamforming in NTN, the system performance is typically dependent on the on-ground user spatial distribution [40]. For this reason, user position datasets may be used in this activity, e.g., part the Satellite Network Dataset provided by HSP in the context of the 5G-STARDUST project [3], or the public datasets listed in Section 5.4.2 of [41].

ONBOARD PROCESSING

A discussion about the design of processing capabilities has been initiated in this deliverable. The opportunities and constraints brought by such processing are highlighted, as well as other drivers. When it comes to digital OBP topic, the antenna beamforming strategy is among the first topic to consider, so at system level the question of Earth moving beams vs. Earth fixed beams has been the subject of a first trade-off analysis. Then, three incremental on-board architecture are considered, each requiring more demanding resources, but also offering more features. These architectures can each be associated with a certain time horizon (short, mid, long term) and the features provided for each case is discussed. The notion of SEC is considered in the last architecture where the full gNB is on-board with some UPF functions.

RAN SOFTWARIZATIOTN

This deliverable outlines the implementation effort for monitoring and controlling various functions within the 5G NTN using the ORAN E2 interface. It leverages the E2AP and E2SM to manage radio resources and optimize RAN performance. Additionally, it details the creation of the OFH under the 7.2x split, which enables quick and efficient integration with any RU. Finally, it provides an overview of the NTN extensions made to the 5G RAN. Future work will focus on expanding service models for specific NTN use cases, refining and advancing the NTN capabilities of the RAN, and performing any further extensions necessary to integrate the RAN solution into the PoC.

8 REFERENCES

[1] 5G-STARDUST D3.2, «End-to-end ground and space integrated architecture,» February 2024.

[2] 5G-STARDUST D3.1, «System Requirements Analysis and Specifications,» July 2023.

[3] 5G-STARDUST D4.1, «Open data sets for ML-based RRM,» January 2024.

[4] 5G-STARDUST D5.1, «D5.1: Open Data Sets for AI Data Driven Networking,» January 2024.

[5] 5G-STARDUST D6.1, «PoC functional architecture document,» March 2024.

[6] 5G-STARDUST D2.1, «Scenarios, use cases, and services,» July 2023.

[7] George EP Box and al., Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[8] R. J. Hyndman y G. Athanasopoulos, Forecasting: principles and practice, OTexts, 2018.

[9] Charles C Holt, «Forecasting seasonals and trends by exponentially weighted moving averages,» *Int. journal of forecasting,* vol. 20, nº 1, pp. 5-10, 2004.

[10] Peter R Winters, "Forecasting sales by exponentially weighted moving averages," *Management science,* vol. 6, no. 3, p. 324–342, 1960.

[11] G Peter Zhang et al., «Neural networks for time-series orecasting.,» *Handbook of natural computing,* vol. 1, p. 4, 2012.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, p. 1735–1780, 1997.

[13] Anastasia Borovykh and al., «Conditional time series forecasting with convolutional neural networks,» *arXiv preprint arXiv:1703.04691,* 2017.

[14] Ziming Liu and al., «Kan: Kolmogorov-arnold networks,» *arXiv preprint arXiv:2404.19756, 2024,* 2024.

[15] Andrei Nikolaevich Kolmogorov, "On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables," *American Mathematical Society,* 1961.

[16] J. Braun y M. Griebel, «On a constructive proof of Kolmogorov's superposition theorem,» *Constructive approximation,* vol. 30, p. 653–675, 2009.

[17] C. J. Vaca-Rubio, L. Blanco, R. Pereira y M. Caus, «Kolmogorov-Arnold Networks (KANs) for Time Series Analysis,» *arxiv preprint arxiv: 2405.08790,* 2024.

[18] 3GPP TS 38.306, «NR; User Equipment (UE) radio access capabilities (Release 18),» April 2024.

[19] 3GPP TS 38.211, «NR; Physical channels and modulation (Release 18),» March 2024.

[20] Q. e. a. Qiu, « ReLU-KAN: New Kolmogorov-Arnold Networks that Only Need Matrix Addition, Dot Multiplication, and ReLU,» *arXiv preprint arXiv:2406.02075,* 2024.

[21] O-RAN, "O-RAN architecture description," Alfter, Germany, 2021.

[22] O-RAN Working Group 3, «O-RAN Near-Real-time RAN Intelligent Controller E2 Service Model (E2SM) KPM 2.0,» de *ORAN-WG3.E2SMKPM*, 2021.

[23] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz y T. Tugcu, «Intelligent network data analytics function in 5G cellular networks using machine learning,» *Journal of Communications and Networks,* vol. 22, nº 3, pp. 269-280, June 2020.

[24] A. Giannopoulos et al., «Supporting Intelligence in Disaggregated Open Radio Access Networks: Architectural Principles, AI/ML Workflow, and Use Cases,» *IEEE Access,* vol. 10, pp. 39580-39595, 2022.

[25] S. Niknam et al., «Intelligent O-RAN for Beyond 5G and 6G Wireless Networks,» de *IEEE Globecom Workshops (GC Wkshps),*, 2022.

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

[27] R. Wiebusch, N. A. Wagner, D. Overbeck, F. Kurtz y C. Wietfeld, «Towards Open 6G: Experimental O-RAN Framework for Predictive Uplink Slicing,» de *ICC 2023 - IEEE International Conference on Communications*, 2023.

[28] H. Zhang, H. Zhou y M. Erol-Kantarci, «Federated Deep Reinforcement Learning for Resource Allocation in O-RAN Slicing,» de *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022.

[29] B. Agarwal, M. A. Togou, M. Ruffini y G. -M. Muntean, «QoE-Driven Optimization in 5G O-RAN-Enabled HetNets for Enhanced Video Service Quality,» *IEEE Communications Magazine,* vol. 61, nº 1, pp. 56-62, January 2023.

[30] H. Zhang, H. Zhou y M. Erol-Kantarci, «Team Learning-Based Resource Allocation for Open Radio Access Network (O-RAN),» de *ICC 2022 - IEEE International Conference on Communications*, 2022.

[31] M. Kouchaki y V. Marojevic, «Actor-Critic Network for O-RAN Resource Allocation: xApp Design, Deployment, and Analysis,» de *IEEE Globecom Workshops (GC Wkshps)*, 2022.

[32] A. Lacava et al., «Programmable and Customized Intelligence for Traffic Steering in 5G Networks Using Open RAN Architectures,» *IEEE Transactions on Mobile Computing,* vol. 23, nº 4, pp. 2882-2897, April 2024.

[33] 5G-STARDUST, «D4.3: Report on the Unified Air Interface, User-Centric and Beamforming Solutions,» April 2024.

[34] B. D. Filippo, R. Campana, A. Guidotti, C. Amatetti y A. Vanelli-Coralli, «Cell-Free MIMO in 6G NTN with AI-predicted CSI,» *submitted B. D. Filippo, R. Campana, A. Guidotti, C. Amatetti y A. Vanelli-Coralli, «Cell-Free MIMO in 6G NTN with AI-predicted CSI,» submitted to IEEE 25th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) ,* April 2024.

[35] Y. Zhang, Y. Wu, A. Liu, X. Xia, T. Pan y X. Liu, «Deep Learning-Based Channel Prediction for LEO Satellite Massive MIMO Communication System,» *IEEE Wireless Communications Letters,* vol. 10, nº 8, pp. 1835-1839, Aug. 2021.

[36] B. Ahmad, D. G. Riviello, A. Guidotti y A. Vanelli-Coralli, «Graph-Based User Scheduling Algorithms for LEO-MIMO Non-Terrestrial Networks,» de *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2023.

[37] e. a. V. Mnih, «Playing atari with deep reinforcement learning,» de *arXiv preprint, arXiv:1312.5602*, 2013.

[38] R. C. L. Liu y G. Yang, «Traffic Demand Matching-based Dynamic Resource Allocation Algorithm for Multi-Beam Satellite Systems,» de *IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023.

[39] 5G-STARDUST D4.2, «Development and Test Plan,» March 2024.

[40] B. D. Filippo, B. Ahmad, D. G. Riviello, A. Guidotti y A. Vanelli-Coralli, «Non-Uniform User Distribution in Non-Terrestrial Networks with Application to User Scheduling,» de *IEEE International Mediterranean Conference on Communications and Networking*, 2024.

[41] 5G-STARDUST D4.3, «Report on the Unified Air Interface, User-Centric and Beamforming Solutions,» April 2024.