D.4.1 Experimentation facility testing as a service and explainable resource management framework design and initial prototyping

Document Summary Information

Project Identifier	HORIZON-JU-SNS-2022. Project 101096954		
Project name	6G-BRICKS: Building Reusable testbed Infrastructures for validating Cloud-to-device breaKthrough technologieS		
Acronym	6G-BRICKS		
Start Date	January 1, 2023	**End Date**	December 31, 2025
Project URL	www.6g-bricks.eu		
Deliverable	D4.1		
Work Package	WP4		
Contractual due date	31/03/2024	**Actual submission date**	02/05/2024
Type	Report	**Dissemination Level**	Public
Lead Beneficiary	ISRD		
Responsible Author	Md Arifur Rahman		
Contributors	Panagiotis Marantis (ISI), Ashima Chawla (LMI), Anne-Marie Bosneag (LMI), Abdelkader Mekrache (EUR), Karim Boutiba (EUR), Adlen Ksentini (EUR), Adam Girycki (ISRD), Michalis Rantopoulos (OTE), Anastasios Bikos (IQU), Kostas Ramantas (IQU) Konstantinos Togias (ISI)		
Peer reviewer(s)	Carolina Fernández (i2CAT), Michalis Rantopoulos (OTE), Anne-Marie Bosneag (LMI)		

Co-funded by the European Union

6G-BRICKS 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 Grand Agreement No 101096954.

Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete	Changes	Contributor(s)
V0.1	20/12/2023	2%	Providing Initial Table of Contents (ToC)	Md Arifur Rahman (ISRD)
V0.2	31/01/2024	30%	Contribution in Section 3 and Section 4	Contributor list from (IQU, ISI, LMI, ICOM)
V0.3	13/02/2024	40%	Contribution in Section 4	Contributor list from (ISI, LMI, OTE)
V0.4	14/02/2024	70%	Contribution in Section 3	Anastasios Bikos (IQU)
V0.5	20/02/2024	80%	Contribution in Section 5	Adam Girycki (ISRD)
V0.6	22/02/2024	98%	Contribution in Sections 3.1 & 3.2(minor revisions), editorial contribution related Section 1	Anastasios Bikos (IQU) Md Arifur Rahman (ISRD)

Disclaimer

The content of this document reflects only the author's view. Neither the European Commission nor the SNS-JU are responsible for any use that may be made of the information it contains.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the 6G-BRICKS consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the 6G-BRICKS Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the 6G-BRICKS Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© 6G-BRICKS Consortium. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgment of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorized provided the source is acknowledged.

Table of Contents

1. Executive Summary 10
2. Introduction 12
2.1 Mapping 6G-BRICKS Outputs 12
2.2 Deliverable Overview and Report Structure 14
3. Experimentation Framework for vertical Testing as a Service and Intent-driven Network Management 15
3.1 6G-BRICKS Portal design 15
3.1.1 6G-BRICKS Portal technology stack 15
3.1.2 Frontend and Backend Modules 16
3.1.3 Portal Wireframe Diagrams 17
3.2 Conflict resolution algorithms for the Intent Management Engine 20
3.2.1 6G-BRICKS Intent-Based Networking workflows 21
3.2.2 Conflict Detection and Policy Resolution Strategies overview 22
3.2.3 6G-BRICKS intent conflict resolution and consensus convergence strategy 26
3.3 Optimization algorithms for intent-driven cloud continuum 29
4. Explainable AI for DMO and Anomaly Detection 34
4.1 Introduction 34
4.2 System Monitoring and Data Collection 34
4.3 AI-driven Fault Management overview 35
4.3.1 Classic Machine Learning 35
4.3.2 Neural network 35
4.4 Explainable AI algorithms for DMO overview 38
4.4.1 Introduction 38
4.4.2 XAI Integration into 6G and Post-hoc Interpretation 38
4.4.3 XAI overview for networking Data 41
4.5 6G-BRICKS Models for anomaly detection and XAI integration 43
4.5.1 Autoencoder with SHAP Integration For 6G-BRICKS 43
4.5.2 Generative model for Anomaly Detection with reconstruction error based Visualised Explanations 47
4.5.3 Combining NWDAF with AI for Abnormal Traffic Detection 50
4.5.3.1 NWDAF Overview 50
4.5.3.2 NWDAF AE-based Abnormal traffic detector 52
4.5.4 LLM-enabled Vertical Trial Fault Root Cause Analysis using XAI 52
4.5.5 System design 53
5. Near-RT RIC Framework for CF mMIMO Radio Resource Management via O-RAN compliant xApps 55
5.1. Near-RT RIC Framework and Signaling Mechanisms of E2 Nodes 55
5.1.1 E2 Indication procedure 56
5.1.2 E2 control procedure 56
5.2. The RIC xApp messaging infrastructure of trajectory prediction in Cell-free mMIMO 57
5.3. The RIC xApp messaging infrastructure of multi-band clustering in Cell-free mMIMO 58
6. Near-RT RIC Framework for Reflected Intelligent Surface (RIS) control via xApps 60
6.1 RIS control framework Exposure innovation 60
6.2 xApp Monitoring and KPM Service Model related to RIS 61
6.3 RIS E2SM 61
7. Conclusions 63
8. References 64

© 6G-BRICKS 2023

List of Figures

Figure 1: The 6G-BRICKS Experimenters Portal high-level architecture. 16
Figure 2: Portal Login Page 17
Figure 3: List of experiment and option to create a new one. 18
Figure 4: Experiment creation page. 19
Figure 5: The xApp list page. 20
Figure 6: Intent Management in 6G-BRICKS. 21
Figure 7: High-level diagram of the intent-reports plane [2]. 22
Figure 8: Bin packing approaches [6] and [20]. 24
Figure 9: 6G-BRICKS intent conflict resolution engine. 27
Figure 10: SAC admission control simulation results. 28
Figure 11: Heatmaps of the 5G/6G network monitoring metrics self-correlation. 29
Figure 12: High-level diagram of the optimization algorithm. 30
Figure 13: Flowchart of the XAD framework. 34
Figure 14: Time series window [29]. 36
Figure 15: Structure of an AE. 36
Figure 16: Structure of the VAE. 37
Figure 17: XAI integration framework in future wireless networks [37]. 40
Figure 18: Comparison of the most important features using different methods. 43
Figure 19: The TELCO dataset. 44
Figure 20: The Autoencoder-based model. 44
Figure 21: Error distribution plots for training and testing sample distribution. 45
Figure 22: Anomalous instances interpretation using SHAP. 46
Figure 23: The Network Data Analytic Function (NWDAF) [58]. 51
Figure 24: Internal NWDAF architecture [59]. 51
Figure 25: LSTM Autoencoder architecture. 52
Figure 26: Trial explainability overview. 53
Figure 27: The framework architecture. 54
Figure 28: The Near-RT xApp API. 55
Figure 29: E2 INDICATION message. 56
Figure 30: The E2 CONTROL message. 57
Figure 31: The xApp control loop. 57
Figure 32: xApp input received in E2 INDICATION. 58

© 6G-BRICKS 2023

Figure 33: xApp inputs received in E2 INDICATION supporting multi-band clustering. 59
Figure 34: RIS platform exposure via 6G-BRICKS platform..... 60
Figure 35: RIS control flow via the E2 interface. 62

List of Tables

Table 1: Adherence to 6G-BRICKS GA Deliverable & Tasks Descriptions 12

Glossary of terms and abbreviations used

Abbreviation / Term	Description
ABP	Alternating Back Propagation
AD	Anomaly Detection
ADMM	Alternating Direction Method of Multipliers
AE	Autoencoder
AI	Artificial Intelligence
CBLOF	Cluster-based Local Outlier Factor
CCCM	Cognitive Compute Continuum Manager
CF mMIMO	Cell-free massive Multiple-Input-Multiple-Output
CN	Core Network
CNN	Convolutional Neural Network
CR	Conflict Reconciliation
CSI	Coded Stochastic Incremental
CSS3	Cascading Style Sheets
DGD	Decentralized Gradient Descent
DL	Deep Learning
DMO	Domain Management Orchestrator
DNN	Deep Neural Network
DQN	Deep Q-Network
DwL	Downlink
ECN	Edge Computing Node
eMBB	Enhanced Mobile Broad Band
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GD	Gradient Descent

© 6G-BRICKS 2023

HTML5	HyperText Markup Language
IAM	Identity and Access Management
IBN	Intent-Based Networking
IF	Isolation Forest
ITU	International Telecommunications Union
KPI	Key Performance Indicator
KPM	Key Performance Measurement
LCM	Learnable Commutative Monoids
LIME	Local Interpretable Model-Agnostic Explanations
LLM	Large Language Model
LRP	Layer-wise Relevance Propagation
MAE	Mean Absolute Error
MANO	Management and Orchestrator
MCMC	Markov Chain Monte Carlo
MCOD	Micro-Cluster based Outlier Detection
ML	Machine Learning
MNO	Mobile Network Operator
MSE	Mean Squared Error
MTS	Multi-variate Time Series
Near-RT	Near Real-Time
NLP	Natural Language Processing
NSD	Network Service Descriptor
O-CU	O-RAN Centralized Unit
O-DU	O-RAN Distributed Unit
OAI	Open Air Interface
OPEX	Operational Expenditure
O-RAN	Open Radio Access Network
PaaS	Platform as a Service

© 6G-BRICKS 2023

PNN	Predictive Neural Network
PoC	Proof of Concept
RAM	Random Access Memory
RAN	Radio Access Network
RDBMS	Relational Database Management System
RIC	RAN Intelligent Controller
RIS	Reflected Intelligent Surface
RRM	Radio Resource Management
RNN	Recurrent Neural Network
RSRP	Received Signal Strength Power
RU	Radio Unit
SAC	Soft Actor-Critic
SBI	Southbound Interface
SDN	Software-Defined Networks
SHAP	SHapley Additive exPlanation
SLA	Service Level Agreement
SLO	Service Level Objectives
SM	Service Model
SoTA	State-of-the-Art
SSO	Single Sign-On
TMN	Telecommunications Management Network
TP	True Positive
UC	Use Case
UE	User Equipment
UL	Uplink
UPF	User Plane Function
URLLC	Ultra-Reliable and Low Latency Communications

© 6G-BRICKS 2023

VAE	Variational Autoencoder
WSN	Wireless Sensor Network
XAD	Explainable Anomaly Detection
XAI	Explainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting
ZSM	Zero-Touch Service Management
6G	6th Generation Networks
5G	5th Generation Networks

© 6G-BRICKS 2023

1. Executive Summary

This deliverable is part of WP4 “Experimentation facility management, domain and security orchestrator” and supports the work under tasks T4.1 “Experimentation Framework for vertical Testing as a Service and Zero-Touch Policy engine”, T4.2 “Explainable AI for Domain resource Management and Orchestration” and T4.3 “Near-RT RIC framework for Radio Resource Management via Open-Radio Access Network (O-RAN) compliant eXtended application (xApps). It covers the implementation and experimental facility tools that can enable the experimenters to access the 6th Generation Networks (6G) sites to deploy and test their solutions. In this regard, the Intent Management Engine encompasses the intents related to service onboarding at the 6G-BRICKS facility by considering autonomously driven decisions and actuations that fulfill requirements, goals and constraints expressed by intent. Several SoTA algorithms for eXplainable AI (XAI) for domain management and orchestration are discussed in this deliverable, along with proposed beyond SoTA solutions and key innovations. The combination of NWDAF with ML for abnormal traffic detection is explored to design the UE abnormal behavior on the network. The LLM-based vertical trail fault root cause analysis combines the results of different AI and XAI models, which are described as well in this deliverable. Moreover, the deliverable introduces Near-RT Radio Intelligent Controller (Near-RT RIC) framework for Radio Resource Management (RRM) of E2 nodes, Near-RT RIC xApp messaging infrastructure, and the Service Models (SMs) related to the trajectory prediction and multi-band clustering xApps that solve resource scheduling, clustering, and precoding of cell-free mMIMO networks.

We divide this deliverable into three technical sections: (1) the first one represents the contributions related to experimentation framework for vertical testing as a service and intent-driven network management; (2) the second section represents explainable AI for Domain Manager Orchestrator (DMO), Orchestration and anomaly detection, and (3) the last one describes the Near-RT RIC Framework for RRM via O-RAN complaint xApps related to key innovation of Cell-free massive Multiple-Input-Multiple-Output (CF mMIMO) and Reflected Intelligent Surface (RIS).

The first technical section, section 3 of the deliverable, describes the experimentation framework and provides simplified approaches of onboarding mechanisms of the experimenters’ applications in the platform. It also provides a brief description of the (interface) wireframes for the experimenters and the workflow of the wireframes. Moreover, this section also provides the intent management engine design methodologies and an example of SoTA conflict resolution algorithms that is further used to validate the proposed conflict resolution algorithms. Regarding conflict resolution, this section also provides a design of an admission control algorithm by leveraging the short and long-term correlation of resource monitoring traffic metrics in the DMO to avoid conflicts and maximize the admission control efficiency. For the intent-driven cloud continuum, an optimization algorithm procedure is briefly discussed that shows the procedures of assigning a request to the system by considering several attributes of the systems.

Section 4, the second technical section, showcases the innovative aspects of explainable AI for DMO, orchestrator, and anomaly detection and highlights the XAI algorithms that are devised to support complex and popular ML algorithms used in networks. The eXplainable Anomaly Detection (XAD) framework is presented in this section, which uses the KPI measurements from the networks, processes the data and creates the anomaly detection model. The reconstruction-based anomaly detection model is used to classify the normal and anomalies in the data. Once the anomalies are detected, the explainable module interprets the reconstruction error of the anomalous samples and the feature importance explanation (e.g., SHAP) visualizes explanations and mitigates the actions. This section highlights several AI-driven SoTA algorithms for fault management, explainable AI, and anomaly detection. Based on the current SoTA solutions, it provides a novel XAD framework based on combining fault management techniques with post-hoc explainability techniques (e.g., SHAP). The proposed solution is tested with a complex dataset extracted during the operation of the DMO of 6G-BRICKS. The combination of NWDAF with AI for anomaly detection is briefly discussed in this section, which leverages the AI approaches to detect abnormal traffic.

© 6G-BRICKS 2023

The third technical section, Section 5 of the deliverable mainly highlights the Near-RT RIC framework and signaling mechanisms of the underlying Radio Access Network (RAN) nodes (e.g., the O-RAN Distributed Unit; or O-DU, and the O-RAN Centralized Unit, or C-DU). This section briefly introduces how the optimization algorithms that are running at the Near-RT RIC as an xApp make use of the APIs to collect the data, run the algorithms and forward the control decision to the underlying RAN nodes. The Near-RT RIC messaging for 6G-BRICKS xApps (i.e., the trajectory prediction and multi-band clustering) are briefly described based on mainly focusing on the message format structure of “E2 INDICATION message” and the “E2 CONTROL message” based on collecting the Key Performance Measurement (KPM) from the underlying RAN and trigger the CONTROL decision over the E2 interface to control the parameter of E2 nodes that enhance the performance of CF mMIMO networks. Similarly, this section also highlights the messaging structure of RIS control xApp and related KPM to be used to enable the control functionalities of the RIS through the E2SM RIS proxy.

© 6G-BRICKS 2023

2. Introduction

6G-BRICKS aims to provide 6G experimental facilities for experiments in the KUL platform that offer enablers related to CF mMIMO and the EUR platform which offers enablers related to RIS. The Facility is accessible through the experimental framework and the experimenters can conduct their individual test on the platform through the ISI portal. This deliverable is an initial report that details the low-level implementation details of the 6G-BRICKS enablers covering the above-mentioned innovations, that is, the RIS control and the compute continuum. Moreover, this deliverable introduces the design guideline experimentation facility and provides the explainable resource management framework design and initial prototyping, which will describe how the 6G-BRICKS enablers can be tested in the platform along with innovation related to intent management engine and explainable AI for DMO.

This deliverable investigates the initial approaches of 6G-BRICKS by providing details on:

- The design of the 6G-BRICKS experimental portal
- The design of the Intent Management Engine and the conflict resolution algorithms
- The 6G-BRICKS methodologies for intent conflict resolution
- Optimization algorithms for intent-driven cloud continuum
- Explainable AI for DMO and Anomaly Detection
- Models for anomaly detection and XAI integration
- Combination of NWDAF with AI for Abnormal Traffic Detection
- Near-RT RIC Framework for RRM via O-RAN-compliant xApps related to CF mMIMO
- Framework for the integration of RIS with Near-RT RIC

2.1 Mapping 6G-BRICKS Outputs

The purpose of this section is to map 6G-BRICKS Grant Agreement commitments, both within the formal Deliverable and Task description, against the project's respective outputs and work performed.

Table 1: Adherence to 6G-BRICKS GA Deliverable & Tasks Descriptions

6G-BRICKS GA Component Title	6G-BRICKS GA Component Outline	Respective Document Chapter(s)	Justification
DELIVERABLE			
This D.4.1 provides a public report on experimentation facility testing as a service and explainable resource management frameworks design and initial prototyping.			
Task 4.1 Experimentation Framework for vertical Testing as a Service and Zero-Touch Policy engine	1. Deliver enhancements of the 5GMediaHUB Experimentation Tools that improve on the scalability and extensibility of the ISI facility to support Use Cases (UCs) and Open Calls participants 2. Interfacing of the Experimentation Tools with the innovative 6G-BRICKS	Section 3	The first part of the deliverable presents the design of the 6G-BRICKS experimental portal in Section 3.1. We added an example of the wireframes of the 6G-BRICKS Experimenters' Portal frontend, where experimenters can log in to the facilities for conducting the experiments. We also provide the design of the

	decentralized management plane and a new Policy Engine		Intent Management Engine and conflict resolution algorithms in Section 3.2, used to detect conflicts and provide policy resolution strategies.
Task 4.2 Explainable AI for Domain resource management and orchestration (DMO)	1. XAI and MR modules that will run inside the XAI-DE of the DMO answering why specified intents and policies are specified 2. New XAI algorithms will be devised and developed to support complex and popular ML algorithms used in network and service management (Recurrent Neural Network and Long Short-Term Memory etc.) 3. Knowledge models will be defined based on intent and policies and MR algorithms to allow the analysis of root cause problems and derive solutions as well as detect and resolve conflicts	Section 4	The second part of the deliverable showcases the innovation aspects of XAI for DMO, highlighting the XAI algorithms devised to support complex and popular ML algorithms that are currently used in the network. It also provides a model for anomaly detection based on an autoencoder with SHAP integration by using a Telco dataset available in 6G-BRICKS. Moreover, a generative model for anomaly detection with reconstruction error-based explanations is briefly described, along with detailed steps available as explainability modules. Lastly, this section concludes with a contribution related to the combination of NWDAF with AI for abnormal traffic detection.
Task 4.3 Near-RT RIC framework for Radio Resource Management via O-RAN compliant xApps	1. Design and develop the Near-RT RIC framework for RRM of E2 nodes along with an xApp subscription manager 2. Develop the RAN DB for the storage of configurations and KPIs, and the support of logging, tracing, and metrics collection. 3. The RIC xApp messaging infrastructure will be designed with a latency tolerance of 10-500ms, for monitoring, managing, and controlling the underlying nodes. 4.The experimenter xApps will communicate with the RAN nodes over the 3-SMs (i.e., E2SM-KPM,	Section 5.1 Section 5.2 Section 5.3	This technical section describes the Near-RT RIC framework and the signaling mechanisms towards the underlying E2 nodes. Especially, the optimization algorithms, e.g., trajectory prediction and multi-band clustering xApps messaging structure of “E2 INDICATION” and “E2 CONTROL” that collect KPM from E2 nodes and deliver control decisions to change the parameter that can enhance the performance metrics of the RAN.

	E2SM-NI, and E2SM-RC) and the SMs will be further modified based on the requirement of the experimenters xApps		

2.2 Deliverable Overview and Report Structure

Section 3 investigates the experimentation framework for vertical Testing as a Service and intent-driven Network Management, while section 4 addresses explainable AI for DMO and anomaly detection. Subsection 3.1 summarizes the 6G-BRICKS portal design aspects for the experimenters. Subsection 3.2 describes the intent management engine and conflict resolution algorithms. In section 4, subsection 4.1 introduces the DMO, subsection 4.2 describes system monitoring and data collection, subsection 4.3 AI-driven fault management SoTA, subsection 4.4 provides the explainable AI algorithm for DMO, and subsection 4.5 provides models for anomaly detection and XAI integration framework for 6G-BRICKS. On the other hand, section 5 describes the Near-RT RIC framework for RRM via O-RAN compliant xApp and section 6 presents the framework for RIS Integration with Near-RT RIC. Finally, section 7 concludes the deliverable.

© 6G-BRICKS 2023

3. Experimentation Framework for vertical Testing as a Service and Intent-driven Network Management

Intent-Based Networking is a critical 6G enabler that helps automate network components and computational infrastructures. In this domain, 6G-BRICKS facilitates **intent-based** automation powered by Artificial Intelligence (AI), which will be able to ensure a high level of security, efficiency, scalability, and sustainability; particularly in cross-domain and interoperable deployment environments, i.e., where Points of Presence (PoPs) are identically distributed and non-independent (non-IID). This is facilitated by the 6G-BRICKS experimentation plane, which is introduced in this section and includes an experimentation Portal service and a novel intent management engine platform. The latter offers intent Conflict Reconciliation (CR) on closed loops from the resource layer to the business layer, possessing fully end-to-end decentralized AI and XAI features, alongside ultra-fast converging Game-Theoretic actors.

3.1 6G-BRICKS Portal design

The 6G-BRICKS Portal will support the 6G-BRICKS and third-party UCs, simplifying the onboarding of applications coming from open calls and offering a private Docker Image service repository. Moreover, the portal allows experimenters to onboard xApps, providing access to experimentation at the RAN domain (e.g., changing parameters related to the RIS). The Experimenter's portal technology stack will be based on an Angular frontend and a Python backend, leveraging the Flask micro web framework. In what follows, further details about the Portal technology stack and baseline technologies are contributed.

3.1.1 6G-BRICKS Portal technology stack

To realize the Experimenters Portal GUI, we will follow an agile interaction design to achieve practical and pragmatic design, short development cycles, and dynamic designs that are responsive to changing needs [1]. The technology that will be used for the development of the frontend and its integration to the backend components will be based on open-source standards and deployed to the cloud as a Docker image[1]. The portal will be built using a combination of programming languages based on the programming Framework, which can be separated into two categories, the client-side languages, and the server-side ones.

Client-Side: the code that sits here is executed on web browsers. Due to the low processing power available on the client side, this code is intended to retrieve the information from the backend system and formalize the data into a user-friendly display manner. Technologies widely used in this case are primarily HTML5 (HyperText Markup Language), CSS3 (Cascading Style Sheets) and JavaScript (Angular JS). Angular[2] is a powerful and widely used open-source framework for building dynamic web applications. Developed and maintained by Google, it allows developers to create rich, interactive client-side applications by utilizing HTML, CSS, and TypeScript—a superset of JavaScript. Angular's architecture is designed around components and services, making it highly modular and allowing for reusable code. Its data binding and dependency injection features significantly reduce the amount of code developers need to write, leading to faster development cycles and robust applications. Angular is an excellent choice for developing the 6G-BRICKS Portal GUI, as it supports agile interaction design principles and facilitates the creation of responsive, user-friendly interfaces.

Server-Side: this code is executed on the server side. Tasks such as 3rd party integrations, database integrations, data manipulation and generally tasks that require large processing power are run in the backend. Within the context of the project, the server-side coding is done in Python Flask[3]. F**lask** is a lightweight and flexible web framework for Python that allows developers to build web applications quickly and easily. It is designed to be

[1] Docker https://www.docker.com/
[2] https://angularjs.org/
[3] https://flask.palletsprojects.com/en/3.0.x/

© 6G-BRICKS 2023

simple and easy to use, yet powerful enough to handle complex web applications. Flask is based on the WSGI (Web Server Gateway0 Interface) specification and provides a simple API for developers to create web applications. It supports routing, request handling, templating, and more. Flask is also highly extensible, with a large number of third-party extensions available to add functionality such as database integration, authentication, and more. Flask is often used for building small to medium-sized web applications or APIs. Its simplicity and flexibility make it a popular choice for prototyping, Proof of Concept (PoC) development, and small-scale production deployments.

3.1.2 Frontend and Backend Modules

Figure 1 shows the high-level architecture of the Experimenters Portal frontend and backend, and their constituent components.

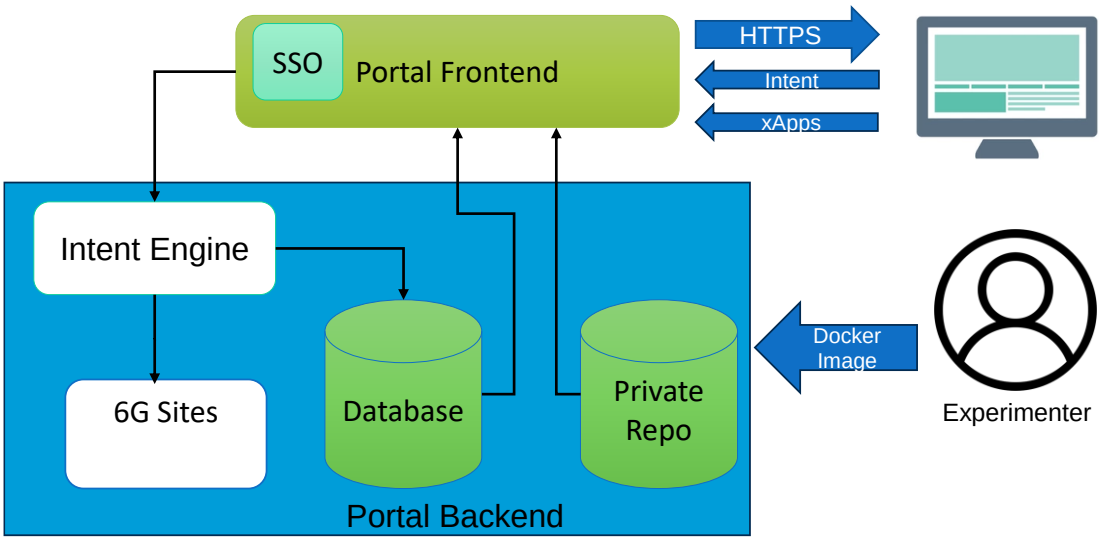

Figure 1: The 6G-BRICKS Experimenters Portal high-level architecture.

The 6G-BRICKS portal will deliver a fully customised frontend, based on Angular. In a nutshell, the portal frontend components act as an entry point to external experimenters, facilitating the execution of tests, as well as the user registration and login workflows. For the latter, the Frontend module leverages an off-the-shelf Single Sign On (SSO) module, which implements the **Identity and Access Management (IAM)**. IAM systems enable the management of user identities, authentication, and authorization, providing a secure mechanism to verify users and control their access to various parts of the application based on their roles and privileges. 6G-BRICKS will leverage Keycloak as an open-source IAM solution aimed at modern applications and services. It offers comprehensive support for OAuth2 and OpenID Connect protocols, making it a robust choice for secure authentication and authorization. Keycloak simplifies the implementation of Single Sign-On (SSO) across multiple applications, providing a seamless user experience and robust security features. With its user-friendly management interface and flexible API, Keycloak allows for easy integration with a wide range of applications, making it an ideal choice for managing identities in the 6G-BRICKS Portal. The offered SSO capabilities significantly decrease inconvenience for the platform developers and users, allowing seamless navigation between 6G-BRICKS services without the requirement of dealing with multiple account systems and authorization requirements.

Moreover, the Portal delivers a set of Backend modules, that closely cooperate with the Frontend modules, delivering server-side functionality. More specifically:

- **Database**: MySQL is an open-source Relational Database Management System (RDBMS) that is widely used for web applications. It is a popular choice for managing and storing data in many types of applications, from small-scale websites to large-scale enterprise systems. MySQL is known for its performance, scalability, and reliability, and it is used by many of the world's largest and most popular websites and applications.
- **Private Docker repository**: A JFrog private Helm repository is used to store chart packages, providing methods for uploading, retrieving, updating and deleting Chart packages via Client URL (cURL) commands. It offers privacy, access control, high availability and massively scalable storage providing advanced, enterprise-ready repository management for charts. An OpenAPI server is used as a proxy for facilitating the interaction of the platform front-end components (Portal) with the Private Helm Repo and supporting these basic functions.

3.1.3 Portal Wireframe Diagrams

The constructed wireframe diagrams of the 6G-BRICKS Experimenters' Portal frontend are illustrated below. These are visual diagrams that outline the skeletal framework of the Portal. They show how the frontend elements relate to each other and how they are structured, without focusing on the visual design or content. These will serve as the blueprint for the Portal Frontend implementation. The following wireframes in Figure 2, Figure 3, Figure 4, and Figure 5 showcase wireframes for the main Portal workflows; i.e., user login & registration, show running experiments, create a new vertical application experiment and create a new xApp experiment.

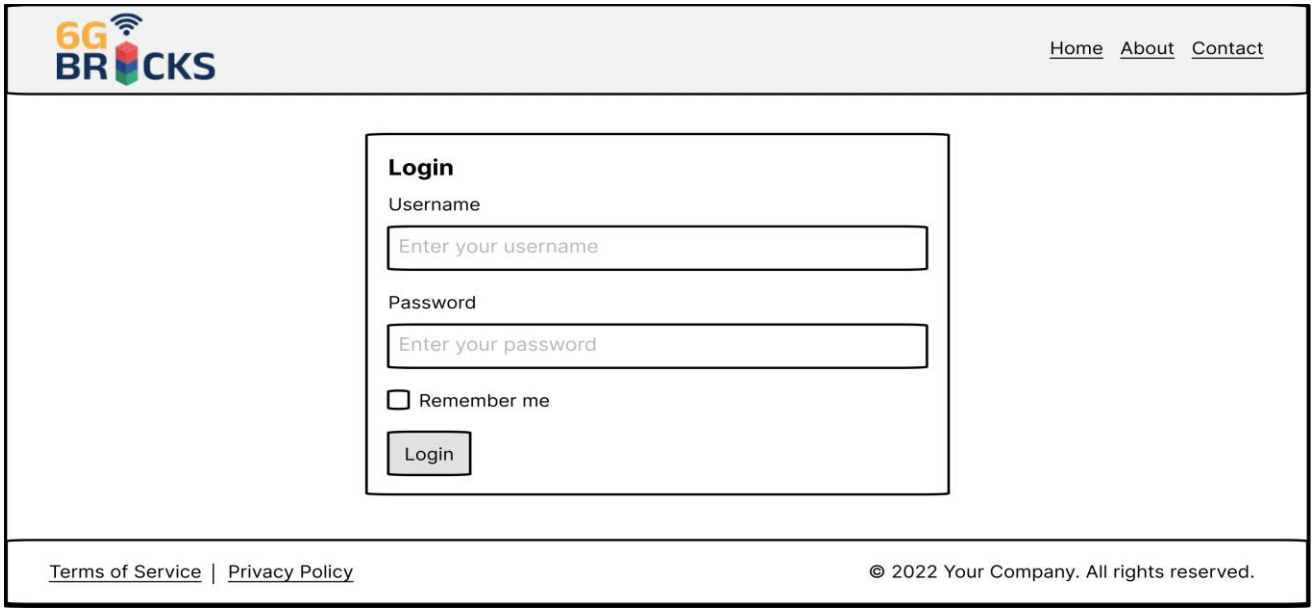

Figure 2: Portal Login Page

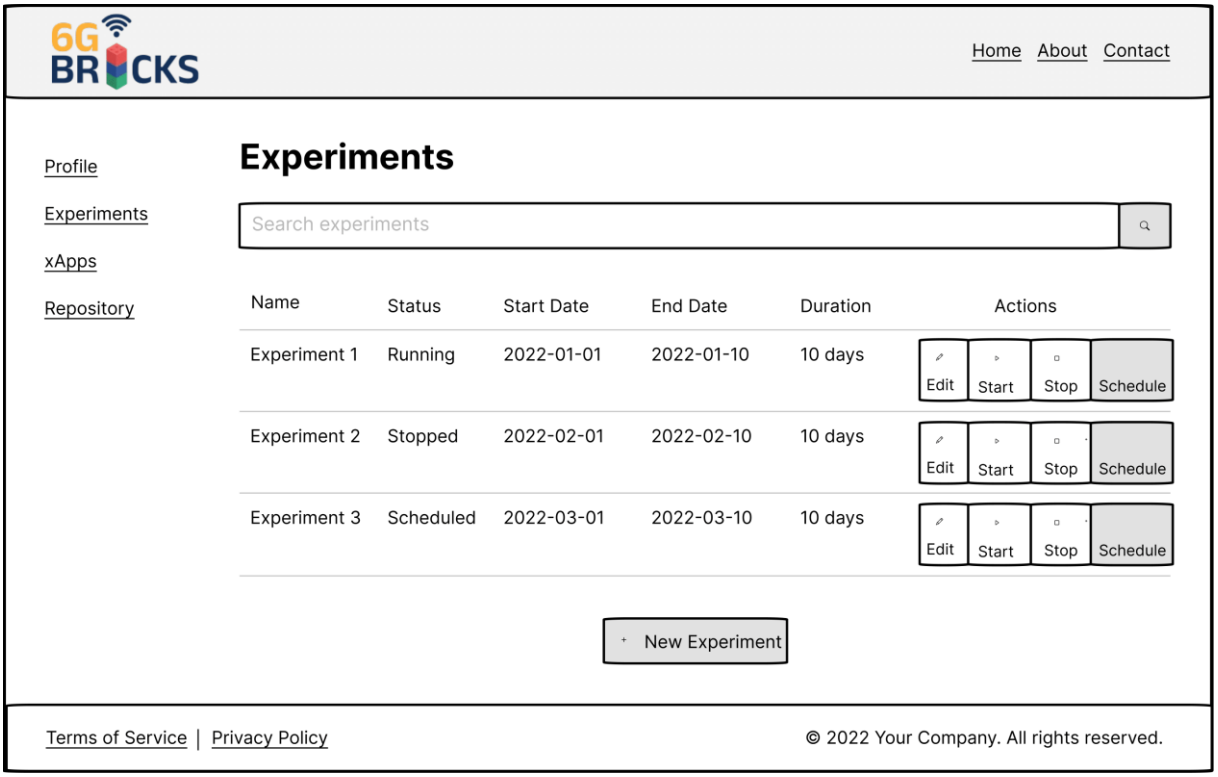

Figure 3: List of experiment and option to create a new one.

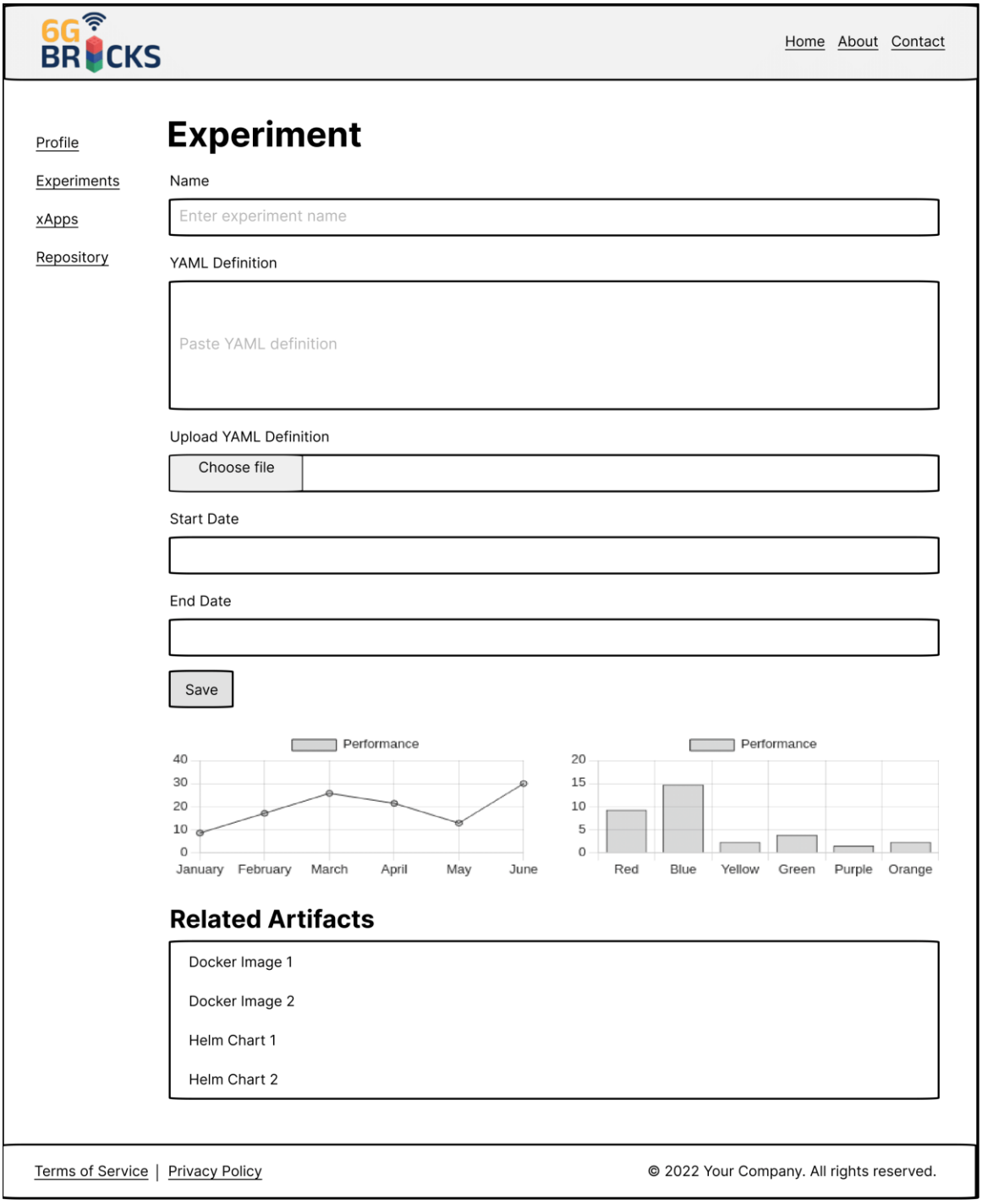

Figure 4: Experiment creation page.

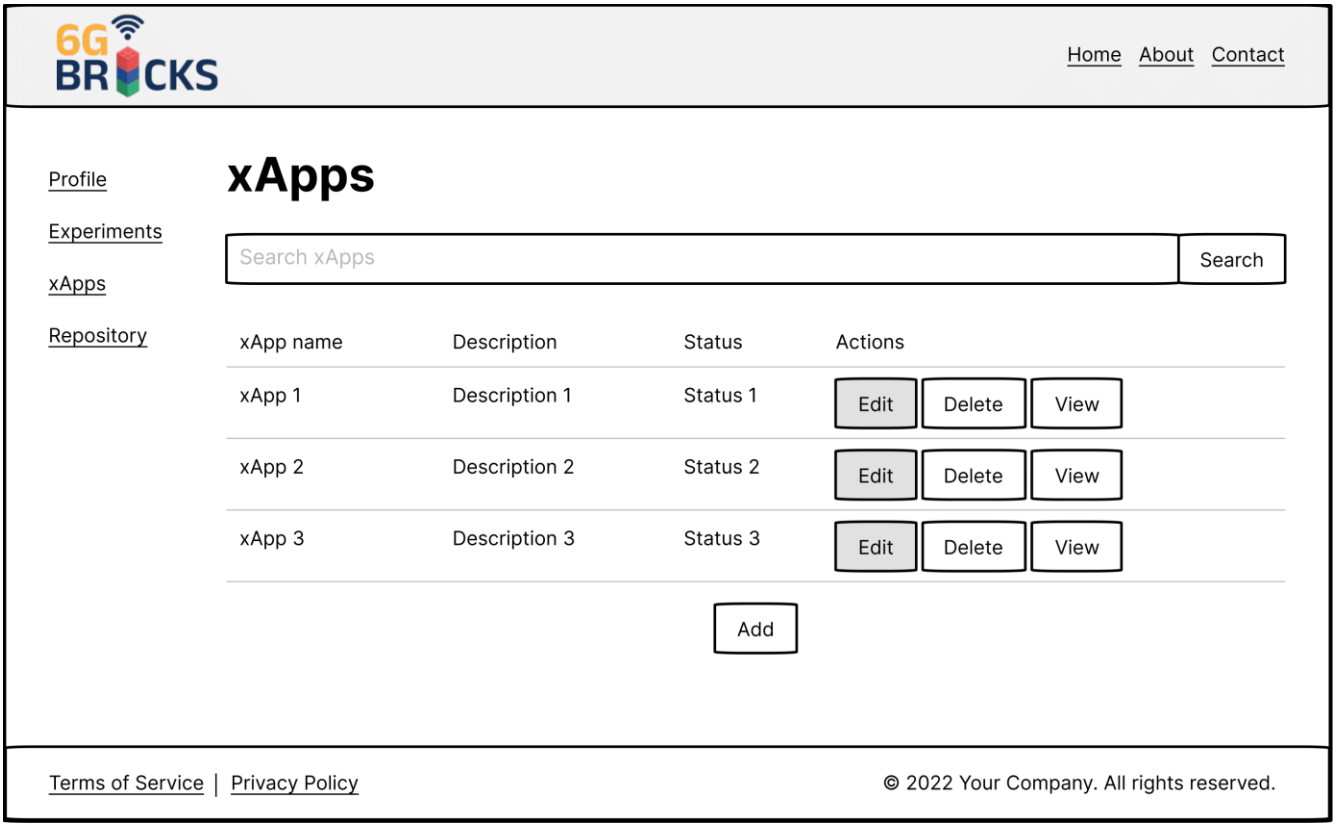

Figure 5: The xApp list page.

3.2 Conflict resolution algorithms for the Intent Management Engine

The 6G-BRICKS Intent Management Engine, as introduced in D2.2, is responsible for processing, reasoning, and fulfilling intents related to Service onboarding at the 6G-BRICKS facility (i.e., autonomously driving decisions and actuations that fulfill requirements, goals and constraints expressed by intent). An intent engine is a critical component of Intent-Based Networking (IBN), i.e., an autonomous paradigm evolution in network and service management [3] - [4]. IBN is poised to provide an easier way for the user to influence the communication system. Enhanced design and performance might be a result of the expressed intent, but intent-based design does not guarantee enhanced performance. As illustrated in Figure 6, a higher-level intervention from the users is regarded as an intent; see Figure 6 for the interaction between the user's high-level requirement, or intent, and the system's feedback in the form of reports. The required information from the user to interact with the network through intents does not encompass low-level or configuration level information. An autonomous network can understand (interpret) this intent from the users and pass it through the layers, extracting required information for different network nodes, finally providing a configuration for the involved network functions. In the following sections, we will describe the innovations of the novel components inside the 6G-BRICKS experimentation plane, in terms of intent negotiation and closed-loops conflicts mitigation(s) [17], from the (lower-level) Resource layer to the (higher) Business layer. Our technical analysis highlights the importance of IBN in autonomous networking, highlighting the need for improved intent integration in service modeling and network management, the need for domain-agnostic abstract representation models, and the need for learning as a fundamental function in autonomous networks.

As per the existing challenges, the International Telecommunications Union (ITU) Telecommunications Management Network (TMN) [1] initially proposed a hierarchical management model, focusing on business and

device objectives for network configurations. However, the motivation for integrating intent-based decision making in future networks is rather driven by the possibility of abstract network management and resource provisioning, driven by users' and service providers' intents. The latter allows for agility and adaptability in service and infrastructure management, but the issue of interpreting the user and provider intents accurately remains a critical challenge.

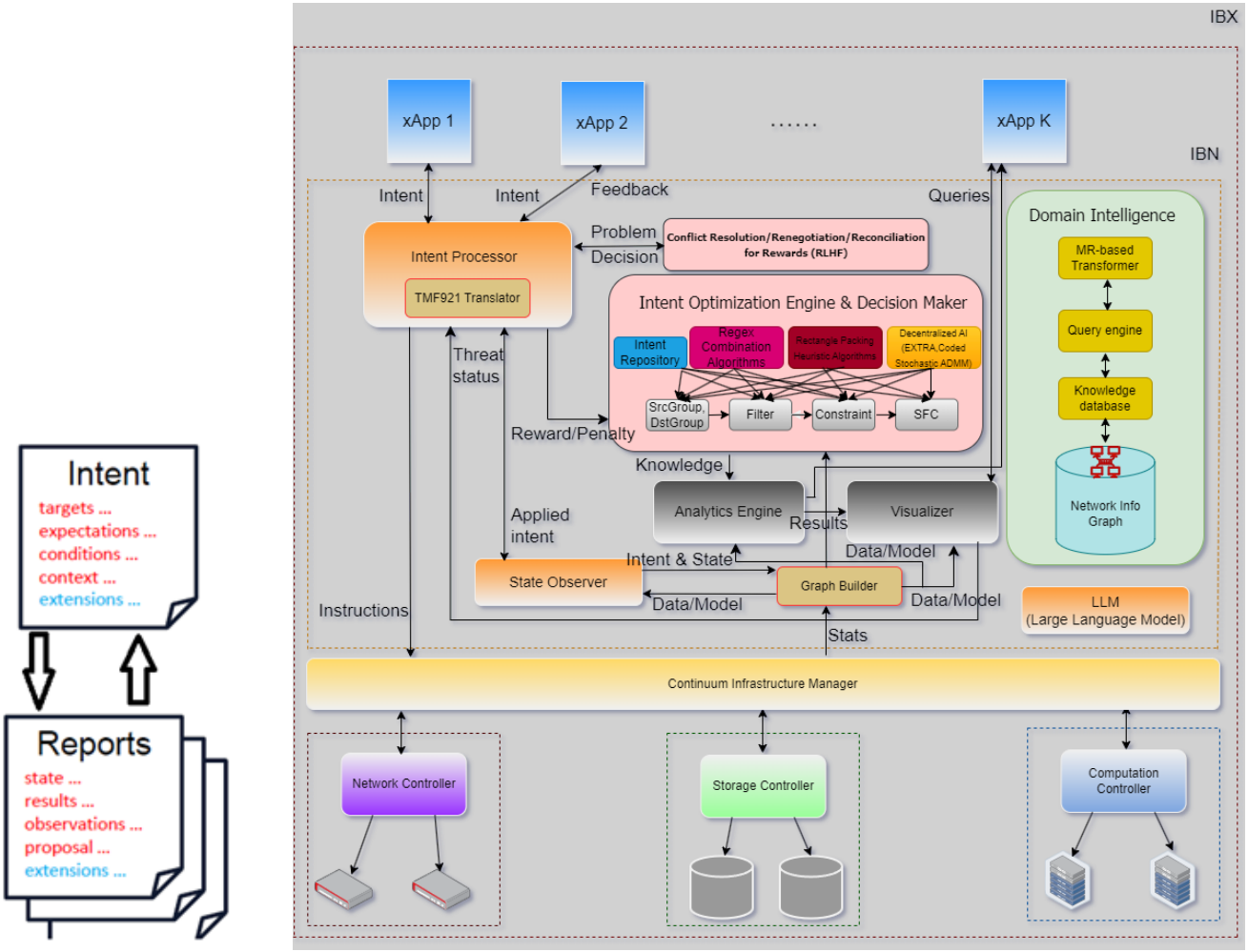

Figure 6: Intent Management in 6G-BRICKS.

3.2.1 6G-BRICKS Intent-Based Networking workflows

6G-BRICKS is aligned with the TMF IDAN Catalyst project, which offers a reference implementation of *intent negotiation* and *intent reporting* workflows, following the TMF921 standard. In Figure 7, a high-level functional flow diagram of the intent-reports plane is demonstrated.

The formal definition of intents adopted in the 6G-BRICKS intent engine is aligned with the TMF921 data model, which essentially supersedes Service Level Agreements (SLAs) with a broader and more flexibly declarative language, which defines application networking and computational requirements. An SLA is a contract between a service provider and a customer, with Intent-Enabled SLAs focusing on declarative intents. In 6G networks, service requests do not need network-specific vocabulary, allowing customers to express their requirements in natural declarative human language. This shift from complicated service provisioning to simpler intents is a

significant improvement in service provisioning. Service providers must ensure performance guarantees, including ultra-low latency, high reliability, spectral and energy efficiency, security, privacy, and trustworthiness. Service subscribers and providers formulate Service Level Objectives (SLOs) to meet these metrics. Flexible network orchestration and management are essential for supporting multiple services. As per Figure 7, it is evident from the academic SoTA [3] - [4] that fully autonomous (zero-touch) intra-domain intent-driven business operations in 6G-IBNs will have a more comprehensive perspective and will oversee converting and breaking down business inputs into domain-specific ones. Technically, networking and IT are used for business purposes, especially in Software-Defined Networks (SDN).

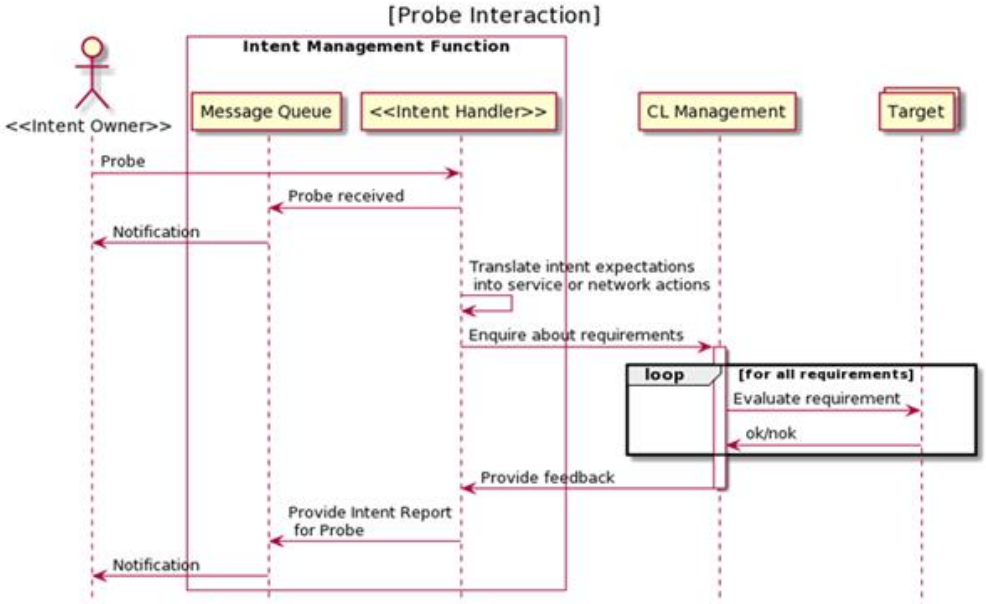

Figure 7: High-level diagram of the intent-reports plane [2].

SoTA [3] - [4] reveals that full autonomous (zero-touch) intra-domain intent-driven business operations in 6G-IBNs will translate and decompose business inputs into domain-specific ones. Challenges include expressing functional and non-functional requirements, constraints, and goals; as well as addressing SLAs, specifications, and objectives. General intents allow for higher-level non-technical goals, such as resource conservation, financial gains, and risk-taking. However, specific problems and challenges remain, such as distributing business intents across management layers and domains, bridging the semantic gap between desired and actual states, and deploying a hierarchy of intent handling functions in scenarios with multiple autonomous layers of management. Even with Closed-Loop Automation, scalability, and network flexibility, which are business-driven and technology-independent aspects that should be achieved via intents, are not always realized. The service (business) intent should be able to be broken down into smaller components, be technology-neutral, and have all implementation details contained by the intent handler and kept secret from the intended owner. The existing SotA criteria do not always meet the latter requirement(s).

3.2.2 Conflict Detection and Policy Resolution Strategies overview

Intent Conflict Reconciliation (CR) is among the key innovations of the 6G-BRICKS intent management system. The CR term is mainly used and associated to indicate the convergence of the user's desires and the system's status); 3 such strategies, that can operate synergistically, are identified in the literature:

1. **Policy resolution** algorithms in the application intent and the connectivity intent scopes via **Regex Combination Algorithms** (pair-wise regular expression comparisons & filtering rules).
2. **Policy resolution (reconciliation)** algorithms solving the **Bin packing heuristic (optimization)** problem.
3. Achieving decentralized consensus **(reconciliation)** with "**end-to-end AI" & Game-Theoretic algorithms** (from the Business Plane to the Resource layer).

We describe below each of the three main solution strategies for the intent CR.

1. **Policy resolution algorithms in the application intent and the connectivity intent scopes via Intent Model UML description.**

The current approach is based on networking service graphs and data modelling languages along with LCM [67]. This conflict detection algorithm takes an incoming intent and performs pair-wise regular expression comparisons against existing intents in the database. Any pair of intents that conflict with each other (i.e. the incoming intent and one from the database) will cause a conflict. If all pair-wise comparisons pass with no conflict, it is safe to introduce the incoming intent to the database. Finally, we apply filtering rules. The objective of the filtering is to decrease the number of candidates that may conflict with the incoming intent, even before being registered.

2. **Policy resolution (reconciliation) algorithms solving the Bin packing heuristic (optimization) problem.**

The principle states that two objects (resource blocks) cannot share the same time slot concurrently -> intent conflict detection. A two-Dimensional resource allocation example is conceptualised for both Application/Business/Resource (Space) domain versus (Time) domain. The intent (resource) conflicts problem can be solved via the rectangle bin packing optimization problem. The bin packing problem is an optimization problem in which items of different sizes must be packed into a finite number of bins or containers, each of a fixed given capacity, in a way that minimizes the number of bins used. In terms of time complexity: It requires $\Theta(n \log n)$ time, where n is the number of intent resources to be packed. Also, many refined algorithms exist. It can be observed that the reasoning logic of bin packing in Figure 8. The formulation idea derives from the scheduling problem. We focus in this PoC on downlink (DwL) scheduling, with one base station (BS) servicing both throughput intensive (Enhanced Mobile Broad Band, or eMMB) and ultra-low latency users (Ultra-Reliable and Low Latency Communications, or URLLC). Then, we formulate a 2D grid-typed Time-Frequency resource allocation. Our goal is to find the resource allocation that satisfies the URLLC users' demands, with minimal losses for eMBB users in terms of throughput and subsequently schedule all the remaining resource blocks to the eMBB services.

© 6G-BRICKS 2023

ALGORITHM 1: Decentralized Consensus Optimization Based on the ADMM

Input functions f_i ; initialize variables $x_i^0 = 0, \alpha_i^0 = 0$; set algorithm parameter $c > 0$;

For $k = 0, 1, \therefore$, every agent I do

Update x_i^{k+1} by solving $\nabla f_i(x_i^{k+1}) + \alpha_i^k + 2c|N_i|x_i^{k+1} - c\left(|N_i|x_i^k + \sum_{j\in N_i} x_j^k\right) = 0$;

Update $\alpha_i^{k+1} = \alpha_i^k + c\left(|N_i|x_i^{k+1} - \sum_{j\in N_i} x_j^{k+1}\right)$;

End for

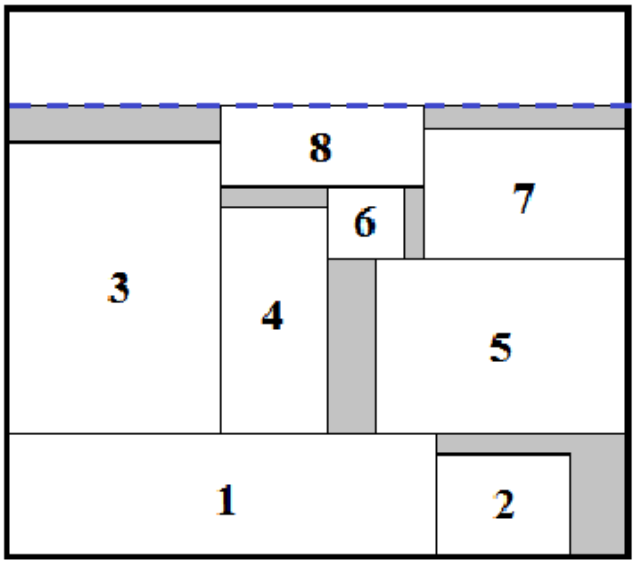

Figure 8: Bin packing approaches [6] and [20].

3. **Achieving decentralized consensus (reconciliation) with "end-to-end AI" (from the Business Plane to the Resource layer).**

Ensuring that the network accurately and automatically deploys the intent throughout its lifecycle is a crucial part of IBN. In the literature, numerous network assurance strategies for data center IBN networks are examined to achieve this goal. Methods of data preparation and Machine Learning (ML) models for the time series forecasting problem are suggested for the assurance model. Three primary ML models are built using the Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) architectures. Most of the SotA literature suggests that the LSTM: Long-Short Term Memory and CNN-LSTM models are the best options due to their high initial accuracy for short-term predictions of one- or two-time steps. Nevertheless, the CNN-LSTM model ought to be disregarded when forecasting multi-time-step CPU use due to its accuracy drop. In conclusion, the best-performing model from the prediction shift results is the LSTM model.

Solution (a) - Linear convergence of the ADMM (Alternating Direction Method of Multipliers)

Decentralized consensus optimization involves a network of agents minimizing their local objective functions over

a common decision variable. This approach uses the ADMM algorithm, which converges quickly and is considered efficient. This study establishes the linear convergence rate for the problem with strongly convex local objective functions, providing a theoretical convergence rate in terms of network topology, local objective function properties, and algorithm parameter. These results serve as a performance guarantee and guideline for ADMM convergence.

Decentralized optimization problems arising in networked multi-agent systems, multi-vendors and tenants include reconciliation and coordination of the intents. Data is collected and stored in a distributed manner, avoiding the need for a storage and processing center. This approach allows for decentralized and collaborative computing tasks among agents, reducing the risk of central computation overload or network congestion. This approach is powerful and efficient.

Hereby, we focus on decentralized consensus optimization, an important class of decentralized optimization in which a network of L agents cooperatively solve:

$$\min_{\tilde{x}} \sum_{i=1}^{L} f_i(\tilde{x}), \tag{1}$$

a common optimization variable $\tilde{x}$, where $f_i(\tilde{x}) : \mathbb{R}^N \rightarrow \mathbb{R}$ is the local objective function known by agent i. Examples of $f(\cdot)$ function include least squares, regularized least squares, as well as generic ones. The values of $\tilde{x}$, can stand for all network intent objectives, such as *SLAs, SLOs, latency, power, bandwidth requirements*, etc. The main goal of this multi-constraint optimization problem is to reach a common consensus convergence goal to ultimately **reduce the Operational Expenditure (OPEX) costs** for vendors.

There exist several methods for decentralized consensus optimization, including distributed sub–Gradient Descent (GD) algorithms, dual averaging methods, and the ADMM algorithm. Among these algorithms, ADMM demonstrates fast convergence in many applications, however, how fast it converges and what factors affect the rate are both dependent on the trial and test stage (hyperparametric).

The ADMM algorithm is applied to a decentralized consensus optimization problem, revealing that if the objective function is strongly convex, the decentralized ADMM converges at a globally linear rate. Factors affecting convergence rate include network topology, objective function condition number, and algorithm parameter. Numerical experiments support these findings and provide insights for fast convergence [16].

Solution (b) - EXTRA: AN EXACT FIRST-ORDER Algorithm for Decentralized Consensus optimization

The decentralized Exact first-order algorithm (EXTRA) [14] is introduced for consensus optimization. EXTRA uses a fixed large step size and synchronized iterations, allowing uniform convergence to an exact minimizer of f. It is different from the Decentralized Gradient Descent (DGD) method, which uses diminishing step sizes. EXTRA has the best-known convergence rates among existing first-order decentralized algorithms for decentralized consensus optimization with convex Lipschitz objectives. It also has an ergodic convergence rate with strongly convex objectives and converges to an optimal solution at a linear rate.

Again, examples of $f(\cdot)$ function can be solved (approximated) with *decentralized averaging, learning, estimation, sparse optimization, and low-rank matrix completion problems. Functions fi can take forms of least squares, regularized least squares, as well as more general ones.*

The solution discusses the issue of DGD converging slowly to an exact solution with diminishing step sizes, while it is faster with fixed step sizes but stalling at an inaccurate solution. The update formula of EXTRA is obtained by comparing two DGD update formulas, arguing that the point is consensual and optimal.

Solution (c) - Coded Stochastic ADMM for Decentralized Consensus Optimization with Edge Computing

ALGORITHM 2: EXTRA

Choose $\alpha > 0$ and mixing matrices $W \in \mathbb{R}^{n \times n}$ and $\widetilde{W} \in \mathbb{R}^{n \times n}$;

Pick any $x^0 \in \mathbb{R}^{n \times p}$;

1. $x^1 \leftarrow Wx^0 - \alpha \nabla \mathrm{f}(x^0)$;
2. for $k = 0, 1, \cdots$ do

$$x^{k+2} \leftarrow (I + W)x^{k+1} - \widetilde{W}x^k - \alpha\left[\nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right];$$

End for

Minibatch stochastic ADMMs algorithms are used to develop the distributed learning model, addressing communication bottlenecks and slow response straggler nodes. Given an appropriate minibatch size, it is shown that the minibatch stochastic ADMM-based method converges in a rate of O(1/√k), where k denotes the number of iterations. Through numerical experiments, it is revealed that the proposed algorithm is communication efficient, rapidly responding, and robust in the presence of straggler nodes compared with state-of-the-art algorithms.

In general, compared to GD, ADMM is better suited for decentralized optimization and has been demonstrated to have fast convergence in many applications, e.g., smart grids, Wireless Sensor Networks (WSNs), and cognitive radio networks (SI -> stochastic incremental). The convergence and communication properties of proposed algorithms are provided through theoretical analysis and experiments. CSI-ADMM [7] has a O(1/√k) convergence rate and O(1/υ2) communication cost, where k denotes the number of iterations and υ is the target accuracy. In addition, the trade-off between the convergence rate and the number of straggler nodes, as well as the relation between minibatch size and stragglers, are theoretically analysed. Numerical results from experiments reveal that the proposed method can be communication efficient, rapidly responding, and robust against the straggler nodes.

3.2.3 6G-BRICKS intent conflict resolution and consensus convergence strategy

Intent-Based Networks (IBNs) in the 6G era should become equipped with toolchains (e.g., OSS/BSS) for governing the autonomic network(s) in such a way that input to the autonomic network, e.g., policies and goals, is "***conflict-free***". In practice, it is required to attain: (1) goals conflict resolution, (2) policy conflict resolution, and (3) resolution of bad Service Chaining (problems such as loops that may occur in certain service chain instances that should be avoided). To this end, 6G-BRICKS contributes a Conflict Resolution engine offering decentralized consensus (reconciliation). The main inputs to the CR engine (as shown in Figure 9) shall include:

1. High Level Business Goals/Objectives for the Network, Service Profiles and SLAs,
2. Network Service Definitions/Graphs and Policies,
3. Application Intents and Profiles/Policies,
4. Other types of configuration data.

© 6G-BRICKS 2023

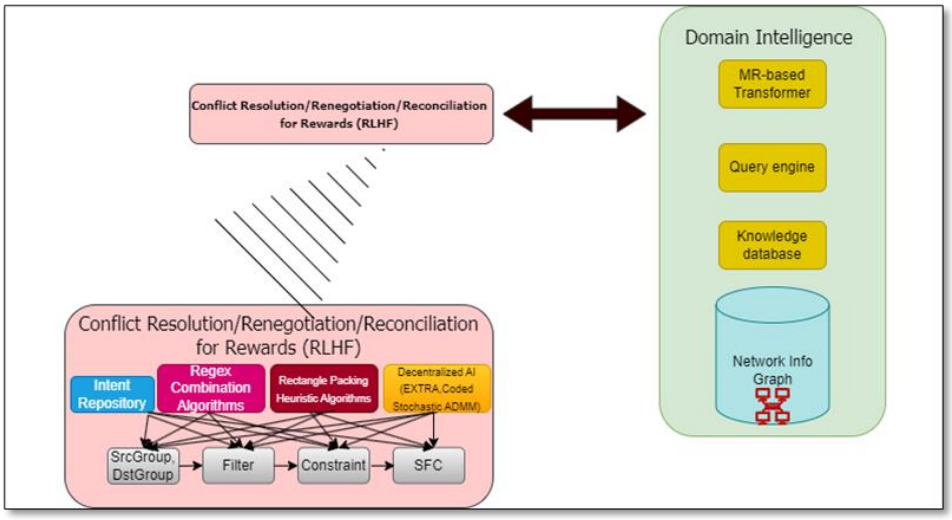

Figure 9: 6G-BRICKS intent conflict resolution engine.

As a core component of the CR engine, 6G-BRICKS will design a new admission control algorithm that leverages the short and long-term correlation of the resource monitoring traffic metrics inside the intent engine to avoid conflicts and maximize admission control efficiency. To apply the decentralized consensus (reconciliation) we follow these steps:

1. Based on a 5G trace Dataset (the MILANO dataset [15]), we have extracted each service type and post-processed (mapped) the network traffic per each Base Station unique id.
2. We utilize our previous Admission Control algorithm(s) e.g., Soft Actor Critic (SAC) [18] as a comparison baseline to see where our next innovation strategy will become more efficient for the intent resolution policies.
3. Finally, we plan to leverage the previous Admission Control algorithms (Deep RL-empowered) with most efficient and faster converging mathematical strategies as well as introduce more input constraint parameters - e.g., blocking probability for the User Equipment (UE), Mobile Network Operators (MNOs) pricing/bidding metrics [8].

The simulation study will leverage the MILANO DATASET, which is a multi-source real dataset of urban life in the city of Milan and the Province of Trentino. The study examines the relationship between meteorological context and cellular traffic loads using telecommunication and weather data from Milan from November 2013 to January 2014. The data includes information on received SMS, sent SMS, incoming calls, outgoing calls, and internet activity. Weather data includes humidity, temperature, and wind speed. The data is aggregated by hours and listed over 300 meters squared within the city, divided into 1x1m grids. Finally, a binary adjacency matrix is presented to represent connectivity between grids.

Our leveraged admission control framework [18] possesses several Actor-Critic Learning Approaches for 5G/6G Network Slicing Control. For example:

- Static allocation (no prediction).
- SAC.
- Statistical SAC with less hyperparameter sensitivity.
- A combination of Deep Q-Network (DQN) and policy gradients in form of Actor-Critic.

© 6G-BRICKS 2023

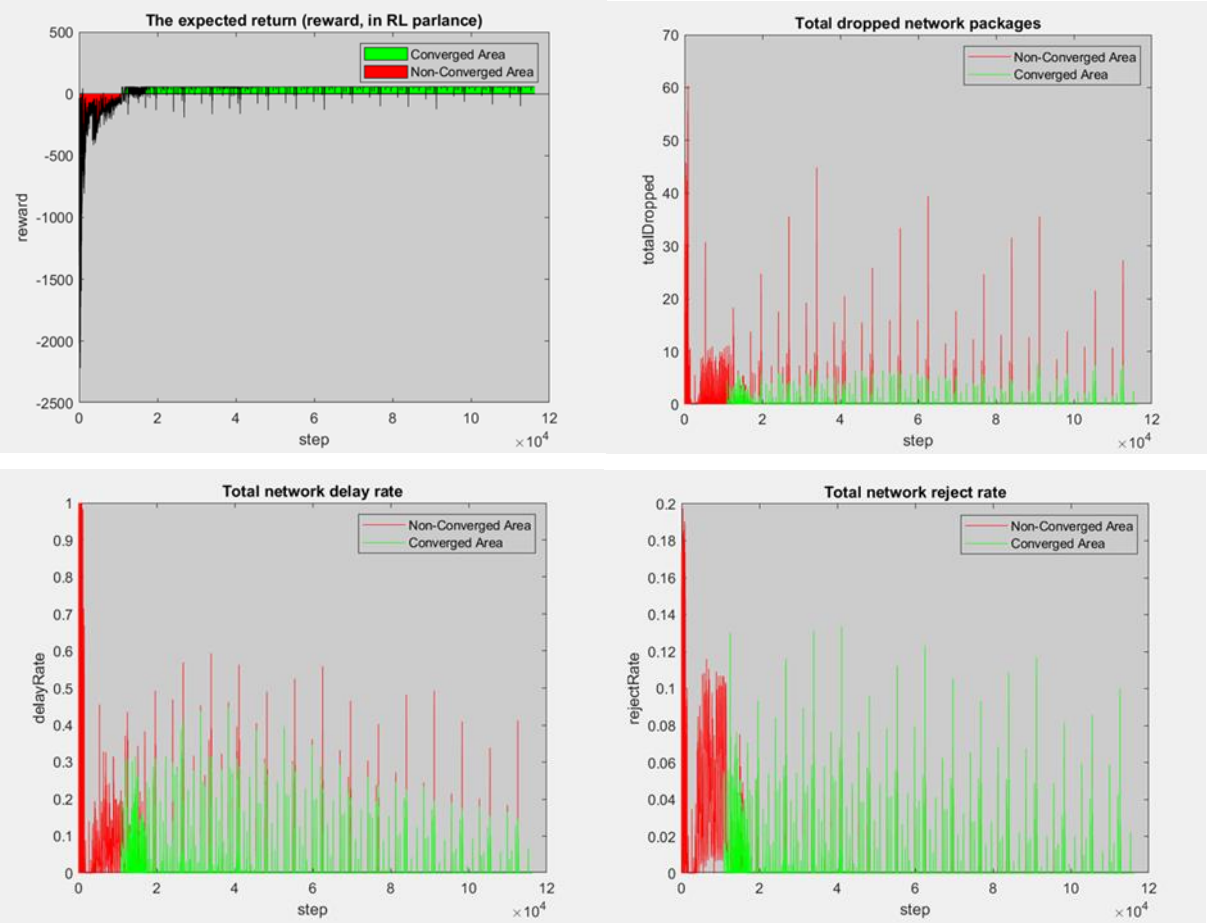

Figure 10: SAC admission control simulation results.

A short description of the SAC algorithms and Applications follows next. It is well known that model-free deep reinforcement learning (RL) algorithms suffer from high sample complexity and brittleness to hyperparameters. SAC is an off-policy actor-critic algorithm based on the maximum entropy RL framework. It aims to maximize expected return and entropy while acting randomly. SAC has been extended to improve training and stability, including a constrained formulation. Technically speaking, the most efficient target is to migrate from the Maximum Entropy Reinforcement Learning stage into the Soft Policy Iteration-leveraged SAC stage. The goal for an efficient admission policy control AND intent conflict resolution is fully Automating Entropy Adjustment for Maximum Entropy RL [9] [10].

In Figure 10 it is shown, that the pre-built SAC algorithm converges very fast; thus, after convergence point, we can obviously observe much less network packets dropped, total network delay rate and total network reject rate. We plan, as stated before, to increase the number of such constraints by adding more customer-oriented SLAs inputs to the game-theoretic engine of a novel SAC model. This SAC will have the potential to converge even faster than the generic SAC version, as seen hereby; and possibly exploit the virtues of ADMM, EXTRA and sub-linear convergence algorithms we described before.

Finally, in Figure 11 , we illustrate the heatplots of the 5G resource network monitoring metrics. Those Heatplot depict the intercorrelation between different resource usage metrics from the network traffic conditions (*actual* and *scheduled*).

We shall exploit the dynamics of time-varying correlation coefficients of the observed networking metrics. Our solution will encompass an admission control (for 6G slicing) via wireless traffic prediction with Gaussian Process

Regression networks (GP: Gaussian Copulas). In the training phase of the GP, our solution will be empowered by (sublinear converged) ADMM framework, to achieve good trade-off between communication overhead and approximation accuracy. Thus, it is vital to consider the correlation structure in any networking simulation model. We will import novel parameters-constraints inside our novel Admission control algorithm for the intent reconciliation:

- UE blocking probability
- RAN slice planning metrics (i.e., min/max Radio Resource Quota (%) per RAN slice)
- Average RAN slice throughput (Mbps).

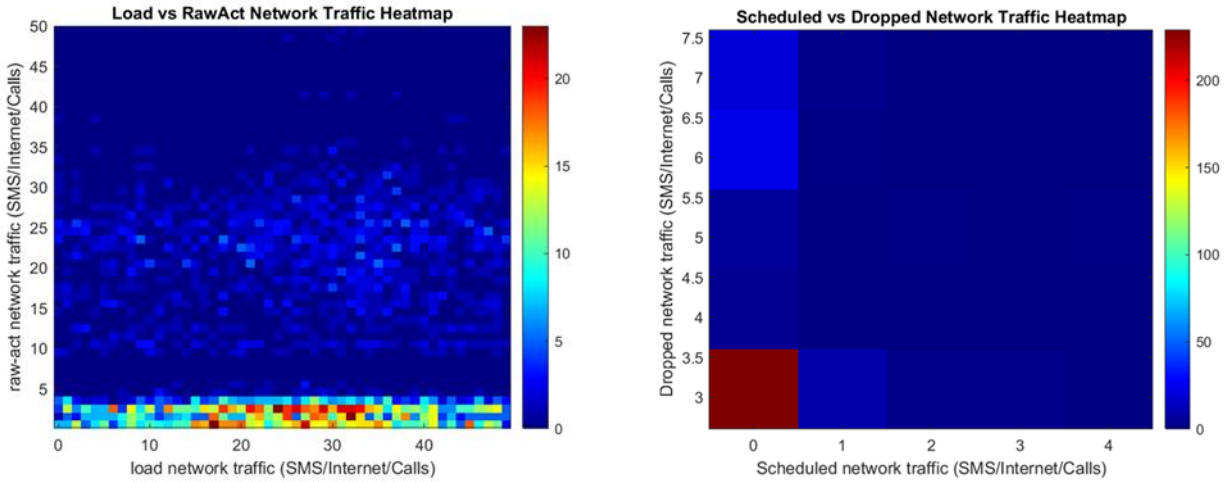

Figure 11: Heatmaps of the 5G/6G network monitoring metrics self-correlation.

3.3 Optimization algorithms for intent-driven cloud continuum

The Cognitive Compute Continuum Manager (CCCM) is responsible for the management of the various π-Edge platforms. A core component of the CCCM is the Application Orchestrator, which interacts with the CCCM's System Catalogue. This catalogue entails the Service Function Catalogue and the Platform as a Service (PaaS) Service Catalogue. The Service Function Catalogue contains several available service functions that the user can request. The catalogue defines the number of resources (CPU, RAM, etc.) required by the Service Function. The PaaS Service Catalogue includes the available PaaS Services offered to users. A PaaS Service is defined by a list of Service Functions from the Service Function Catalogue. For both catalogues, the user can define parameters regarding the scaling policy, storage capacity, and the number of copies for a Service Function or a PaaS Service.

The Application Orchestrator provides the optimization algorithms that aim to determine the ideal node that will host a user's request for a service instantiation. For this purpose, various hard and soft requirements are considered. The hard requirements are the minimum amount of compute resources that is required for the execution of a PaaS or Service Function, while the soft requirements refer to the necessity to minimize the latency, maximize the bandwidth, relieve overloaded nodes/clusters, minimize energy, and others. The soft requirements and their importance create the optimization strategy that is considered when placing the service. There is a list of optimization strategies in the system, and each one serves a different purpose (e.g., minimizing delay, relieving overloaded systems, etc.).

The optimization algorithm is depicted in Figure 12. Firstly, when attempting to assign a request to the system, the algorithm considers the request's hard and soft requirements. In the case of a Service Function request, all available nodes are checked, and a list of the nodes that meet the hard requirements is created. In the case of a

PaaS Service request, due to the requirement that the entire PaaS Service must be placed in the same cluster, a list of available clusters is created. Then, considering the optimization strategy, in the case of the Service Function request, a score is assigned to each node, and the node with the highest score is selected for placement. For the PaaS request, the clusters near the requested coverage area are selected, and then, iteratively, each Service Function is placed (virtually) on the best node of the cluster under evaluation. Then, the scores of each Service Function placement are summed up to determine the overall score of the cluster for the placement of the PaaS Service. In the end, the cluster with the highest score is selected.

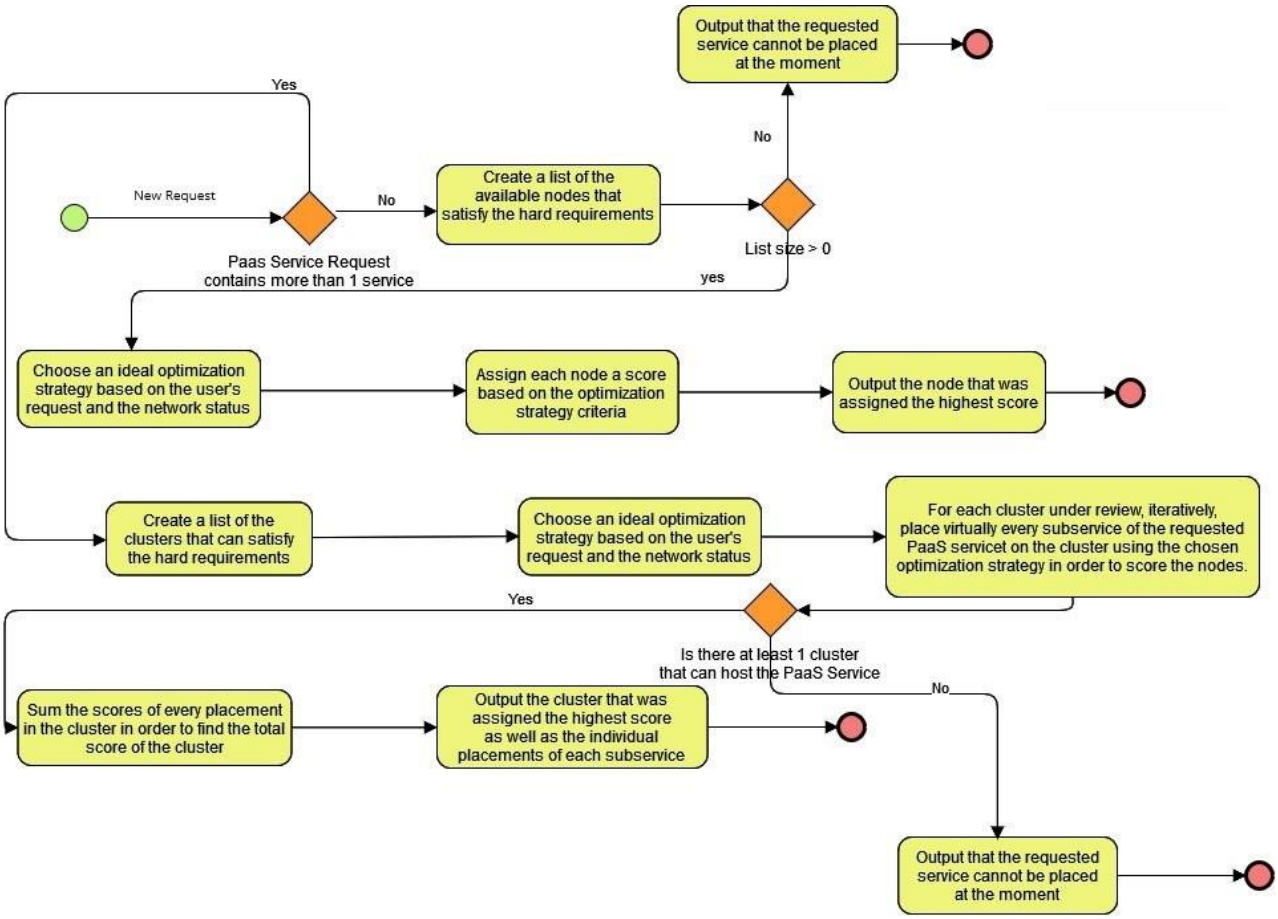

Figure 12: High-level diagram of the optimization algorithm.

The algorithm utilizes certain attributes that have been defined, which make up the optimization strategy, to decide on the optimal cluster and node for placing a Service Function or a PaaS Service. The attributes under consideration are:

- CPU: The compute power that the service intends to use. If the node under consideration cannot provide the required computing power to the service, it is automatically rejected.
- RAM: The memory the service intends to use. If the node under consideration cannot provide the required memory to the service, it is automatically rejected.
- Latency: The latency between the controlled node and the end user of the service. Latency is one of the most important attributes. It functions both as a hard requirement, meaning that the maximum latency of the service must be the defined one, and as a soft requirement, meaning that we try to minimize it as much as possible.

- Bandwidth: The network capacity. Also, an important attribute which can be used both as a hard and as a soft requirement when placing a service.
- Energy: The metric relates to energy consumption. According to related research [21], [22] , idle processors consume relatively little energy, and this increases sharply when the processor starts to be used, and then the increase is relatively linear. Therefore, when taking this attribute into account, the algorithm tries to place services on nodes that are not idle.
- Duplication Space: This attribute exists to provide preference to nodes that will allow the service to create more copies of itself on the node where it will be placed, if needed.
- Prioritize Full: This attribute, if used in the optimization strategy, gives points to nodes that tend to fill up with memory and computing power demands.
- Prioritize Empty: This attribute, if used in the optimization strategy, gives points to nodes that are not using a large percentage of their computing power and memory.
- Prioritize Underloaded clusters: This attribute gives weight to placing services in clusters that are less loaded.
- Location Requested: If the user requests their service to be placed in a specific cluster, this attribute will give more weight to the nodes of that cluster.
- Load Prediction: This attribute collaborates with Predictive Neural Networks (PNNs) to estimate the consumption of nodes at the future. Based on the other attributes and the estimates of neural networks, the future state of the nodes is considered for placing the service.
- Same Node Bias: When we have a PaaS Service which consists of many Service Functions, often for simplicity, we want the services to be placed on the same node. This attribute examines the increase in resource consumption (in percentage) for the node where we placed the previous service, in case we place the next one there as well.

To better understand the algorithm, let us see an example. First, the input request to the optimizer has a form like the next one:

```
{
  "service_name": "myService",
  "autoscaling_type": "maximize_performance",
  "optimization_strategy ": "minimize_latency",
  "count_min": 1,
  "count_max": 5,
  "location": "Athens"
}
```

Where “service_name” is the name of the requested service, “autoscaling_type” defines the required resources and their autoscaling policy, “optimization_strategy” defines the optimization criteria and their priorities, “count_min” and “count_max” are the minimum and maximum numbers of replicas for the service, respectively, and the “location” is the desired service zone. The last three parameters are optional, and they may be incorporated on optimization strategy. In this example, we assume that the optimization strategy considers the metrics “*latency*” and “*Prioritize Empty*” to be of the same importance and everything else to be unimportant. We also have 2 clusters with 2 nodes each:

- Cluster A:
  - NodeA1:
    - CPU: 3 cores used out of 12 cores available

        - RAM: 5 GB used out of 16GB available
        - calculated latency: 8ms
    - NodeA2:
        - CPU: 4 cores used out of 8 cores available
        - RAM: 15 GB used out of 16GB available
        - calculated latency: 8ms
- Cluster B
    - NodeB1:
        - CPU: 15 cores used out of 16 cores available
        - RAM: 8 GB used out of 32GB available
        - calculated latency: 12ms
    - NodeB2:
        - CPU: 8 cores used out of 16 cores available
        - RAM: 5 GB used out of 16GB available
        - calculated latency: 12ms)

The service request needs 2 CPUs and 2GB of RAM for its placement. Thus, only NodeA1 and NodeB2 are placed on the list of candidate nodes that can handle the request. The other nodes don't have the resources to handle the request, i.e. they don't satisfy the hard requirements, and so they are discarded. Next, the optimization strategy will assign a score to each candidate node. NodeA1 will receive a higher score because its latency is better that NodeB2's and its percentage of used resources is lower. So, the service will be placed on NodeA1 of Cluster A and the optimiser's response will be like the following:

```
{
  "selected_cluster": "Cluster A",
  "selected_node": "NodeA1",
  "ip": "example_ip",
  "port": "example_port"
}
```

Overall, the optimization algorithm brings many new features and strategies to improve service deployment in edge clusters. A key innovation lies in considering both hard and soft requirements, allowing the algorithm to effectively balance resource constraints as well as metrics related to user experience and network adaptability. In addition, the algorithm can handle complex networks of clusters and nodes. The algorithm has also the ability to handle different types of services, single and composite ones.

In addition, the integration of PNNs for load forecasting reflects a forward-looking approach, predicting future resource consumption. The use of forecasting techniques and PNNs is used to predict the traffic of the network and therefore make better placement decisions, such as in [11], which proposes neural networks for auto-scaling by predicting the required number of virtual network function instances based on traffic demand. Many researchers have tried using forecasting networks for such purposes, as analysed in literature review papers [12] and [13]. Specifically, [12] presents a systematic review of ML methods for service placement, revealing trends such as a preference for distributed microservice architecture, on-demand resource estimation, and multi-objective approaches. It highlights the reinforcement learning as a prominent trend in ML research for addressing service placement challenges in Internet of Everything (IoE) environments. On the other hand, [13] provides a comprehensive survey of ML/DL-based resource allocation (RA) mechanisms in MEC, including tutorials on their advantages and enabling technologies. In our case, we are using predictions about the future state of each node, ensuring, with high possibility, that the selected node will remain the optimal one for the near future as well.

© 6G-BRICKS 2023

All things considered, the above-mentioned developments contribute to an intelligent and flexible optimization algorithm tailored for intent-driven cloud continuum management.

© 6G-BRICKS 2023

4. Explainable AI for DMO and Anomaly Detection

4.1 Introduction

Anomaly detection, fault detection, or outlier detection are similar terms that refer to the process of identifying data points that are dramatically different from the rest of the data points [23] - and thus these terms will be used interchangeably as they describe a similar workflow in the 6G-BRICKS DMO. As the newly proposed standard-compliant 6G architectures aim to introduce fully automated networking, it is apparent that a very important part of this automation is detecting abnormal behavior in next-generation networks. Accurate detection of such behavior will allow network managers to take early and precise action to restore the system to normal operating conditions or more advanced networks will be able to self-heal through selecting auto-deployed mitigating actions.

Anomaly detection is one of the main functions of the 6G-BRICKS DMO, so that for any experiments that fail to be deployed or do not operate as intended, both operators and experimenters can be aware of the issues and take appropriate actions. The DMO Anomaly Detection layer also includes an Explainability Module, that aims to provide sufficient explanations regarding the nature of the detected anomalies to experimenters, thus completing a novel XAD framework applied to the 6G domain, that combines SoTA anomaly detection algorithms with robust and comprehensible explanations for experimenters.

A basic flowchart of the framework is presented in Figure 13. In the following sections we present the techniques and algorithms applied for XAD. In section 4.2 we explain how the system is monitored to extract the necessary data for training the proposed AI algorithms, whilst in 4.3 we present SoTA approaches for fault detection and in section 4.4 we analyze the SoTA algorithms in XAI that will enable us to provide the explanations of our models. Finally, in section 4.5 we explain multiple novel XAD approaches designed specifically for 6G-BRICKS.

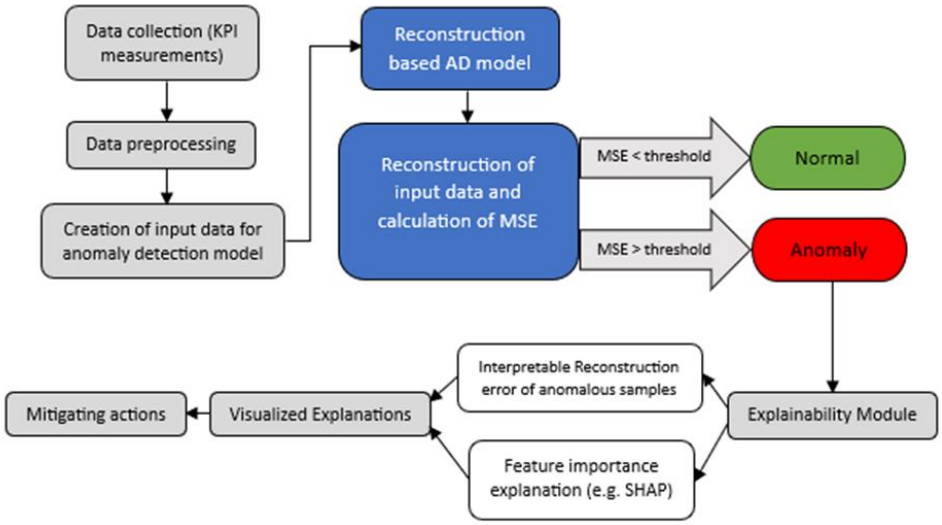

Figure 13: Flowchart of the XAD framework.

4.2 System Monitoring and Data Collection

The performance of the 6G testbeds from the project is constantly being monitored by services such as the KPM XApps, NWDAF and NFVO Collectors; where each of them monitors respectively the Radio Access, Core and Edge networks. These services collect an abundance of system state Key Performance Indicators (KPIs), that are provided as input to the XAD framework, which then identifies and explains faults.

© 6G-BRICKS 2023

The collected data will also be leveraged to create high-quality datasets, which are necessary for training and testing the XAD algorithms before they are deployed. While there are plenty of open-source datasets from various domains (industrial KPIs, server monitoring, financial data) created for anomaly detection [24], some are not adequately preprocessed. Some other datasets lack labeling to be used for evaluating model performance and include trivial anomalies that can be very easily detected; but most importantly are not specific to the 6G networking domain. While it is true that most SoTA anomaly detection frameworks can produce somewhat accurate results regardless of the domain the datasets are derived from; the models should be tested, finetuned and improved with datasets from the area they are deployed in, to ensure high detection accuracy and time efficient performance. When explanations are also involved, like in the case of 6G-BRICKS, datasets produced from the testbeds become of paramount importance to devise explainability modules that provide operators and experimenters with domain-specific, accurate and understandable explanations.

Given the inherent high complexity of 6G networks, the competitiveness of the field and the tediousness of producing data for fault detection, there is a high scarcity of such datasets and the ones available contain simulated data. The project innovates in this regard, aiming to produce datasets from the RAN, Core or Edge domains that can also be reused for 6G automation experiments in research and industry areas alike.

It is important to note that in a 6G network setting, the collected KPIs are mostly Multivariate Time Series (MTS), meaning that multiple KPIs are collected from sensors and/or devices at each time frame with a specified temporal granularity. For instance, data can be collected using any period (second, minute, hour, etc.); creating a high-dimensional dataset with multiple features. The data must be preprocessed after collection to make it usable through means such as data cleaning, normalization, feature independence testing, and thorough labeling of normal and anomalous data points.

4.3 AI-driven Fault Management overview

4.3.1 Classic Machine Learning

Although approaches other than those based on Deep Learning (DL) exist for anomaly detection, they are usually not as effective for high-dimensional data and cannot detect complex anomalies. Such methods include Isolation Forest (IF) [25], K-Nearest Neighbors [26] and Cluster-based Local Outlier Factor (CBLOF) [27]. Despite the drawbacks of these methods, they require less computational resources and training time, so they can be deployed in layers where the anomalies show less complexity and where they can produce similar results to their deep-learning counterparts, thus increasing overall computational and time-efficient performance.

4.3.2 Neural network

Most SotA techniques for Anomaly Detection (AD) in MTS data are based on DL. From the multiple methods proposed in research and industry, the current focus is mainly on Reconstruction-Based models to be implemented and tested. Such models have shown superior results and are generally preferred for real-time AD and continuously changing data over other approaches [28]. Reconstruction-Based models are trained on normal data (that are collected from a system operating in normal conditions, with no anomalies present) and their purpose is to encode the subsequences of high-dimensional time series into a latent space with a lower number of dimensions.

Model inputs are sliding windows, that represent the subsequences of the data and provide the temporal context of the reconstruction process. As described in Figure 14**,** every window is formed according to its size Ws, which indicates the number of time steps contained in the window, and step Ls, which sets the distance between the starting points of two adjacent windows as well as the overlapping of the windows.

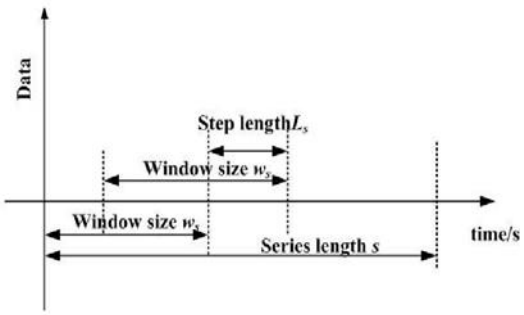

Figure 14: Time series window [29].

The underlying idea is that it is not possible for the model to reconstruct anomalous subsequences in the test phase since it was only trained on normal data. As a result, anomalies are detected by reconstructing a window from test data and comparing them to the actual values, where the difference is defined as the reconstruction error. In some models, the detection of anomalies is triggered when the reconstruction probability is below a specified threshold since anomalous subsequences have a low reconstruction probability [28]. The most common DL model architectures for reconstruction-based AD are presented below:

1. Autoencoder (AE): AEs are a form of neural networks with nonlinear dimensionality reduction capability. They consist of two symmetrical components: (i) the encoder that converts the input into an encoded low-dimensional space representation and (ii) the decoder that attempts to reconstruct the input from the encoded representation.

   The goal of training the model is to minimize the reconstruction error of the AE. The basic AE structure is depicted in Figure 15 and can be described in the following equations:

$$Z_{\text{t-w:t}} = E(X_{\text{t-w:t}}, \phi) \qquad (2)$$
$$\hat{X}_{\text{t-w:t}} = D(Z_{\text{t-w:t}}, \theta)$$

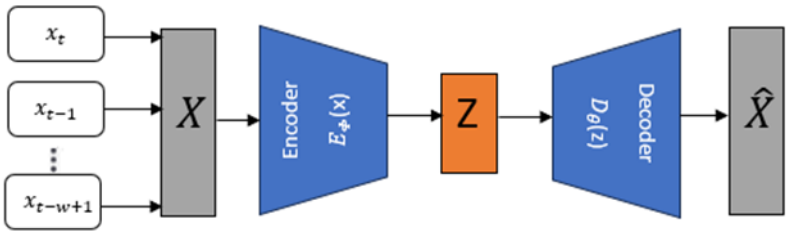

Figure 15: Structure of an AE.

where $Xt-w$ is a time window of length w, E is the encoder with parameters ϕ, Z is the latent space representation of the input X and D is the decoder with θ parameters that produces the reconstructed output. During training, ϕ and θ parameters are updated according to:

$$(\phi *, \theta *) = arg\ m\ \phi,\theta\ in\ Er(X_{\text{t-w:t}}, D(E(X_{\text{t-w:t}}, \phi), \theta)) \qquad (3)$$

where Er is the reconstruction error metric between the input and the output. The anomaly score of each window (reconstruction loss) is usually defined as a mean squared error [28]:

$$\text{score} = ||X_{t-w:t} - D\,(E(X_{t-w:t}\,, \phi), \theta\,)||2 \tag{4}$$

2. Variational Autoencoder (VAE): The latent space in a simple AE is not regularized, thus often resulting in overfitting. A VAE addresses this issue by encoding the input as a distribution (instead of latent vectors in an AE) and outputting parameters of this distribution in the latent space for every input. The VAE then imposes a constraint on the latent distribution, forcing it to be a normal distribution and thus making the latent space regularized. From the structure of the VAE (shown in Figure 16), we can see that the direct output of the encoder is not the latent vector Z, but the mean μ_χ and the variance σ_χ of the latent distribution. If we define the encoder function as:

$$E_\varphi\,(Z_{t-w+1:t}\,|X_{t-w+1:t}\,), \text{ with } \phi \text{ parameters} \tag{5}$$

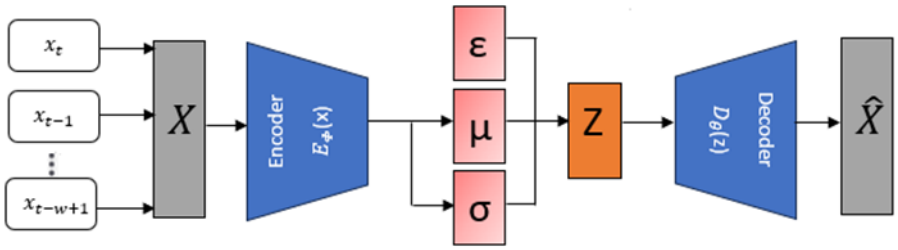

Figure 16: Structure of the VAE.

and the decoder function as:

$$D_\theta\,(\hat{X}_{t-w+1:t}\,|Z_{t-w+1:t}\,), \text{ with } \theta \text{ parameters} \tag{6}$$

A problem that arises is that the latent vector is sampled at random from the distribution, before being inputted to the decoder. This random sampling makes it difficult for the backpropagation algorithm to function, hence a parameterization trick is used to model the sampling process and represent the latent vector Z as a function of the encoder outputs:

$$Z = \mu_x + \sigma_x \varepsilon,\ \varepsilon \sim N(0, I) \tag{7}$$

The loss function is defined as the sum of the *reconstruction loss* and the *similarity loss*. The reconstruction loss is the mean squared error of the input and the reconstructed output, like in the AE, and the similarity loss is the KL (Kullback–Leibler) divergence between the latent distribution and the standard Gaussian. The total loss is the sum of these two losses:

$$\begin{aligned} loss = ||X_{t-w+1:t} - D(Z_{t-w+1:t}\,)||2 + KL(N\,(\mu_x, \sigma_x\,), N(0, 1)) = \\ ||X_{t-w+1:t} - D(\mu + \sigma\varepsilon)||2 + KL(N\,(\mu_x, \sigma_x\,), N(0, 1)) \end{aligned} \tag{8}$$

3. Generative Adversarial Network (GAN): A GAN is an algorithm designed for generative modeling based on game theory [30]. In GANs, training examples are explored to learn the probability distribution that generated them. Two neural networks, namely Generator *G* and Discriminator *D*, contest in the form of a zero-sum game where one network's loss is the other's gain. The generative network's training aim is to increase the error rate of the discriminative network, while the discriminator is trained to distinguish

samples produced by *G* from the normal distribution. This approach can be written as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p(X)}[log\ D(X_{t-w+1:t})] + E_{z\sim p(Z)}[log\ (1 - D(Z_{t-w+1:t}))] \quad (9)$$

where $p(x)$ is the probability distribution of input data, $X_{t-w+1:t}$ is a sliding window, $p(z)$ is the prior probability distribution of the generated variable and $Z_{t-w+1:t}$ is a generated input window.

When performing time-series AD, GANs are usually used along with other DL architectures, in order to map the temporal relations between data and effectively use the reconstruction error as a measure for anomalous samples. For reference, in [31] a GAN architecture is combined with a Convolutional encoder and decoder and in [32] an LSTM generator and discriminator are used, and both the reconstruction error and discrimination loss are utilized for detecting anomalies.

4. Transformers: Transformer networks have also been used for AD, producing competitive results. In [33] a reconstruction-based architecture is proposed, combining adversarial training with transformer AEs; and in [34] a GAN with transformer generators to reconstruct samples is combined with a Dilated CNN that extracts temporal information from latent spaces.

4.4 Explainable AI algorithms for DMO overview

4.4.1 Introduction

The human-centered AI-enabled 6G networks with automated closed loops will be producing a high number of decisions every second, which can be leveraged by processes like resource allocation, fault detection, security orchestration and many more. The risk of losing control over these decisions is high for both operators and experimenters in the case of any possible network fault or failure. Also, any decision the network makes should be able to be explained, at least to a certain degree, to all system stakeholders, since such decisions will be of critical importance to 6G applications.

The black box nature of the most advanced AI models makes the network's decisions hard for operators and users to understand. XAI methods increase the transparency of ML models and provide insight into the produced output. In the case of 6G-BRICKS, explanations will be crucial for network operators to locate the cause of any system faults and take mitigating actions and for experimenters to onboard and configure their applications in the desired way.

The explainability module deployed in 6G-BRICKS (part of Figure 13) will be based on popular XAI techniques, described in the following subsections, and aims to fulfill and measure the following properties:

- Accuracy: accuracy of explanations compared to the real cause of the output.
- Comprehensibility: To which degree the explanations are understandable to different types of stakeholders.
- Scalability: ability to scale based on large input data.

In the following subsections, we first explain how XAI can be integrated into 6G and present the most popular algorithms for post-hoc interpretation (4.4.2). Secondly, in 4.4.3 we provide an analysis of SoTA frameworks that have been applied to various networking data scenarios.

4.4.2 XAI Integration into 6G and Post-hoc Interpretation

MNOs are currently deploying their 5th Generation Networks (5G), bringing new societal benefits to customers. At the same time, the design phase for 6th Generation Networks (6G) has already started. 6G aims to be an enabler

of greater autonomy, and broader human-machine interfacing and to achieve deep connectivity in more verticals. As the transition from traditional model-based explanation to DL progresses, the trust in optimization modules decreases. This reduction in trust means less understanding of the impact of (i) poor/biased/malicious data, (ii) neural network design on decisions, and (iii) network actions for the engineer. Towards this scope, explainability to enable trust in 6G plays a critical role, since it manages a wide range of services from autonomous driving to remote surgical operations.

A new era of mass digital connectivity is underway where increasingly more people, machines and Internet of Things are being connected to automate and digitize traditional services. At the same time, wireless networks have transitioned from an information channel to a critical leader in the new industrial revolution [35]. This evolution has resulted in a 1000x growth in communication data rate demand and an increase in network requirements such as URLLC for tactile control of autonomous entities across various vertical sectors. Orchestrating co-existence via spectrum aggregation between different Radio access technologies is essential towards the end of meeting this demand. Additionally, real-time RRM is critically important but has become too complex for conventional optimization, therefore the need for evolution to an AI-driven ecosystem [36].

Figure 17 [37] below illustrates a proposed XAI Integration Framework in Future Wireless Networks. In the context of this framework, Network Devices are illustrated as Network Environment and Experimenters are denoted as Engineers/Users. XAI provides explanations to consumer users and radio engineers with varying levels of expertise and demand. The mental state of the users affects their interpretation, and the output is fed back to the ecosystem. In Figure 17, the main challenges are highlighted such as malicious attacks and designer noise.

According to the proposed framework in Figure 17, the users and radio engineers are given adequate explanations of the network's actions. More particularly, in this example: (a) service is denied to users (here, experimenters), (b) the engineer (here, operator) is aware of the problem through classical mechanisms of network monitoring and diagnostics. XAI provides different levels of explanations to them in order to achieve different goals, (c) the user is given a high-level statement of the problem, (d) the engineer is given quantitative analysis via local linear model classification of a key feature set, (e) the mental state of the users affects their understanding levels, (f) for the users empathy is a critical aspect to maintain high customer satisfaction, otherwise it leads to increased complaints from customers resulting in loss of revenues for the network and (g) for the engineers, comprehension is critical for further actions.

There are several challenges identified, and the most important are highlighted here: (i) Developing a Human Machine (Brain) Interface that communicates statements to users and engineers without disrupting their lives and fits into existing processes. Particularly by tackling the cognitive biases in human minds, which all impact their degree of comprehension and empathy; (ii) developing an XAI Twin system to work in parallel to the DL systems that are designed for optimization performance. The twin system enables not to sacrifice performance, where at the same time offers adequate explanations, (iii) defense mechanisms against targeted attacks at DL and XAI engines must be developed.

The post-hoc explanations in XAI refer to analyzing models to offer explanations after the model is trained. There is a need for a method which operates post-hoc, after training and which can be used to provide the interpretability of models on the datasets; where the domain knowledge is not available. Post-hoc interpretation can further be divided into two categories:

1. **Global Interpretation**: Global methods describe the average behavior of a **ML** model. This level of interpretability is about understanding how the model makes decisions, based on a holistic view of its features and each of the learned components such as weights, other parameters, and structures. A global explanation describes the working process of the ML method overall. This approach helps to understand the distribution of the target outcome based on the features.

2. **Local Interpretation**: Local interpretation methods explain individual predictions. Local interpretation focuses on the specifics of each individual and provides explanations that can lead to a better understanding of the feature contribution in individuals and smaller groups that are often overlooked by global interpretation techniques.

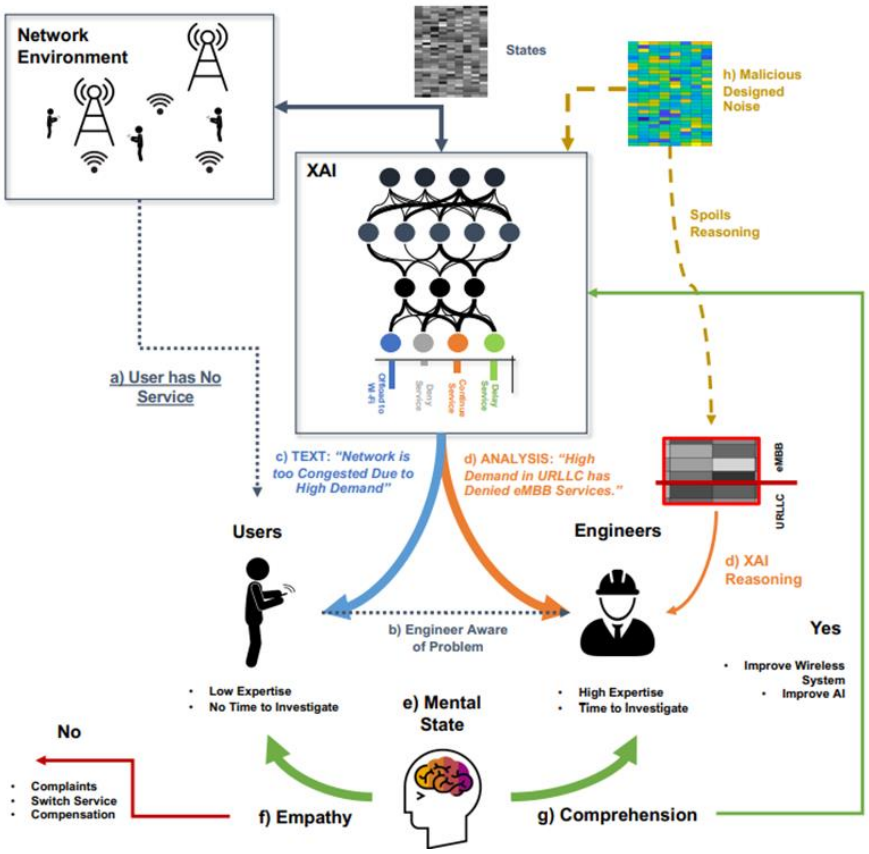

Figure 17: XAI integration framework in future wireless networks [37].

The sophistication of Deep Neural Networks (DNNs) algorithms has recently increased with the ability to process massive network management data sets while extracting useful features to quickly identify anomalies [38]. These algorithms are often difficult to interpret, and their *black box* nature makes it difficult for stakeholders to understand the model's output.

Local model-agnostic explanation methods aim to explain a single prediction of an arbitrary ML model by studying the model through its inputs and corresponding outputs. The neural network-based models are capable of handling very large, high-dimensional data sets with billions of parameters that pass through nonlinear functions.

1. **LIME**: It would be beneficial to know why a model gives a certain prediction, which Local Interpretable Model-Agnostic Explanations (LIME) aims to solve with the proposed LIME methodology [39]. Explaining the predictions of a complex ML model helps practitioners debug the model and build users' trust in the predictions. LIME aims to make the predictions of the neural network model interpretable at the same time by perturbing the input of data samples to observe the resulting impact on the output and how the predictions interact with it. The output of LIME is a list of local interpretations, reflecting the contribution of each feature to the prediction of a data sample. The approach works well with linear models to approximate local behavior. To a certain extent, this assumption is correct when looking at a very small region around the data sample. By expanding this region, however, it is possible that a linear model might not be powerful

enough to explain the behavior of the original model. Non-linearity at local regions happens for those datasets that require complex, non-interpretable models. Not being able to apply LIME in these scenarios is a significant pitfall. LIME generates the prediction probabilities, which still results in poor model interpretability visually, and still makes it hard for the end-user to understand.

2. **SHAP**: The authors in [40] proposed a relatively new technique in ML known as SHapley Additive Explanations (SHAP) to overcome the visualization interpretability of the model output. SHAP supports the interpretation of the neural network, or any complex ML model, by determining how input features contribute to the value of output features.
   This game theory-based framework, known as SHAP, combines several existing explanation methods into one class of additive feature attribution methods [41]. The key idea of SHAP is to calculate the Shapley values for each feature of the sample to be interpreted, where each Shapley value represents the impact that its associated feature generates in the prediction.
   The Shapley value is the average contribution of a feature value to the prediction in all possible coalitions defined as:

$$\Phi(m,x) = \sum_{z' \subseteq x'_1,\ldots,x'_n} \frac{|z'|!\,(M-|z'|-1)!}{M!}\left[m\left(z^i \cup x'_i\right) - m(z')\right] \quad (10)$$

   where z' is a subset of the features used in the model, x is the vector of feature values of the instance to be explained M is the number of features, and $m(z')$ is the prediction for feature values in set z'. When calculating $m(z')$, the i^{th} feature is masked out and then simulated by drawing random instances or the random values of the i^{th} feature from the dataset. Calculating the Shapley value requires computation time and power since there are 2^k possible coalitions of the feature values. One of the advantages of using SHAP is that it can distribute the difference between the prediction and the average prediction evenly among the features of the instance's feature values because of the Shapley values efficiency characteristic.

In [42] the authors introduced the idea of providing interpretation of neural network models. They demonstrated the idea of providing interpretation to the neural network model. The authors emphasized the importance of how hard it is to explain the anomalies extracted by anomaly detection algorithms and extended SHAP to explain anomalies detected by an AE, which is an unsupervised model. Subsequently, they proposed a black box interpretable method which aims to provide a comprehensive root cause analysis. In the framework, the top features with higher reconstruction errors were connected to the features that contributed the most to the reconstruction error. This detailed interpretation of why an instance is anomalous enables domain experts to focus their investigation on the most important anomalies and increase their trust in the algorithm.

The authors in [43] proposed a novel Depth-based Feature Importance for the Isolation Forest (DIFFI), a method to provide interpretation traits using the IF algorithm. This algorithm emphasizes the feature importance required for anomaly detection algorithms. Their algorithm computational cost has been compared to the SotA SHAP and presents significantly lower computational costs.

4.4.3 XAI overview for networking Data

The XAI field studies methods to provide explanations of the decisions taken by unintelligible, black box models, such as DNNs [44] or [45] XAI, is a research field related to trust and transparency of AI-based systems. XAI aims to create trust between machines and humans and create transparent AI that would explain its decisions to its users. In the last few years, many researchers have investigated this area of explainability of AI, which has become a major concern for many application sectors.

The authors in [46] provided an entry point to the problem of interpreting a DNN model and explaining its predictions. They focused on the Layer-wise Relevance Propagation (LRP) technique, to provide interpretations

on real data. The authors demonstrated the ML model's transparency decisions by identifying the relevant input variables. The authors discussed the effect of linking prototypes to the data using a data density function with a graph structure technique.

Model interpretation mainly consists of subjective reasoning from the perspective of an end user or a domain expert. It plays an integral part in how the model evaluations are displayed. The authors in [47] facilitated the model interpretability by leveraging the hidden latent space of a stacked trained LSTM autoencoder model by employing t-Distributed Stochastic Neighbor Embedding (t-SNE). Parallel coordinate plots and heatmaps were generated to identify the essential music elements (features) to understand the value distribution of latent representations by comparing multiple features for a collection of music representation.

The authors in [48] proposed an approach to detect traffic anomalies using LSTM-based RNN. The mobile network information is collected from an LTE Physical Downlink Control Channel (PDCCH), which contains the radio scheduling information and has the benefit (regarding analysis) of being unencrypted and fine-grained, since the messages are exchanged every LTE subframe of 1 ms. The proposed framework presents an F1 score of 1 by monitoring events with a high concentration of people. Similarly, [49] demonstrated novel research on sliding window-based clustering algorithm to work upon continuous 24-hour period time-series data. The proposal intended to capture the malicious network attacks present in a network through a parallel clustering technique. The authors approached Micro-Cluster based Outlier Detection (MCOD) classifier to identify outliers over streaming data. As part of an anomaly detection algorithm, parallel clustering algorithms were presented, which worked concurrently to detect anomalies that might be lost while traversing time-series windows.

The authors in [50] presented their work on detecting anomalies in cellular mobile networks using LSTM-based RNN. The novel methodology adopted by the authors for cell level performance profile generation leveraged the sequence learning capability of LSTMs to train the radio time series data (Call Drop Rate). The proposed framework calculates the error distribution using an RMSE of 14.31 and 15.72 for training and test samples, respectively. This profile generation method is then used to recreate the actual profile and predict the profile of any cell for a given time series. A novel self-adaptive DL framework is proposed in [38] for detecting anomalies in 5G RAN and EPC. The publicly available Czech Technical University (CTU) dataset comprises of unknown traffic collected from a large network with normal and background traffic [51]. An AE-based AD present in RAN quickly captures the anomaly symptoms, which are further analyzed by an LSTM recurrent network model in EPC. The anomalies captured are further communicated to the Monitoring and Diagnoser Module [52]. The experimental results reach the SoTA classification accuracy in cyber defense architecture, hence optimizing the computing resources.

Causal Dataframe: According to the works of [53], a series of global and local explainability methods is investigated in order to analyze the cause of SLA violation prediction in a 5G Network Setup. Towards this scope, a comparative analysis of the methods is performed to analyze the explainability of the predicted violation. The Causal Dataframe method is used to improve the identified cause of the problem and to validate the global explainability results. The results of this work are particularly important for telecommunication providers to quickly identify or prevent a problem that may occur in its operation. Additionally, this method can be applied to a system where preventive action can be taken to avoid unwanted events in the future.

Figure 18 illustrates the most important features that are generated when examining the explainability across the whole dataset. Conclusively, according to the results of this work, it is noted that the most important feature *'crc11_BytesSent_queue5'* ranks first for eXtreme Gradient Boosting (XGBoost) and Causal Dataframe, however, it ranks 2nd for SHAP and Permutation Importance. The two latter methods have identical top four features. Causal Dataframe requires a large set of data that represents the whole dataset. The authors finally propose SHAP to be used in investigating the events of SLA violations.

No	XGBoost	Causal Dataframe	SHAP	Permutation Importance
1	crc11_BytesSent_queue5	crc11_BytesSent_queue5	client_tcp_timeouts/s	client_tcp_timeouts/s
2	client_tcp_timeouts/s	client_tcp_timeouts/s	crc11_BytesSent_queue5	crc11_BytesSent_queue5
3	udp_rx_latency_average	server_tcp_connection_requests_failed/s	server_tcp_connection_requests_failed/s	server_tcp_connection_requests_failed/s
4	server_tcp_connection_requests_failed/s	udp_rx_latency_average	udp_rx_latency_average	udp_rx_latency_average
5	server_tcp_connections_in_established_state	L2_L3_client_bits_received_rate_(kbps)	client_tcp_syn_sent/s	crc10_BytesSent_queue7

Figure 18: Comparison of the most important features using different methods.

4.5 6G-BRICKS Models for anomaly detection and XAI integration

This section describes the novel XAD framework designed for the 6G-BRICKS system. The presented approaches combine SotA Fault Detection techniques, similar to the ones described in section 4.3 (e.g. Autoencoders, LSTM-Autoencoders and Generative models) with post-hoc explainability techniques as seen in Section 4.4 (e.g. SHAP) to complete innovative architectures that will for the first time be tested on the complex datasets extracted during the operation of the 6G-BRICKS's DMO. Our framework aims to provide high AD accuracy along with comprehensive visualized explanations, while also incorporating quantifiable metrics for the quality of the explanations.

4.5.1 Autoencoder with SHAP Integration For 6G-BRICKS

In this subsection, we first introduce our approach for the anomaly detection module, then we present the explainability aspects coupled with the anomaly detection model.

Anomaly Detection: The AE model has been trained using the TELCO dataset (Figure 19), as shown in Figure 20. The model design matched the cardinality of the input and output layers. The experiment comprised of a 3-stacked perceptron model with 120 trainable parameters to train 25920 rows with 10 feature columns. The model is trained in minibatches with 30 epochs with GD with the set of normal cell trace data set. The model optimizes the Mean Squared Error (MSE) loss using the Adam optimizer followed by non-linear activation ReLU function (used in training with dense layer defined) as:

$$f(x) = max(0, x) \quad (11)$$

TELCO Training Dataset

	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8	TS9	TS10	
count	25920.000000	25920.000000	25920.000000	25920.000000	25920.000000	25920.000000	25920.000000	25920.000000	25920.000000	25920.000000	25920.0
mean	0.045872	0.007832	0.005093	0.056173	0.002855	0.002469	0.004668	0.004900	0.024421	0.052546	0.0
std	0.209211	0.088152	0.071182	0.230260	0.053356	0.049630	0.068166	0.069827	0.154356	0.223130	0.0
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0

TELCO Testing Dataset

	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8	TS9	TS10	
count	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26
mean	0.020643	-0.044448	-0.137240	-0.090922	-0.183360	-0.132546	0.095170	-0.024927	0.155276	0.248513	
std	1.010461	1.017034	0.969700	0.965440	0.925530	0.948112	1.118530	1.052804	1.244574	1.172490	
min	-1.147632	-1.260386	-1.791845	-1.944433	-1.638559	-1.633862	-1.304647	-1.380304	-1.057118	-1.481383	
25%	-1.019435	-1.082052	-0.917874	-1.012967	-1.161002	-1.110957	-1.045607	-1.111563	-0.797544	-0.864446	
50%	-0.121470	-0.215110	-0.196092	0.123698	-0.024393	-0.037831	-0.004514	-0.092200	-0.172245	0.389428	
75%	0.932976	0.997093	0.456574	0.636316	0.619230	0.691895	1.108359	0.954963	0.663840	1.162632	
max	2.830161	2.153787	4.492976	5.021790	4.605212	3.878756	5.302026	3.975983	8.089642	4.599852	

Figure 19: The TELCO dataset.

```
Model: "encoded"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 12)]              0

 dense_31 (Dense)            (None, 6)                 78

 dense_33 (Dense)            (None, 3)                 21

 dense_34 (Dense)            (None, 3)                 12

=================================================================
Total params: 111 (444.00 Byte)
Trainable params: 111 (444.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Model: "decoded"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_8 (InputLayer)        [(None, 3)]               0

 dense_36 (Dense)            (None, 3)                 12

 dense_38 (Dense)            (None, 6)                 24

 dense_39 (Dense)            (None, 12)                84

=================================================================
Total params: 120 (480.00 Byte)
Trainable params: 120 (480.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 20: The Autoencoder-based model.

The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time. A callback with an early stopping mechanism of validation loss has been used as one of the regularization techniques to avoid model overfitting. This technique stops training at the point when performance on a validation data set stops improving. The model trained after tweaking the hyperparameters is then used to reconstruct the input data. The trained model then detects anomalies in multivariate network data. When given an anomalous instance, it may not be able to reconstruct it accurately, and this leads to higher reconstruction errors compared to the reconstruction errors for the normal cell traces. The reconstruction errors measure the distance between the original input and its AE-based reconstruction. The reconstruction errors

calculated are then used to detect the anomalies in the test set cell trace file. The MSE for normal instances and anomalous instances are 0.0046 and 0.0086 respectively, as shown in Figure 21. Here, the high MSE scores depict the anomalous cell trace which was not reconstructed well by the trained model.

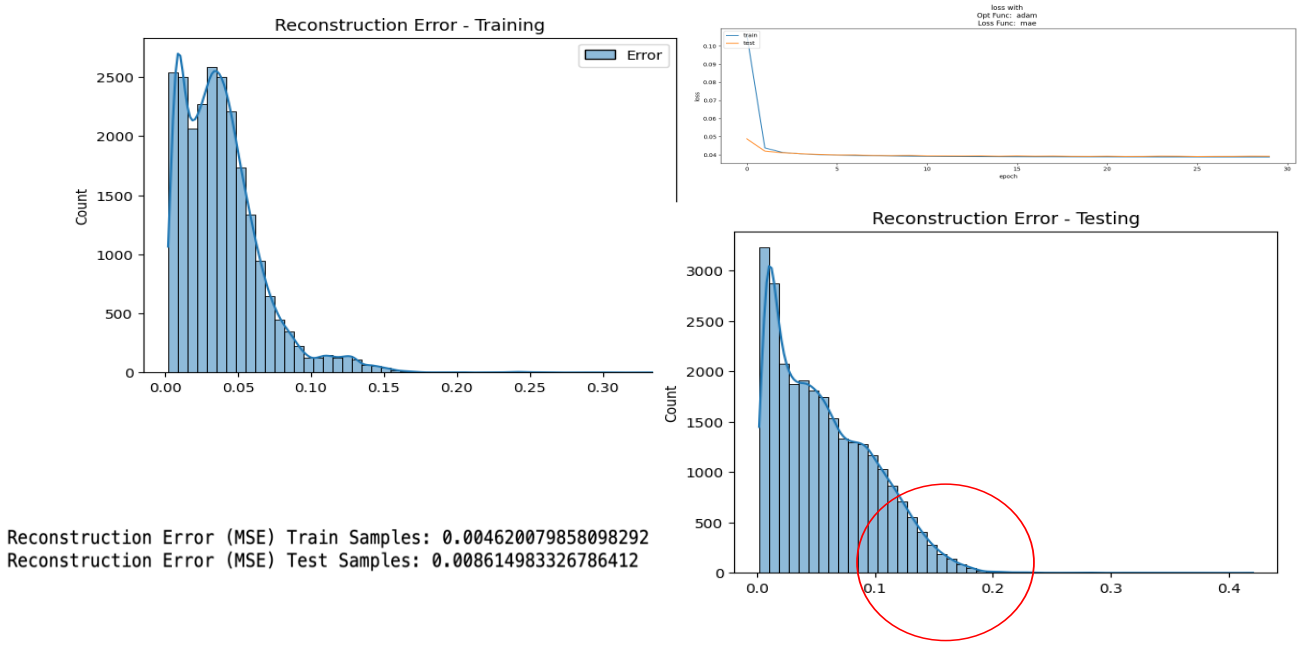

Figure 21: Error distribution plots for training and testing sample distribution.

a) XAI Integration:

Motivated by the above-mentioned observation in Section 4.5 and the emphasis on the importance of explaining the anomalies extracted by an AE-based AD algorithm. It can extract and visually depict the instances and the top contributing features with the highest reconstruction errors. The illustration below provides a framework to explain anomalies identified by an AE without knowing the underlying architecture of the autoencoder model. The force plot method of SHAP is a visual intuitive method better for understanding the anomalies. Hence, making the complex black box models more interpretable. SHAP provides a force plot to visually explain the features that contribute the most to give an insight into the output.

ALGORITHM 1: AUTOENCODER ANOMALIES INTERPRETATION

Input: trainData – Background instances

f – Autoencoder model

x – Anomalous cell trace to be explained

Output: SHAP values saved in CSV file and Top Contributors Feature list for stakeholders

1. **procedure** AnomaliesInterpretation
2. allFeatures, shapValue, inputValue ← KernelExplainer.getShapValues(f, trainData, x)
3. sortedFeatures ← sortDecrease[allFeatures, shapValue)
4. **for** *i* in sortedFeatures.length: **do**
5. feature ← sortedFeatures[*i*]
6. featureEffect ← feature.shapValue

7. featureValue ← feature.inputValue
8. **if** *featureEffect > 0* **then**
9. print2CSV(featureValue, featureEffect)
10. **endif**
11. topContributor ← sortedFeatures[0]
12. **end for**
13. print Unique topContributor
14. **end procedure**

The KernelExplainer function (Algorithm 1) takes as input the model, an instance of anomalous input x and a set of background instances. For a particular feature x'_i it calculates a set of SHAP values which measures the importance of each of the features x_1, x_2....x_n in predicting x'_i. Here, x'_i is the reconstructed input feature from the trained model. The local interpretability can be represented graphically by using force plots, which enables us to pinpoint the SHAP value of features with respect to each other. Algorithm 1 explains the process of how Shapley values are being interpreted to explain the anomalies using the SHAP output.

The Kernel Explainer Shapley function outputs allFeatures, shapValue and inputValue for an anomalous cell trace. The shapValue corresponds to the impact of each feature on all other features in the cell trace. The features are sorted based on the shapValue in descending order. Then, the positive shapValue features and their corresponding inputValue are stored in the CSV output as Effect(featureEffect) and Value(featureValue) respectively. Lastly, it displays the top unique contributing feature.

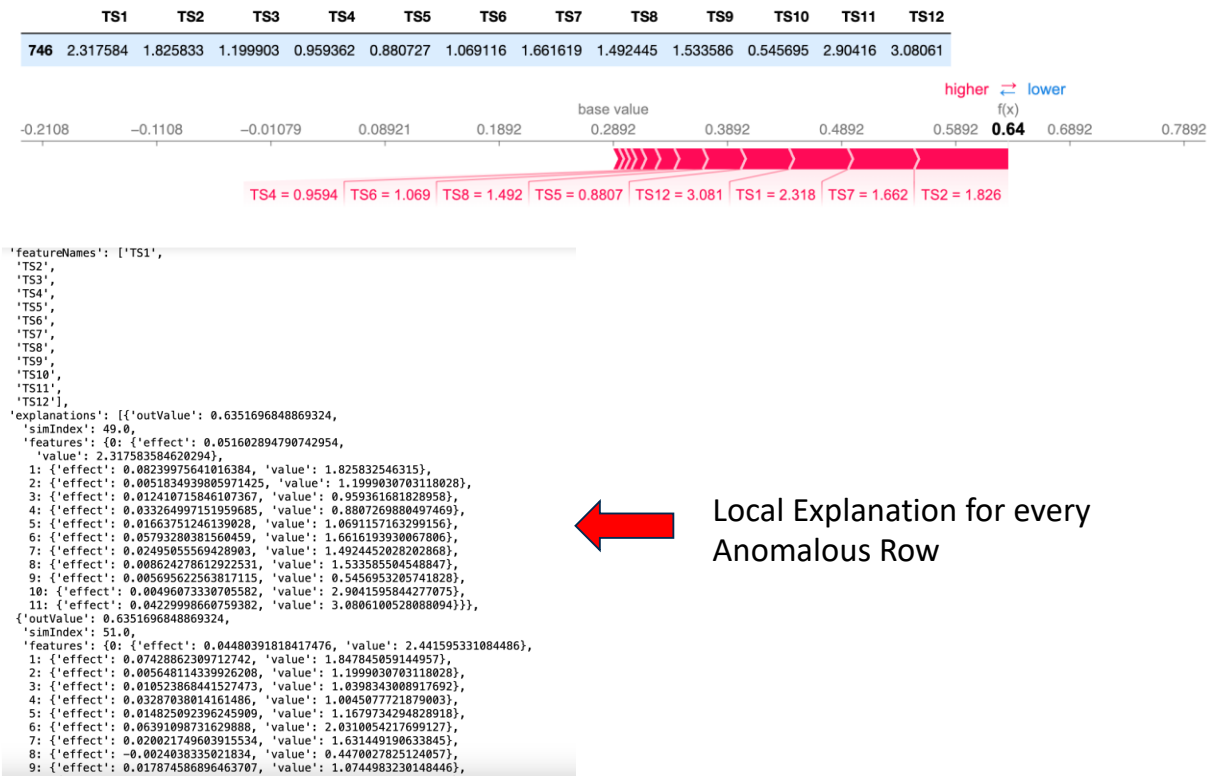

Figure 22: Anomalous instances interpretation using SHAP.

b) SHAP Interpretation:

The interpretable framework provides network operators with actionable insights for every anomalous

network session, which enables a deeper investigation of influencing features. The unsupervised approach assists the analyst in identifying anomalies, categorizing them, and identifying their causes.

4.5.2 Generative model for Anomaly Detection with reconstruction error based Visualised Explanations

One of the approaches currently deployed and tested for XAD at the 6G-BRICKS DMO is combining a SotA generative AD model with a module that provides simple yet comprehensive explanations through the reconstruction error of the features of the detected anomalies.

The model that performs the AD is an implementation of the approach presented in [54] that has been finetuned and applied to networking data. This novel approach utilizes the Alternating Back Propagation (ABP) algorithm presented in [55] for training a generator network that reconstructs data from a latent space and short run Markov Chain Monte Carlo (MCMC) [56] for estimating the posterior of latent variables and inferring latent vectors. To map the complex temporal dependencies of the input data, hierarchical latent factors are used, as explained below.

Hierarchical Latent Factors: Let ***Y*** **be the input of the model, a time-series window of size** s_w with ***m*** features. The window is divided into sub-windows of size s_w / a_l , where $a = [a_1 \dots \dots . a_L]$ is the hierarchy structure, $\boldsymbol{L}$ is the number of levels and a_L is the **number of consecutive sub-windows that share the same latent vector at level** $\boldsymbol{L}$. The model for each sub-window Y_j is given by:

$$s_j = F_a(z^1_{\lfloor \frac{j}{a_1} \rfloor}, \dots, z^L_{\lfloor \frac{j}{a_L} \rfloor}) \qquad (12)$$
$$Y_j = G_\beta(s_j) + e_j$$

$s_j = F_a(z^1_{\lfloor \frac{j}{a_1} \rfloor}$ where F_a is the state model, G_β is the generator model, $\theta = [\alpha, \beta]$ are the parameters, s_j is the state vector and e_j is a vector drawn from the normal distribution. Then:

$$Z = \{z^l_{\lfloor \frac{j}{A_l} \rfloor} \in \mathbb{R}^{d_l}\}_{l,j} \qquad (13)$$

$Z = \{z^l_{\lfloor \frac{j}{A_l} \rfloor} \in \mathbb{R}^{d_l}\}_{l,j}$ Where ***Z*** is the hierarchical latent factor space for ***Y***. The generator is a CNN which maps a latent vector ***Z*** to a time-series window and the state model is a concatenation layer. The idea of the hierarchical latent space involves harnessing the dynamics within time-series, to efficiently encode information in the latent space using lower-dimensional vectors. The hierarchical structure can either be fine-tuned as hyper-parameters or predefined based on domain expertise, aligning with known patterns in the time-series. The joint inference of the hierarchical latent space ***Z*** is achieved through Langevin Dynamics. The proportionate size between the lower-level state vector and the upper levels serves as a control mechanism for the model's flexibility. Larger lower hierarchy level vectors enhance flexibility, rendering the model robust to normal variations or randomness in the long-term dependencies of the time-series, thereby reducing false positives by minimizing reconstruction errors. Conversely, larger tied vectors enforce a stricter model, enhancing its ability to detect contextual anomalies.

Alternating Back Propagation (ABP): We use the ABP algorithm for the model to learn the parameters $\boldsymbol{\vartheta}$. After dividing the input time-series to a training set of windows $Y^{(i)}$, the algorithm learns $\boldsymbol{\vartheta}$ by maximizing the observed log-likelihood directly, which is:

$$L(\theta) = \sum_{i=1}^{n} \log p_\Theta\left(Y^{(i)}\right) = \sum_{i=1}^{n} \log \int p_\Theta\left(Y^{(i)}, Z^{(i)}\right) dZ^{(i)} \tag{14}$$

The gradients for $L'(\theta)$ can be simplified as:

$$\frac{\partial}{\partial\theta} \log p_\Theta\left(Y^{(i)}\right) = \frac{1}{p_\theta(Y^{(i)})} \frac{\partial}{\partial\Theta} \int p_\Theta\left(Y^{(i)}, Z^{(i)}\right) dz = \mathbb{E}_{P_\theta(Z^{(i)}|Y^{(i)})}\left[\frac{\partial}{\partial\Theta} \log p_\Theta\left(Y^{(i)}, Z^{(i)}\right)\right] \tag{15}$$

Where $p_\theta(Z^{(i)}\left|Y^{(i)}\right) = p_\theta(Y^{(i)}, Z^{(i)}) / p_\theta((Y^{(i)})$ is the posterior. The expectation in the equation above can be approximated with a MCMC method, specifically Langevin Dynamics in this case, which iterates:

$$\begin{aligned} Z_{t+1}^{(i)} &= Z_t^{(i)} + \frac{s}{\sigma_z} \frac{\partial}{\partial Z^{(i)}} \log p_\theta\left(Z_t^{(i)} | Y^{(i)}\right) + \sqrt{2s}\, \epsilon_t \\ &= Z_t^{(i)} + \sqrt{2s}\, \epsilon_t + \frac{s}{\sigma_z}\left[(Y^{(i)} - f(Z_t^{(i)}, \theta)) \frac{\partial}{\partial Z^{(i)}} f(Z_t^{(i)}, \theta) - Y_t^{(i)}\right] \end{aligned} \tag{16}$$

where ***t*** is the time step of the dynamics, ***s*** is the step size, and σ_z controls the relative size of the injected noise. Latent factors are chosen such that the current residual on the reconstruction, $Y^{(i)} - f\left(Z_t^{(i)}, \theta\right)$, is minimized. The iterative process is truncated to a predefined number of iterations, and the rejection step is not considered. For an observation $Y^{(i)}$ the resulting vector is a sample from an approximated posterior $p(\theta)\left(Z^{(i)}\ given\ Y^{(i)}\right)$. The MCMC approximation for the gradient is computed as:

$$L'(\theta) \approx \frac{\partial}{\partial\theta} \log p_\Theta\left(Z^{(i)}, Y^{(i)}\right) = \frac{1}{\sigma^2}\left(Y^{(i)} - f(Z^{(i)}, \theta)\right) \frac{\partial}{\partial\theta} f\left(Z^{(i)}, \theta\right) \tag{17}$$

Algorithm 1 presents ABP with mini batches. The algorithm iterates two steps: (1) Inferential backpropagation and (2) Learning backpropagation. In Eq. 16 the latent vectors $\{Z^{(i)}\}$ are inferred for a sample$\{Y^{(i)}\}$ and in Eq. 17 $\{Z^{(i)}\}$ are used as input of the generator model and parameters θ are updated with stochastic gradient descent.

Algorithm 1: Alternating Backpropagation with mini batches

Input: *Training examples* $\{\boldsymbol{Y^{(i)}}, \boldsymbol{i = 1, \ldots . n}\}$
Langevin steps l, learning iterations Iter
Output: *Parameters ϑ learned after training the model*
Latent Vectors $\{\boldsymbol{Z^{(i)}}, \boldsymbol{i = 1, \ldots . n}\}$

1 *Let* $\boldsymbol{t = 0, initialize\ \theta}$
2 *Initialize* $\boldsymbol{Z^{(i)}\ for\ i = 1, \ldots n}$
3 ***While t < Iter:***
4 *Choose a mini batch* $\{\boldsymbol{Y^{(j)}, j = 1, \ldots . b}\}$
5 ***Inferential Backpropagation:*** *For each j run l steps of Langevin Dynamics to sample* $\boldsymbol{Z^{(j)} \sim p\left(Z|Y^{(j)}, \theta\right)}$ *according to Equation 16*
6 ***Learning Backpropagation:*** *Compute the approximation of the gradients according to Equation 17 and update parameters ϑ*
7 *t = t + 1*
8 ***End***

The approach described in this section was selected over other SoTA models, as it presents advantages that are beneficial for its application in the networking domain, and specifically in 6G-BRICKS. Specifically, it produces similar or better detection metrics in the datasets which have been tested and is computationally efficient with lower training times than RNN-based models., This of utmost importance in the DMO setting where AD must be performed in real-time and with low detection time. The model has also been enhanced with a hyperparameter grid search, to finetune the parameters for the hierarchical latent factors, Langevin dynamics and ABP for any dataset that is used.

When the training phase is completed and the parameters of the model have been calculated, we proceed to the testing phase. The testing data are again separated into windows, the latent vectors are inferred via MCMC, and the latent representation Z is the input for the now trained generator network. The generator reconstructs the input windows and the MSE metric is used as the anomaly score/reconstruction error for each time step of each window:

$$score_t = \frac{1}{m}\sum_{i=1}^{m}(Y_{i,t} - \widehat{Y_{i,t}})^2 \quad (18)$$

where m is the number of features, Y is the input window and $\hat{Y}$ is the reconstructed window. Then, if the score exceeds a predefined threshold, the sample is flagged as an anomaly.

Reconstruction Error-based explanations

Reconstruction-based models for AD are considered to have a level of inherent interpretability. When an anomaly is detected, the reconstruction error of each of the features can be computed, with the intuition that features with a higher reconstruction error are the ones where the anomaly is mainly present.

The explainability module presented in this section visualizes the anomalies detected by the system in each time step, ranks the reconstruction error of the features in anomalous regions and returns the features most relevant to the anomaly as well as quantified metrics to indicate how many of the features were identified correctly. The module is not specific to our generative model but can be used with any reconstruction model. The main advantage of this approach is its time efficiency and very low computational cost. In the DMO, most explanations will have to be provided in a time-sensitive setting and other popular XAI techniques like SHAP and LIME have high computation times.

The explainability module completes the following steps:

1) Extract reconstruction errors of each feature of each time series sample from the generative model.
2) Create a list of anomalous ranges where the anomalies are detected. Most reconstruction-based models do not identify all time points of an anomaly, but only a part of said anomaly. With this in mind and considering an unsupervised scenario (since it is not possible to label all the input data from the DMO); we set a user-defined parameter that expands an anomaly around the detected point and can be finetuned according to past anomalies. For instance, if an anomaly is detected at time step 100 and we define a range of 25, the anomalous range will be time steps 75-125).
3) Set a reconstruction error percentage threshold (T_r), which is required to define the top features with the highest reconstruction errors, that contribute to (and are the possible causes of) each anomaly. When detecting an anomaly, the total reconstruction error is the sum of the reconstruction errors of all features, including multiple features will very low errors that are not possible contributors to the fault. To filter them out, only the features that make up a specified percentage of the total error are reported.

© 6G-BRICKS 2023

This percentage is defined by T_r. If T_r is set to 0.8, the module will only report the features that make up 80% of the total reconstruction error.

4) Retrieve the explanations, which are the top features with the highest reconstruction error for each anomaly.
5) Compute quantifiable metrics to evaluate the explanations: We calculate the numbers of True Positive (TP) features, indicating the number of features reported as anomalous by the model that are indeed (ground truth) anomalous (true root causes of the anomaly), False Positive (FP) features that were reported as anomalous while not being causes of the anomaly, and False Negative (FN) features which are true causes of the anomaly but were missed by the algorithm and not reported.
   We then calculate the metrics $Explainability_{precision}$, $Explainability_{recall}$, $Explainability_{f1-score}$ which are metrics equivalent to the common precision, recall and f1-score used in ML, but adjusted for quantifying the quality of the explanations. They are calculated as follows:

$$Explainability_{precision} = \frac{TP}{TP + FP} \quad (19)$$

$$Explainability_{recall} = \frac{TP}{TP + FN} \quad (20)$$

$$Explainability_{f1-score} = \frac{2TP}{2TP + FP + FN} \quad (21)$$

6) Provide visualization plots with the detected anomalies, along with the top features as explanations and the computed evaluation metrics.

4.5.3 Combining NWDAF with AI for Abnormal Traffic Detection

The DMO will gather information from diverse components across different technological domains. It interfaces with NWDAF for Core Network (CN) data, with the KPM xApps for Radio Access Network data, and with the NFVO Collectors for NFVO data. The DMO leverages AI/ML approaches to detect abnormal traffic, employing algorithms such as AEs. On the other hand, in adherence to 3GPP standards, NWDAF stands out for its ability to provide the probability of abnormal traffic. To streamline the process, ML-based approaches can be embedded within NWDAF to calculate this probability. Therefore, we can use this NWDAF UC to have the probabilities of abnormal traffic in the DMO, i.e., the DMO will subscribe to the event that delivers UE abnormal traffic probability from NWDAF, then it will receive periodically probabilities about abnormal traffic that will be forwarded directly to the root cause analysis in the fault management component. Below, we provide an overview of NWDAF. Then, we present the functional property of the AE-based abnormal traffic detector engine in NWDAF.

4.5.3.1 NWDAF Overview

NWDAF is a key component of the 5G CN architecture, standardized by 3GPP [57]. Its role is to generate analytics and insights from network data to accommodate end-users and improve network performance. NWDAF allows the collection, processing, and analysis of network data to enable various applications, such as UE mobility analytics and abnormal UE behavior. Figure 23 illustrates the NWDAF Network Function as defined by 3GPP. In the figure, NWDAF collects 5G-related data from other network functions and then delivers services (UCs) to NWDAF clients (consumers) either in real-time or via subscription/notification.

© 6G-BRICKS 2023

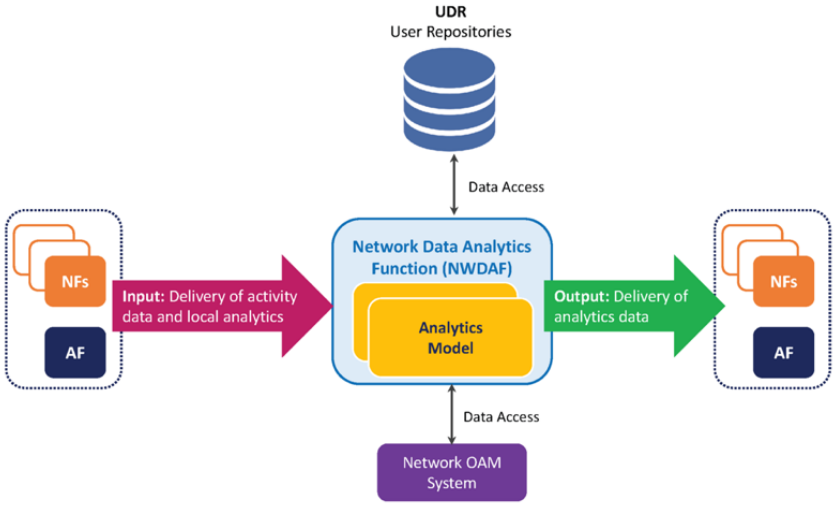

Figure 23: The Network Data Analytic Function (NWDAF) [58].

As mentioned earlier, one of the uses defined by 3GPP is the detection of abnormal traffic for a set of UEs. Therefore, NWDAF clients can subscribe to this event using event ID "abnormal behavior" and exception ID "unexpected large rate flow" and receive notifications containing probabilities of abnormal UE traffic. At EURECOM, the architecture of NWDAF is illustrated in Figure 24 [59]. It comprises three layers: (i) Exposure, (ii) Analytics, and (iii) Monitoring. The Exposure layer serves as the system's entry point, where clients can submit requests following 3GPP specifications. This layer requires computed data, such as abnormal traffic probabilities, and it requests this information from the Analytics layer. The Analytics layer performs computing tasks to derive the requested information, fetching data from the database in the Monitoring layer. The Monitoring layer collects data from the infrastructure and stores it in a database. In this architecture, each UC has its own service. Therefore, we designed an AD engine responsible for calculating abnormal traffic probabilities for a given UE. This engine utilizes the data collected by the NWDAF Southbound Interface (SBI) in the database. This allows NWDAF to have a service exposed to obtain abnormal traffic data, and the DMO will only need to subscribe to this event to receive the relevant information.

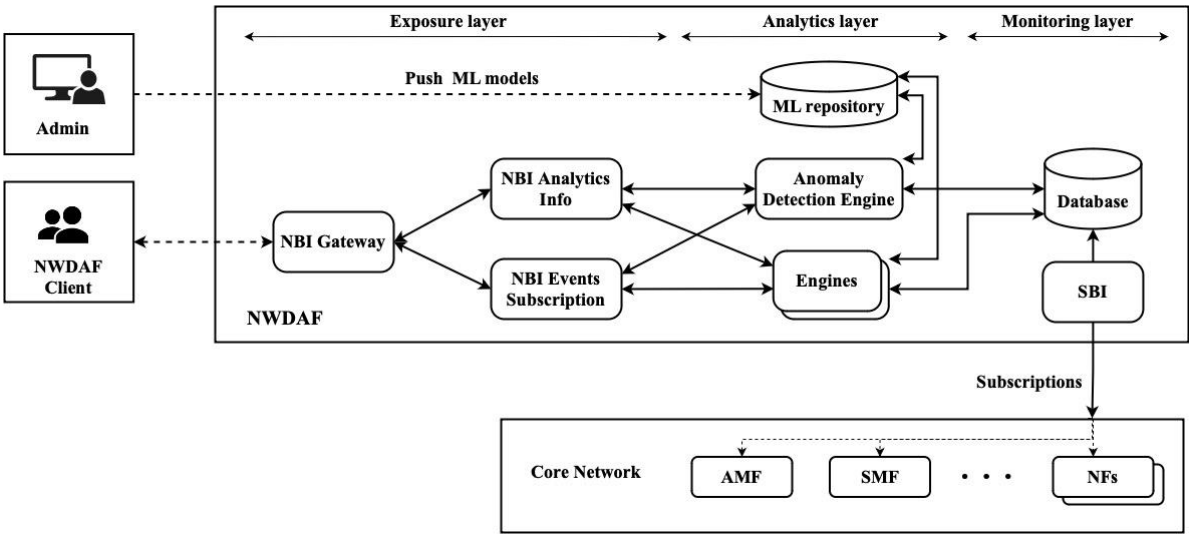

Figure 24: Internal NWDAF architecture [59].

4.5.3.2 NWDAF AE-based Abnormal traffic detector

The NWDAF clients can subscribe to the "abnormal_behavior" event to receive periodic updates on the probability of abnormal traffic. The latter is computed by the Anomaly Detection Engine leveraging an LSTM-based AE. The LSTM-based AE learns the traffic pattern using the history of the UE data measured from the User Plane Function (UPF) in both uplink and downlink directions. The traffic pattern contains information such as the data size in bytes at a given timestamp during a week. Figure 25 depicts a basic illustration of the LSTM AE model architecture.

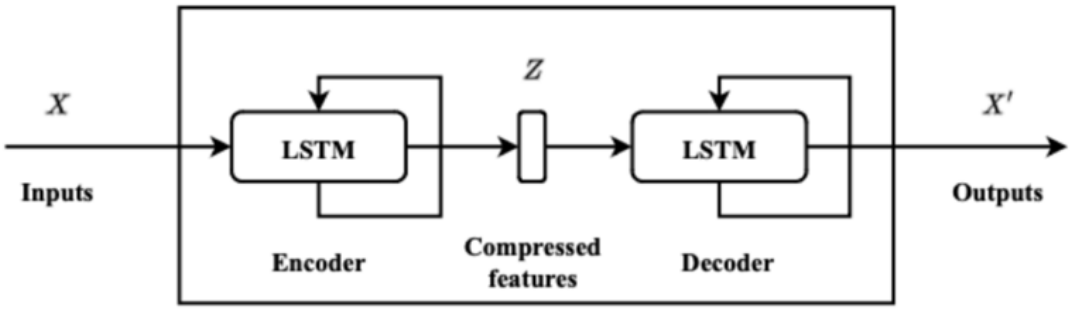

Figure 25: LSTM Autoencoder architecture.

The AE architecture's motivation is that it reduces the variance between input and output by training only on normal traffic, without considering abnormal traffic; whereas the motivation behind using the LSTM is driven by the nature of the input, which is correlated with time (e.g., traffic during the night is different from traffic during the day), and the LSTM is well-known for dealing with the vanishing gradient problem, which makes the training more stable. In equation 22, δ represents the Mean Absolute Error (MAE) between the input and output of the AE.

$$\delta = MAE = \frac{\sum_{i=1}^{n}|x_i' - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n} \quad (22)$$

Let β denote the average MAE over the training data and β' the MAE of the inference data. Equation 23 calculates the traffic anomaly probability p using the difference between β and β'.

$$p = \min(\alpha \times |\beta' - \beta|, 1) \quad (23)$$

Where α is a weight to control the impact of the distances scale, i.e., when α is small enough, the distances will have less impact on the probability.

4.5.4 LLM-enabled Vertical Trial Fault Root Cause Analysis using XAI

In the realm of cloud-native infrastructure, deploying application sets for trials is a straightforward process for vertical users. However, the susceptibility of these applications to performance degradation, influenced by factors such as fluctuating resource needs and unforeseen conditions, poses a significant challenge. The lack of comprehension regarding these underlying causes not only disrupts users from interpreting trial results effectively but also obstructs their ability to enhance and refine their experimental setups. Therefore, the integration of an intelligent management system emerges as a pivotal solution to address these challenges. This system will guide users through the trial lifecycle by pinpointing performance degradations within the deployed applications, acting as a critical diagnostic tool, and unraveling potential issues. More than just identifying

problems, the intelligent management system goes a step further by offering concrete and actionable solutions for improvement. A high-level overview is illustrated in Figure 26.

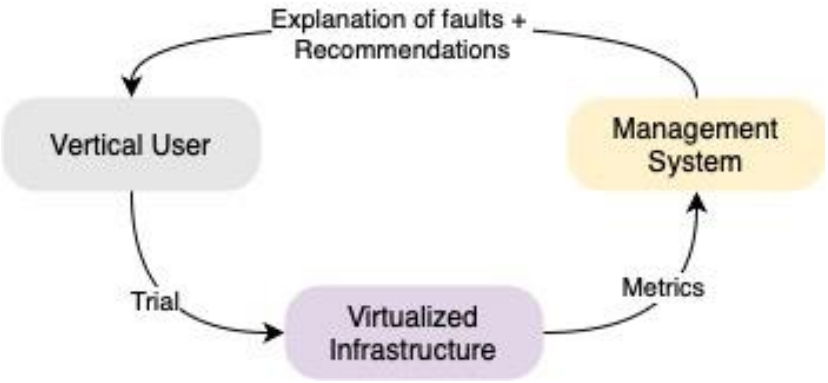

Figure 26: Trial explainability overview.

The SoTA includes many AI-based approaches that predict faults in these applications, such as [60]. However, these AI models typically only predict the occurrence of degradation, not the root cause. Understanding the root cause is crucial for effective intervention. Therefore, the research community is extensively using XAI approaches to interpret "black box" AI models and facilitate root cause analysis [61]. However, simply presenting XAI output values to non-experts, such as vertical users who may not have specific technical knowledge, may not provide them with enough information to understand the root cause of the problem. Fortunately, with the recent advancements in Natural Language Processing (NLP) LLMs offer a promising solution. LLMs can generate human-like text that is understandable by all users and can be translated into different languages [62]. This makes LLMs an attractive choice for designing a management system that could bridge the gap between AI models and non-technical users.

4.5.5 System design

Figure 27 illustrates the comprehensive architecture design of the system, showcasing the seamless integration of XAI and LLMs to deliver user-friendly insights and solutions. In this proposed system, our approach relies on ML models to predict the degradation of performance in vertical applications. Upon detection of service degradation in the Monitoring System, XAI algorithms are employed to interpret the ML predictions and identify the features contributing to poor performance.

Specifically, our approach utilizes an ML algorithm based on XGBoost [63] to predict violations related to the performance of running applications. The application response time metric is used to characterize application performance. The trained ML model considers various features related to CPU and memory, including CPU usage, CPU limit, memory usage, and memory limit. Simultaneously, an XAI algorithm, namely SHAP [40], is deployed to deduce the most important features influencing such violations based on ML outputs. Subsequently, the output from the XAI algorithm and the prediction from the AI model are transmitted to the LLM. The LLM processes the numerical values and generates a human-language explanation of the root cause, providing the underlying reasoning for the prediction to the experimenter through the Vertical portal. Moreover, the LLM has the capability to suggest recommendations and actions for the vertical user to enhance trial efficiency.

© 6G-BRICKS 2023

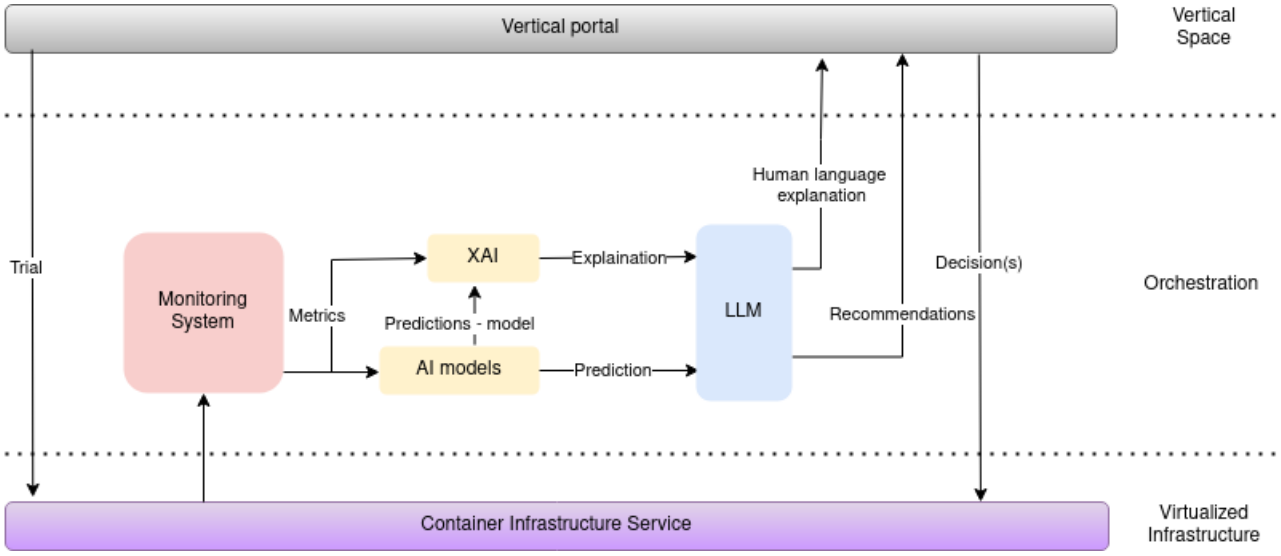

Figure 27: The framework architecture.

5. Near-RT RIC Framework for CF mMIMO Radio Resource Management via O-RAN compliant xApps

The E2 interface the specifications from O-RAN ALLIANCE provide newest specification, the main functions/procedures supported over E2, the detailed signaling procedures between the two endpoints. The E2 Service Model (E2SM), which defines how specific functions located in the O-RAN Node can be controlled by the Near-RT RIC over E2. The E2 interface should support service models, Key Performance Measurements (KPM) RAN Control (RC). The open Radio Access Networks (RAN) paradigm transforms the RAN towards virtualized, fully interoperable and intelligent architecture. The open RAN brings flexibility and intelligent control opportunities which are necessary for CF mMIMO architectures. To exploit the O-RAN entities optimally and meet the diverse requirements of 6G will provide enhanced flexibility through the dynamic scheduling of the available resources. Given the densification of networks, and the introduction of cell-free mMIMO architecture, the availability of RAN resources is unseen, and is only limited by the potential of the resource allocation methods. Several research works are already being conducted on the way to map CF mMIMO in O-RAN architecture. In [64], a connection between the handover and clustering solutions strategies with O-RAN architecture is established that offers open interfaces and network-wide control capabilities. Similarly, the authors in [65] mapped the clustering and the uplink detection method to the O-RAN architecture. The authors further identify what is currently possible, and we propose the needed extensions to O-RAN to fully exploit the SoTA cell-free processing schemes. The main goal of the research in [66] is to introduce how the CF mMIMO architecture links to the O-RAN-specified architecture.

In 6G-BRICKS, we are further translating the Cell-Free to O-RAN architecture by extending the E2 interface to support CF mMIMO related xApps (e.g., trajectory prediction and multi-band clustering) that are not provided in [65][66]. We further offer CF experimenter xApps to the experimenters to configure the RAN elements (CU, DU, MAC scheduler) via the Near-RT RIC framework. The Near-RT RIC communicates with the CF mMIMO RAN system over the E2 interface and configures underlying E2 nodes. In this deliverable, section 5.2 describes xApp messaging infrastructure of the trajectory prediction xApp for Cell-free mMIMO and section 5.3 provides the messaging infrastructure of the Multi-band clustering xApp that can cluster the O-RUs in different bands.

5.1. Near-RT RIC Framework and Signaling Mechanisms of E2 Nodes

The optimization algorithms, e.g., trajectory prediction in Cell-free mMIMO or multi-band clustering in CF mMIMO, are coded in the xApp and access the RAN and Near-RT RIC information with use APIs, as shown in Figure 28.

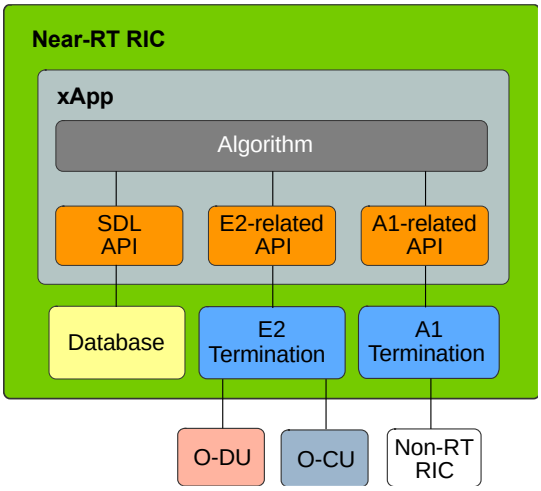

Figure 28: The Near-RT xApp API.

Within each xApp, the following APIs are identified:

- The Shared Data Layer API allows to collect the information about the underlining CF mMIMO system nodes including their types, supported component interfaces and SMs.
- The E2-related API supports the E2 Subscription, E2 Indication, E2 Control and E2 Guidance.
- The A1-related API supports the A1 policy and A1 Enrichment Information.

The E2-related API supports E2 Indication and E2 Control procedures, which are further elaborated below.

5.1.1 E2 Indication procedure

The E2 Indication procedure is used to provide the xApps with the measurements related to the CF mMIMO system. The procedure consists of E2 INDICATION message shown in Figure 29.

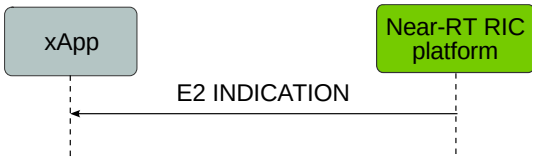

Figure 29: E2 INDICATION message.

The E2 INDICATION message is coded with JSON with the following format:

```
{“message_type”:{“procedure_type”: “E2_indication”, “type_of_message”: “request”},
“GlobalE2node-ID":{“GlobalgNB-ID”: 1, “gNB-DU-ID”: 1},
“e2_request_id”:INTEGER,
“ran_function_id”:4000,
“e2_action_id”:INTEGER,
“e2_indication_sn”:INTEGER,
“e2_indication_type”:“Report”,
“e2_indication_header”:{“ue_id”: 3},
“e2_indication_message”:OCTET STRING}}
```

The “ran_function_id” is set to 4000, indicating the Cell-free mMIMO SM.

The “e2_indication_message” is specific to the UC and is described in the following section 5.2.

5.1.2 E2 control procedure

The E2 control procedure is used by the xApp to send a E2 CONTROL (Request) message concerning clustering, as depicted in Figure 30:

- If the control message is successfully received, then the Near-RT RIC platform responds with the E2 CONTROL message (indicative of success).
- If the control request is rejected by the Near-RT platform, then it replies with the E2CONTROL message (indicative ofrReject).
- If the control request is accepted by the Near-RT RIC platform, but has failed in the E2AP RIC Control procedure, the Near-RT RIC platform responds with the E2 CONTROL message (indicating a failure) and provides a cause for such failure.

© 6G-BRICKS 2023

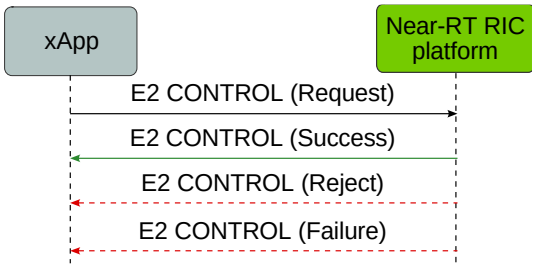

Figure 30: The E2 CONTROL message.

```
{“message_type”:{“procedure_type”:},
“procedure_transaction_ID”:
“GlobalE2node-ID":{“GlobalgNB-ID”:1,“gNB-DU-ID”:1},
“ran_function_id”:4000,
“e2_control_header”:{“ue_id”:3},
“e2_control_message”:OCTET STRING}}
```

The E2 Indication and E2 Control procedures are used in the xApp control loop, as shown in Figure 31. The E2 INDICATION message is sent to the xApp periodically to provide the inputs to the xApp algorithm. Whenever the modification in the RAN is necessary, the xApp sends the E2 CONTROL request message.

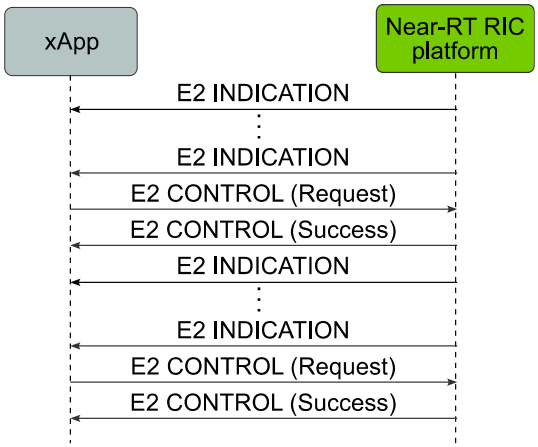

Figure 31: The xApp control loop.

5.2. The RIC xApp messaging infrastructure of trajectory prediction in Cell-free mMIMO

The CF mMIMO trajectory prediction xApp uses as input the following data, as it can be seen in Figure 32:

- The Received Signal Strength Power (RSRP) vector for each UE with the RSSP by the UE from considered Radio Units (RUs).
- In DwL direction, the O-RUs measured the RSSP by the RUs from the UEs.
- The DwL and UL (uplink) buffer size, i.e., the number of bits waiting at the O-DU buffer for the downlink transmission and the number of bits waiting at the UE buffer respectively.

© 6G-BRICKS 2023

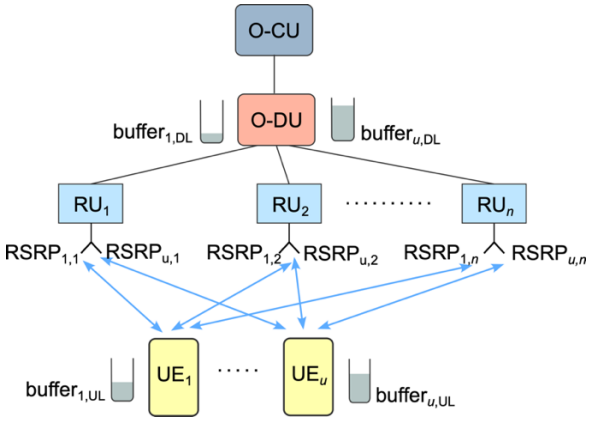

Figure 32: xApp input received in E2 INDICATION.

Therefore, the "e2_indication_message", which was introduced in section 5.1, has the following format coded with JSON:

```
"e2_indication_message": {
"rsrp":{"ru_id":[1, 2, 4, 5], "rsrp_dl":[32,41,12, 30]},
buffer_dl:392,
buffer_ul:465
}
```

In the above message the RSPR values are sent with 1 dB resolution.

After the xApp receives the e2_indication_message it may create the E2 CONTROL message with a list of RUs, which should serve each UE. The E2 INDICATION and E2 CONTROL messages are sent per UE, however, the xApp may wait to send the E2 CONTROL message for a UE until it collects the E2 INDICATION messages from more or all UEs. For example, the xApp may choose the RU1, RU2 and RU5 to serve the UE1 for the downlink transmission, and RU2 and RU3 to serve the UE2 for the downlink transmission. In this case, the xApp sends the E2 CONTROL (Request) message with JSON coding in the following format:

Control message for UE1: "ru_id_cluster_dl": [1, 2, 5].

Control message for UE2: "ru_id_cluster_dl": [2, 3].

There could be the same or different cluster used for downlink and uplink transmission.

It should also be noted that the E2 INDICATION and E2 CONTROL messages are sent per UE. However, the xApp may wait to send the E2 CONTROL message for a UE until collecting the E2 INDICATION messages from more UEs or all of them.

5.3. The RIC xApp messaging infrastructure of multi-band clustering in Cell-free mMIMO

Similarly to the Trajectory prediction xApp, the Multi-band clustering xApp also collects the inputs in the form of the RSRP and buffer size to determine the serving cluster, but the cluster is determined for each band. For that reason, the RSRP inputs are collected for RUs of each frequency band as shown in Figure 33.

Therefore, the "e2_indication_message" is extended by the "band" information element:

```
“e2_indication_message”:{
“rsrp”:{“band”:1, “ru_id”:[1, 2], “rsrp_dl”:[12, 23]}},
“rsrp”:{“band”:2, “ru_id”:[4, 5], “rsrp_dl”:[32, 41]},
“buffer_dl”:392,
“buffer_ul”:465
}
```

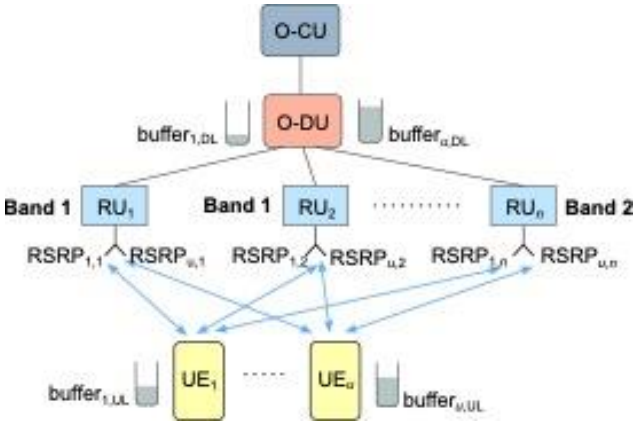

Figure 33: xApp inputs received in E2 INDICATION supporting multi-band clustering.

In the case of Multi-band clustering xApp, the control action determines the service cluster for each band. Thus, the control message format has an additional “band” information element:

```
{“band”: 1, “ru_id_cluster_dl”: [1, 2]}, {“band”: 2, “ru_id_cluster_dl”:[3, 4]}.
```

In the above message, the xApp requests the CF mMIMO RAN to serve the UE with RU1 and RU2 in Band 1, while RU4 and RU5 should serve the UE in Band 2.

6. Near-RT RIC Framework for Reflected Intelligent Surface (RIS) control via xApps

In D3.2, we detailed the RIS control framework enabler. The latter leverages O-RAN RIC to control the RIS. First, we explain how this enabler is exposed to the experimenter via the 6G-BRICKS platform. Then, we explain the different SMs used in the RIS control framework (i.e., KPM E2SM and RIS E2SM).

6.1 RIS control framework Exposure innovation

The experimenter has a single entry-point to our experimentation platform (i.e. the 6G-BRICKS Web Portal), as depicted in Figure 34. The developer provides the GitLab link of any given RIS control algorithm source code. The DMO will build a Network Service Descriptor (NSD) using the input provided by the developer and send it to the facility orchestration platform at EURECOM. The facility components will pull the source code and build the image if the GitLab repository is provided in the NSD. Then, the orchestration platform will instantiate the xApp on top of EURECOM's virtual infrastructure based on Kubernetes. The xApp is automatically connected to FlexRIC and hence can send and receive E2 messages to the available E2 Nodes such as the RIS and OAI's DU.

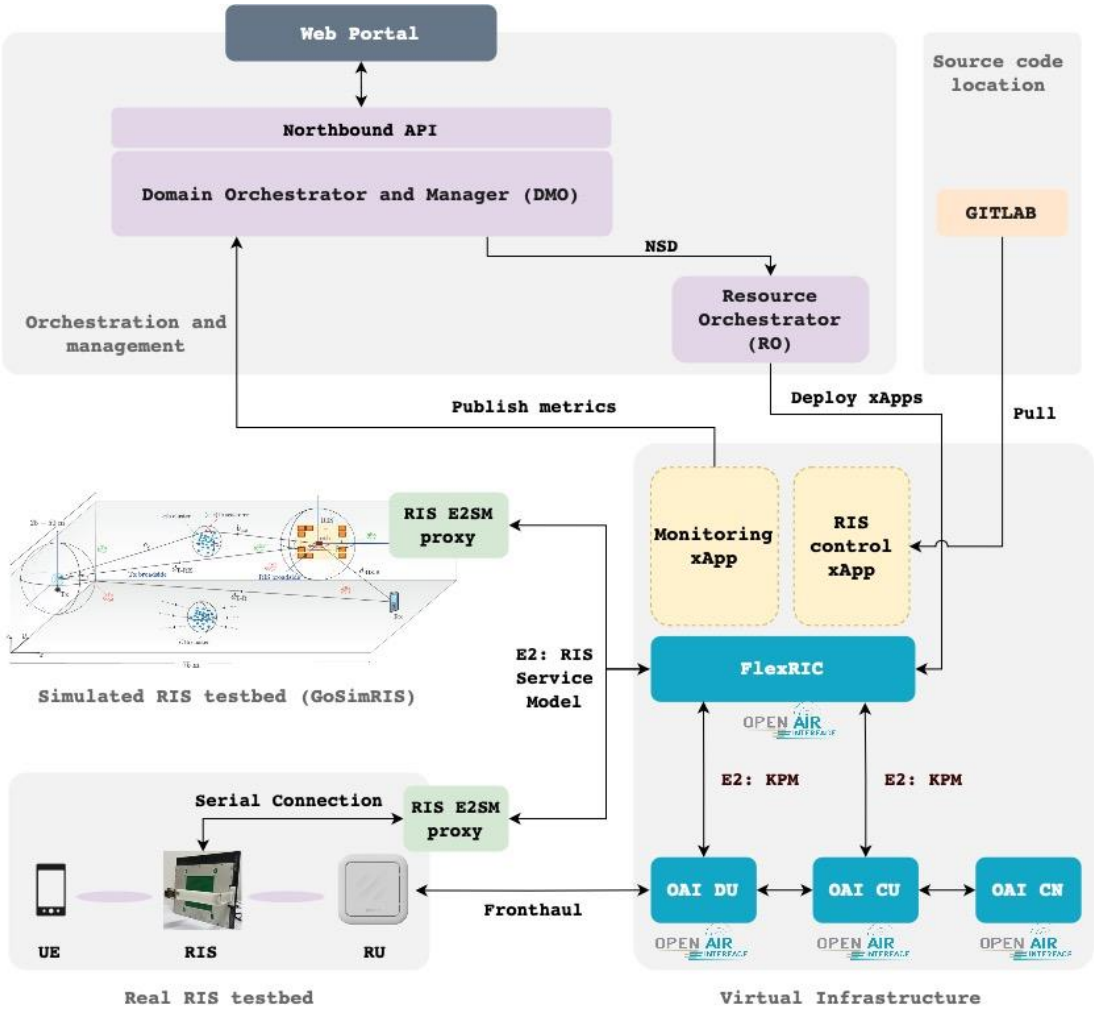

Figure 34: RIS platform exposure via 6G-BRICKS platform.

As per the SMs, the DMO and the RIS control xApps monitor the RAN performance using the KPM SM from the DU. The KPM SM includes UE's channel conditions metrics such as SNR, RSRP and BLER. The RIS control algorithms can leverage these metrics to compute the RIS coefficients optimally. On the other hand, to send the control message to the RIS, the developer leverages the RIS SM. Both KPM and RIS SMs are abstracted via FlexRIC SDK. The latter allows the developer to focus on his application logic instead of O-RAN internal details (e.g., the E2 communication and data storage). Details on each SM are provided in the next sections.

6.2 xApp Monitoring and KPM Service Model related to RIS

KPM is one of the key SMs introduced by the O-RAN alliance specifications. It exposes the available measurements from the E2 nodes including the OAI's CU and DU via RAN Functions.

The measurements can be: (i) node-specific, common between all the UEs connected to the same node; (ii) UE-specific, dedicated to a UE; or (iii) bearer-specific, dedicated to a particular bearer inside a particular UE. The KPM SM allows the developer to write applications, called xApps, to collect these measurements from the E2 nodes.

In the RIS control context, xApps use KPM to monitor the channel information of the UEs. The KPM SM exposes different channel metrics such as RSRP and SNR. We also extend KPM to collect UE's position leveraging the UWB enabler (detailed in D3.1).

The KPM SM in FlexRIC is exposed as an SDK. The experimenter will not see E2 messages but instead as an object, in the chosen programming language (e.g., Python).

6.3 RIS E2SM

To abstract the interaction with the RIS, we introduce the RIS E2SM. The RIS E2SM defines two types of information: the (i) E2 indication messages and the (ii) E2 control messages.

1) The E2 indication messages contain:
    a. The RIS properties from each RIS E2 node include the RIS dimension (length and height), number of RIS elements, spacing between elements, number of RIS element states, position of the RIS, the supported frequency and the list of the available beams (angles). The xApp receives this information from each E2 node and forms a catalogue. The user can choose which RIS to use according to the experiment.
    b. UE's channel metrics: The channel between the RIS and the UE is a cascade channel (i.e., composed of: (i) channel between RIS and UE; (ii) channel between the RIS and gNB; and (iii) direct channel between RIS and UE). The E2 indication message can contain one or many metrics.
2) The E2 control messages contain the RIS configuration (i.e., the RIS elements' coefficients), sent from the xApp to the RIS' E2SM proxy. The configuration can have two abstraction levels:
    a. High-level configuration: the angle of the beam in a 2D plane (an integer value) or two angles in a 3D plane (two integers).
    b. Low-level configuration: the phase distribution for all the RIS elements (list of bytes).

The RIS control flow is detailed in Figure 35 (extracted from D3.2).

© 6G-BRICKS 2023

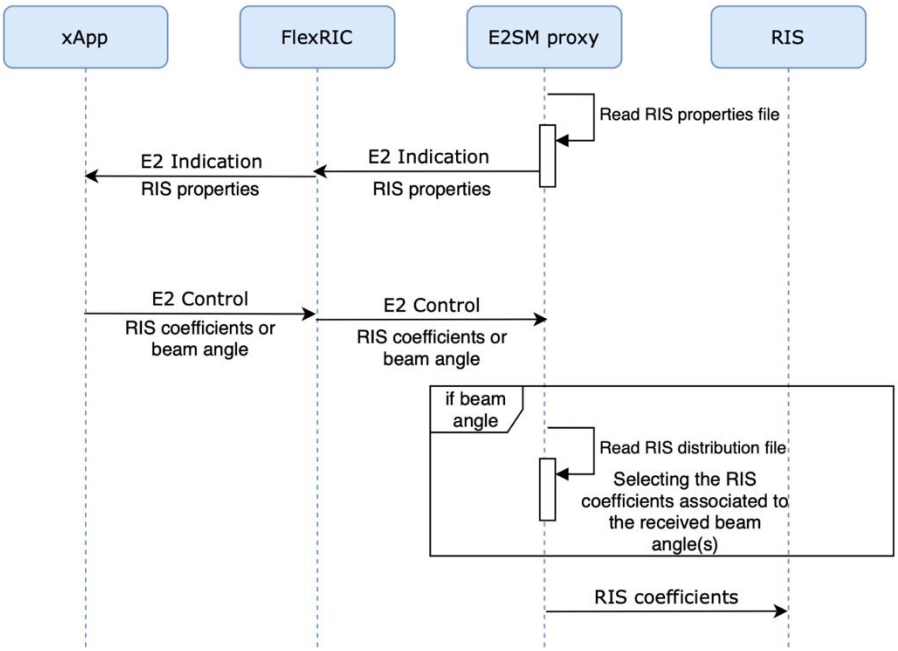

Figure 35: RIS control flow via the E2 interface.

7. Conclusions

This deliverable presented in detail the experimental facility testing framework and the design of the XAI framework for DMO, the models for anomaly detection and XAI integration, a Near-RT RIC Framework for RRM via O-RAN compliant xApps related to CF mMIMO, and the framework that provides the integration of RIS with the Near-RT RIC.

The document is divided into three main technical parts.

The first technical section presented the portal design shown to the 6G-BRICKS experimenters, the intent management engine, and the conflict resolution algorithms. In this context, this deliverable provides a novel admission control algorithm that leverages the short- and long-term correlation of the resource monitoring traffic metrics inside the DMO intent engine to avoid conflicts and maximize admission control efficiency.

The second technical part of this deliverable described the XAI approach followed for DMO and the AD process. In this regard, the document described models for anomaly detection and autoencoder with SHAP integration for 6G-BRICKS.

The third technical part of this deliverable describes the Near-RT RIC framework and xApp messaging structure for the xApps developed in 6G-BRICKS that relate to the interaction with the CF mMIMO and the RIS.

© 6G-BRICKS 2023

8. References

[1] TM Forum. Intent Ontology (TIO) – References v3.2.0 (TR292R). Technical Report IG 1218, TM Forum, Aug. 2023, Version 3.2.0.

[2] TM Forum. Autonomous Networks – Business requirements & architecture. Technical Report IG 1218, TM Forum, Jul. 2020, Version 1.0.0.

[3] P. Lingga, J.J. Kim and J.P. Jeong, "Intent-Based Network Management in 6G Core Networks" - *13th International Conference on Information and Communication Technology Convergence (ICTC)* 2022, pp. 760-762.

[4] K. Mehmood, K. Kralevska, and D. Palma, "Intent-driven autonomous network and service management in future cellular networks: A structured literature review", in *Computer Networks*, vol. 220:109477, 2023.

[5] J. Zhang, J. Guo, C. Yang, X. Mi, L. Jiao, X. Zhu, L. Cao, and R. Li, "A Conflict Resolution Scheme in Intent-Driven Network" - *IEEE/CIC International Conference on Communications (ICCC)*, 2021, pp. 23-28.

[6] S. Skaperas, N. Ferdosian, A. Chorti and L. Mamatas, "Scheduling Optimization of Heterogeneous Services by Resolving Conflicts", *arXiv preprint arXiv:2103.01897*, 2021.

[7] H. Che, Y. Ye, M. Xiao, M. Skoglund, and H.V. Poor, "Coded Stochastic ADMM for Decentralized Consensus Optimization with Edge Computing", in *IEEE Internet of Things Journal*, vol. 8, pp. 5360-5373, 2021.

[8] N. Golowich, S. Pattathil and C. Daskalakis, "Tight last-iterate convergence rates for no-regret learning in multi-player games" - *Advances in neural information processing systems*, vol. 33, 2020, pp. 20766-20778

[9] A. Dargazany, "Model-based actor-critic: GAN + DRL (actor-critic) => AGI", *arXiv preprint arXiv:2004.04574*, 2020.

[10] C. Daskalakis and I. Panageas, "Last-Iterate Convergence: Zero-Sum Games and Constrained Min-Max Optimization", *arXiv preprint arXiv:1807.04252*, 2018.

[11] T. Subramanya, D. Harutyunyan and R. Riggio, "Machine Learning-Driven Service Function Chain Placement and Scaling in MEC-enabled 5G Networks", in *Computer Networks*, vol. 166:106980, November 2019, doi: 10.1016/j.comnet.2019.106980.

[12] P. K. Haddadha, M. H. Rezvani, M. Mollamotalebi and A. Shankar, "Machine Learning methods for service placement: a systematic review", in *Artificial Intelligence Review*, vol. 57, no. 3, February 2024, doi: 10.1007/s1046-023-10684-0.

[13] H. Djigal, J. Xu, L. Liu and Y. Zhang, "Machine and Deep Learning for Resource Allocation in Multi-Access Edge Computing: A Survey", in *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2449-2494, August 2022, doi: 10.1109/COMST.2022.3199544.

[14] W. Shi, Q. Ling, G. Wu and W. Yin, "EXTRA: An Exact First-Order Algorithm for Decentralized Consensus Optimization", in *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944-966, 2015.

[15] G. Barlacchi et al. "A multi-source dataset of urban life in the city of Milan and the Province of Trentino." Scientific data 2.1 (2015): 1-15

[16] W. Shi, Q. Ling, K. Yuan, G. Wu and W. Yin, "On the Linear Convergence of the ADMM in Decentralized Consensus Optimization", in *IEEE Transactions on Signal Processing*, vol. 62, pp. 1750-1761, 2013.

[17] A. C. Baktir et al., "Intent-based cognitive closed-loop management with built-in conflict handling" - *IEEE 8th International Conference on Network Softwarization (NetSoft), Milan, Italy*, 2022, pp. 73-78, doi: 10.1109/NetSoft54395.2022.9844074.

[18] L. A. Garrido, K. Ramantas, A. Dalgkitsis, A. Ksentini and C. Verikoukis, "Admission Control with Resource Efficiency Using Reinforcement Learning in Beyond-5G Networks," *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Toronto, ON, Canada, 2023, pp. 1-6, doi: 10.1109/PIMRC56721.2023.10294028.

[19] B. K. Saha, D. Tandur, L. Haab, and L. Podleski, 2018. "Intent-based Networks: An Industrial Perspective", in *Proceedings of the 1st International Workshop on Future Industrial Communication Networks (FICN '18)*, 2018, pp. 35-40.

© 6G-BRICKS 2023

[20] H. Liu, J. Zhou, X. Wu and P. Yuan, "Optimization algorithm for rectangle packing problem based on varied-factor genetic algorithm and lowest front-line strategy" - *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 352-357.
[21]Hennessy, J. L., & Patterson, D. A. (2011). Computer Architecture: A Quantitative Approach.
[22]Mittal, S. (2014). "A survey of techniques for improving energy efficiency in embedded computing systems." ACM Computing Surveys (CSUR), 47(2), 1-36.
[23]D. Samariya and A. Thakkar, "A Comprehensive Survey of Anomaly Detection Algorithms", in *Annals of Data Science*, vol.10, pp. 829–850, 2023, https://doi.org/10.1007/s40745-021-00362-9
[24] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network", in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2019, pp. 2828–2837, https://doi.org/10.1145/3292500.3330672.
[25] F. T. Liu, K. M. Ting and Zhi-Hua Zhou, "Isolation forest - *2008 eighth ieee international conference on data mining*, 2008, pp. 413–422.
[26] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces" - *European conference on principles of data mining and knowledge discovery*, 2002, pp. 15–27.
[27] Z. He, X. Xu and S. Deng, "Discovering cluster-based local outliers", in *Pattern recognition letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.
[28] Z.Z. Darban, G.I. Webb, S. Pan, C.C. Aggarwal, M. Salehi, "Deep Learning for Time Series Anomaly Detection: A Survey", *arXiv preprint arXiv:2211.05244*, 2022.
[29] C. Peng, Z. Tang, W. Gui, Q. Chen, J. He, "A bidirectional weighted boundary distance algorithm for time series similarity computation based on optimized sliding window size", in *Journal of Industrial & Management Optimization*, vol 17, no. 1, 2021.
[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets" - *Advances in neural information processing systems*, vol. 27, 2014.
[31] X. Chen, L. Deng, F. Huang, C. Zhang, Z. Zhang, Y. Zhao and K. Zheng, "Daemon: Unsupervised anomaly detection and interpretation for multivariate time series" - *IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 2225–2230.
[32] D. Li, D. Chen, B. Jin, L. Shi, J. Goh and S. K. Ng., "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks" - *International conference on artificial neural networks*, Springer pp. 703–716, 2019.
[33] S. Tuli, G. Casale and N. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data", *arXiv preprint arXiv:2201.07284*, 2022.
[34] Y. Li, X. Peng, J. Zhang, Z. Li and M. Wen, "DCT-GAN: Dilated Convolutional Transformer-based GAN for Time Series Anomaly Detection", in *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[35] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems", in *IEEE Network*, vol.34, no. 3, pp. 134-142, May/June 2020.
[36] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence", in *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, October 2017.
[37] W. Guo, "Explainable Artificial Intelligence (XAI) for 6G: Improving Trust between Human and Machine", in *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39-45, 2020.
[38] L. Fernandez Maimo, A. L. Perales Gomez, F. J. Garcia Clemente, M. Gil Perez and G. Martinez Perez, "A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks", in *IEEE Access*, vol. 6, pp.7700-7712, 2018, DOI: 10.1109/ACCESS.2018.2803446.
[39] M. T. Ribeiro, S. Singh and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier" - *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 2016, DOI: 10.18653/v1/n16-3020.

© 6G-BRICKS 2023

[40] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions", in *Advances in Neural Information Processing Systems*, vol. 30, December 2017.
[41] L. S. Shapley, "A Value for n-Person Games", in *Princeton University Press*, 2016, pp. 307–318, DOI: 10.1515/9781400881970-018.
[42] L. Antwarg, R. M. Miller, B. Shapira and L. Rokach, "Explaining anomalies detected by autoencoders using Shapley Additive Explanations", in *Expert Systems with Applications*, vol. 186, no. 115736, 2021, doi: 10.1016/j.eswa.2021.115736.
[43] M. Carletti, C. Masiero, A. Beghi, and G. A. Susto, "Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis - *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 21–26.
[44] M. Moradi and M. Samwald, "Post-hoc explanation of black-box classifiers using confident itemsets", in *Expert Systems with Applications*, vol. 165, 2021, Art. no. 113941. doi: 10.1016/j.eswa.2020.113941.
[45] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri and F. Turini, "Factual and Counterfactual Explanations for Black Box Decision Making", in *IEEE Intelligent Systems*, vol. 34, no. 6, pp. 14-23, Nov. 2019, pp. DOI: 10.1109/MIS.2019.2957223.
[46] G. Montavon, W. Samek and K.-R. Müller, "Methods for interpreting and understanding deep neural networks" in *Digital Signal Processing*, vol. 73, pp. 1-15, 2018, https://doi.org/10.1016/j.dsp.2017.10.011.
[47] J. Shen, R. Wang and H. W. Shen, "Visual exploration of latent space for traditional Chinese music", in *Visual Informatics*, vol. 4, no. 2, 2020, Art. no. DOI: 10.1016/j.visinf.2020.04.003.
[48] H. D. Trinh, L. Giupponi, and P. Dini, "Urban Anomaly Detection by processing Mobile Traffic Traces with LSTM Neural Networks", in *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, vol. 2019-June, 2019, pp. DOI: 10.1109/SAHCN.2019.8824981.
[49] K. Flanagan, E. Fallon, P. Connolly, and A. Awad, "NetFlow anomaly detection though parallel cluster density analysis in continuous time-series", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10372 LNCS, 2017, pp. DOI: 10.1007/978-3-319-61382-6_18.
[50] S. Al Mamun and M. Beyaz, "LSTM recurrent neural network (RNN) for anomaly detection in cellular mobile networks" - *International Conference on Machine Learning for Networking*, Springer, 2018, pp. 222–237.
[51]"Malware Capture Facility Project, 'The CTU-13 Dataset: A labeled dataset with botnet, normal, and background traffic", https://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html/. Accessed: Feb 21, 2024."
[52] S. Garcia, M.Grill, J. Stiborek and A. Zunino, "An empirical comparison of botnet detection methods", in *Computers & Security*, vol. 45, pp. 100–123, 2014.
[53] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick and E. Fersman, "Explainability Methods for Identifying Root-Cause of SLA Violation Prediction in 5G Network" - *IEEE Global Communications Conference*, 2020.
[54] C.I. Challu, P. Jiang, Y.N. Wu, L. Callot, "Deep Generative model with Hierarchical Latent Factors for Time Series Anomaly Detection" - *International Conference on Artificial Intelligence and Statistics*, pp.1643-1654, 2022.
[55] T. Han, Y. Lu, S.C. Zhu and Y. N. Wu, "Alternating back-propagation for generator network", in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
[56] E. Nijkamp, B. Pang, T. Han, L. Zhou, S.C. Zhu, and Y.N. Wu, "Learning multi-layer latent variable model with short run mcmc inference dynamics" - *European Conference on Computer Vision*, 2021.
[57] 3GPP. Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS). Tech. rep. 23.288. Version 17.8.0. 2023.
[58] S. Malik, M. A. Khan, H. El-Sayed, J. Khan and O. Ullah, "Implanting Intelligence in 5G Mobile Networks—A Practical Approach", in *Electronics*, vol. 11, no. 23, pp. 3933, 2022.

[59] A. Mekrache, K. Boutiba and A. Ksentini, "Combining Network Data Analytics Function and Machine Learning for Abnormal Traffic Detection in Beyond 5G" - *IEEE Global Communications Conference*, pp. 1204-1209, 2023.

[60] M. Said Elsayed, N. A. Le-Khac, S. Dev, and A.D. Jurcut, "Network anomaly detection using LSTM based autoencoder", in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 37-45, Nov. 2020.

[61] A. Chawla, P. Jacob, P. Farrell, E. Aumayr, and S. Fallon, "Towards interpretable anomaly detection: Unsupervised deep neural network approach using feedback loop", in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1-9, Apr. 2022.

[62] Y. Chang et al., "A survey on evaluation of large language models", in *ACM Transactions on Intelligent Systems and Technology*, vol. 15, pp. 1-45, 2024.

[63] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[64] R. Beerten, V. Ranjbar, A. P. Guevara and S. Pollin, "Cell-Free Massive MIMO in the O-RAN Architecture: Cluster and Handover Strategies" - *IEEE Global Communications Conference*, 2023, pp. 5943-5948, doi: 10.1109/GLOBECOM54140.2023.10437450.

[65] R. Beerten, A. Girycki and S. Pollin, "User Centric Cell-Free Massive MIMO in the O-RAN Architecture: Signalling and Algorithm Integration" - *IEEE Conference on Standards for Communications and Networking (CSCN)*, 2022, pp. 181-187, doi: 10.1109/CSCN57023.2022.10050934.

[66] V. Ranjbar, A. Girycki, M. A. Rahman, S. Pollin, M. Moonen and E. Vinogradov, "Cell-Free mMIMO Support in the O-RAN Architecture: A PHY Layer Perspective for 5G and Beyond Networks" in IEEE Communications Standards Magazine, vol. 6, no. 1, pp. 28-34, March 2022, doi: 10.1109/MCOMSTD.0001.2100067.

[67] E. Ong and P. Velickovic, "Learnable Commutative Monoids for Graph Neural Networks" in *Learning on Graphs Conference*. PMLR, 2022.

© 6G-BRICKS 2023