Decentralized AI and Architectures for Massive Wireless Network Slicing Scalability and Sustainability in 6G

RESILIENT

Grant No. TSI-063000-2021-55

E5: Final report on AI for the 6G DAWN AE/MS/DE

Financiado por la Unión Europea
NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

UNICO I+D

GOBIERNO DE ESPAÑA

Centre Tecnològic de Telecomunicacions de Catalunya

Abstract

This report describes the implementation of decentralized 6G DAWN AI (MS-AE-DE) in all RESILIENT PoCs in detail, their KPI measurements according to the defined ones in E3 and their lessons learned during the implementation and experiments.

Document properties

Document number	E5
Document title	Final report on AI for the 6G DAWN AE/MS/DE
Document responsible	Sarang Kahvazadeh (CTTC)
Document editor	Sarang Kahvazadeh (CTTC), Farhad Rezazadeh (CTTC), Selva Via (CTTC)
Authors	Sarang Kahvazadeh (CTTC), Farhad Rezazadeh (CTTC), Engin Zeydan (CTTC), Albert Bel (CTTC), Josep Mangues-Bafalluy (CTTC), Luis Blanco (CTTC), Farhana Javed (CTTC), Fatemehsadat Tabatabaeimehr (CTTC), Jorge Baranda (CTTC), Miquel Payaró Llisterri (CTTC), Oriol Font-Bach (SRS), Ismael Gomez (SRS), Manuel Lorenzo (Ericsson), Saravanan Kalimuthu (Ericsson), JoseLuis Jimenez (Ericsson), Marc Molla (Ericsson), Inmaculada Rafael (Ericsson), Miguel Angel Lopez Serrano (Ericsson), Alvaro Vlad (Ericsson), Carlos Javier Soleto Ramos (Ericsson), Rubén Cerezo (Ericsson), Diego San Cristobal Epalza (Ericsson), Fernando Beltran Gonzalez (Ericsson), Alejandro Ramiro Muñoz (Ericsson), Hristo Koshutanski (ATOS), Ignacio Labrador (ATOS), Manuel Jiménez (ATOS), Sonia Castro (ATOS), Jaime Azcorra (Telcaria), Aitor Zabala (Telcaria)
Target dissemination level	Public
Status of the document	Final
Version	1.0
Delivery date	31 December 2025
Actual delivery date	31 December 2025

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

Disclaimer

This document has been produced in the context of the 6G DAWN Project. The research leading to these results has received funding from the Ministerio de Asuntos Económicos y Transformación Digital (MINECO), under grant TSI-063000-2021-54/-55.

All information in this document is provided “as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the MINECO has no liability in respect of this document, which is merely representing the authors view.

Contents

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

UNICO I+D

GOBIERNO DE ESPAÑA

cttc Centre Tecnològic de Telecomunicacions de Catalunya

List of Figures

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

UNICO I+D

GOBIERNO DE ESPAÑA

ttc Centre Tecnològic de Telecomunicacions de Catalunya

List of Tables

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

List of Acronyms

3GPP – 3rd Generation Partnership Project

5G-ACIA - 5G Alliance for Connected Industries and Automation

ACT- Actuator

AD – Anomaly Detection

AE – Analytics Engine

AF – Application Function

AI – Artificial Intelligence

B5G – Beyond 5G

CaaS – Container as a Service

CQI – Channel Quality Indicator

CNF – Containerized Network Function

COTS – Commercial Off-The-Shelf

CPU - Central Processing Unit

CSP – Communications Service Provider

CSI – Channel State Information

CU- Central Unit

DE – Decision Engine

DL – Downlink

DMO - Domain Manager and Orchestrator

DU- Distributed Unit

E2E – End To End

EC – Energy Consumption

eMBB – Enhanced Mobile Broadband

ETSI – European Telecommunications Standards Institute

gNB - next Generation Node B (5G node B)

GNSS – Global Navigation Satellite system

GPS – Global Positioning System

ICMP – Internet Control Message Protocol

IDMO- Inter-Domain Manager and Orchestrator

ILE- Infrastructure Layer Emulator

IRU – Indoor Radio Unit

ISPM – Infrastructure Status Prediction Module

KPI – Key Performance Indicator

KPM- Key Performance Measurement

MCData – Mission Critical Data

MCPTT – Mission Critical Push To Talk

MCVideo – Mission Critical Video

MCX – Mission Critical services

ML – Machine Learning

MPLS – Multi-Protocol Label Switching

M&O- Management and Orchestration

MS – Monitoring System

NDT – Network Digital Twin

NEF – Network Exposure Function (3gpp)

NETCONF – Network Configuration Protocol

NF – Network Function

NPN- Non-Public Networks

NR – New Radio

NWDAF – Network Data Analytics Function

OAM – Operations, Administration and Maintenance

O-RAN – Open Radio Access Network

PoC- Proof of Concept

OTA – Over-the-Air

OWD – One Way Delay

PCF – Policy Control Function

PN – Public Network (3gpp)

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

PNI-NPN – Public Network Integrated Non-Public Network (3gpp)

PPDR – Public Protection & Disaster Relief

PT-Paquete de Trabajo (Work package)

QoS – Quality of Service

RAN – Radio Access Network

RAN WG3 – RAN (Radio Access Network) Workgroup 3

R- RESILIENT

RB – Resource Block

RC – Radio Controller

RDI – Radio Dot Interface

RDS – Radio Dot System

RIC - RAN Intelligent Controller

RF – Radio Frequency

RLC – Radio Link Control

RT- Real Time

RTT - Round-Trip Time

SA1 – System Aspects Workgroup 1

SDN – Software Defined Networking

SDR-Software Define Radio

TDD – Time Division Duplex

TETRA – Terrestrial Trunked Radio

TS – Technical Specification

TR – Technical Report

UC- Use Case

UE – User Equipment

UL – Uplink

UPF- User Plane Function

URLLC – Ultra-reliable Low-Latency Communication

USA – United States of America

Financiado por la Unión Europea NextGenerationEU · Plan de Recuperación, Transformación y Resiliencia · UNICO I+D · GOBIERNO DE ESPAÑA · cttc Centre Tecnològic de Telecomunicacions de Catalunya

USRP - Universal Software Radio Peripheral

VPN – Virtual Private Network

WG- Working Group

xApp-Cross Application

YANG – Yet Another Next Generation

ZDM – Zero-Defect Manufacturing

ZSM – Zero-touch Service Management

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

Executive Summary

This document presents the implemented 6GDAWN AI closed-loop (MS-AE-DE) in all defined RESILIENT PoCs and their requirements according to Deliverable E3. This deliverable illustrates the AI closed-loop (MS-AE-DE) implementation, KPI measurement methods, and lessons learned across RESILIENT PoCs. This project decentralized security —intrusion detection and mitigation— and analyzes the trade-offs between network resiliency, QoS, and energy efficiency, demonstrating the architecture applicability to critical operational challenges. We conclude the deliverable with successful implementation of decentralized AI (MS-AE-DE) in all RESILIENT PoCs by measuring the associated KPIs as defined in E3.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

1 Introduction

This deliverable reports the final results of 6G DAWN's decentralized AI closed loop—Monitoring System, Analytics Engine, and Decision Engine (MS–AE–DE)—validated through Proofs of Concept (PoCs) targeting resilience in next-generation wireless networks. The work is grounded in cross-domain, real-world deployments with measurable KPIs and repeatable control loops that manage energy, performance, and security under operational constraints. The RESILIENT project integrates decentralized security and trust, so threats are detected and mitigated without degrading service integrity, including improving network reliability by identifying and isolating users with disproportionate energy consumption.

Concretely, resilience is demonstrated through (i) decentralized intrusion detection and mitigation in Kubernetes with end-to-end 5G connectivity, (ii) strengthening the trustworthiness of federated learning via blockchain, and (iii) enhancing NPN resilience by detecting and isolating heavy energy-consuming users.

The mentioned PoCs operationalize the decentralized AI (MS–AE–DE) loop over the defined interfaces, turning telemetry into timely, auditable action.

In the following sections, we describe all RESLIENT PoCs implementation details, KPIs measurements and lessons learned according to E3. In Table 1, a summary of Use case and PoC mapping with decentralized AI (MS-AE-DE) is illustrated.

TABLE 1. ARCHITECTURAL OVERVIEW

PoC	MS	AE	DE	ACT
R-UC1-PoC1	LADS-Sensor (Flow Telemetry)	LADS-Brain	Mitigator	Mitigator
R-UC1-PoC2	SMO-Client FL Training Host, Chainlink Adapter / CLSP-BC-CAI, performanceSubmission.sol (submitNMSE)	Reputation evaluation logic (off-chain reputation script, reputationCalculation.sol / updateScores)	ReputationCalculation.sol / selectTopPerformers, SMO-Aggregator (client selection policy)	SMO-Aggregator FL Aggregator (global model update), SMO-Client FL Inference Host
R-UC2-PoC1	NDT platform/MS, AF	NDT platform/AE, AF	AF	NDT platform/ACT

Financiado por la Unión Europea NextGenerationEU · Plan de Recuperación, Transformación y Resiliencia · UNICO I+D · Gobierno de España · cttc Centre Tecnològic de Telecomunicacions de Catalunya

2 RESILIENT Use Cases

Two different use cases, which comprise a total of three PoCs, are defined under 6G DAWN RESILIENT project:

- Use case R1 Decentralized 6G Security & Trust
  - R UC1 PoC1 -Decentralized intrusion detection and mitigation
  - R UC1 PoC2 – Trustworthy Federated Learning enabled through smart contracts and Blockchain
- Use case R2 Network Resiliency vs QoS and Energy Efficiency
  - R UC2 PoC1 - Detecting and Isolating Energy Consumption-Heavy Users

The following table presents the mapping of Key concepts and PoCs that apply for the RESILIENT project:

TABLE 2. MAPPING OF KEY CONCEPTS AND POCS

Key Concepts	**R UC1 PoC1**	**R UC1 PoC2**	**R UC2 PoC1**
NPN Digital Twin System			X
Extreme Edge	X		
AI/ML agent for control loops	X		X
xApps in O-RAN	X		
Relation of vertical KPIs with the network configuration	X		X
Inter(a)-slice reconfiguration and massive slicing			X
NEF instance for KPI data and configuration capabilities exposures of NPNs			X
AI/ML methods for reducing energy consumption			X
Develop standard compliant network interfaces to support AI driven network fault management systems at NPN			X
Decentralized Intrusion Detection to integrate novel trust-based evaluation mechanisms	X		
Blockchain in 5G		X	
Identification and handling of abnormally heavy energy components			X

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

2.1 Use Case RESILIENT Decentralized 6G Security & Trust

2.1.1 RESILIENT UC1 PoC1 -Decentralized intrusion detection and mitigation

This PoC will leverage an existing proprietary solution named LADS. LADS is a soft-real time anomaly-based network intrusion detection system (of type A-NIDS). It is based on an in–house engine for feature extraction from network traffic, and two deep learning algorithms – one for anomaly detection and another one for attack classification. The PoC of LADS in the 6G DAWN project is formed by the feature extraction engine and the unsupervised deep learning algorithm for anomaly detection. An important functionality of LADS is the capacity to explain why and what network behavioral aspects cause an anomaly. This feature facilitates decision making on what mitigations to be applied on the affected assets in the monitored environment.

2.1.1.1 PoC Implementation details

PoC Testbed Architecture

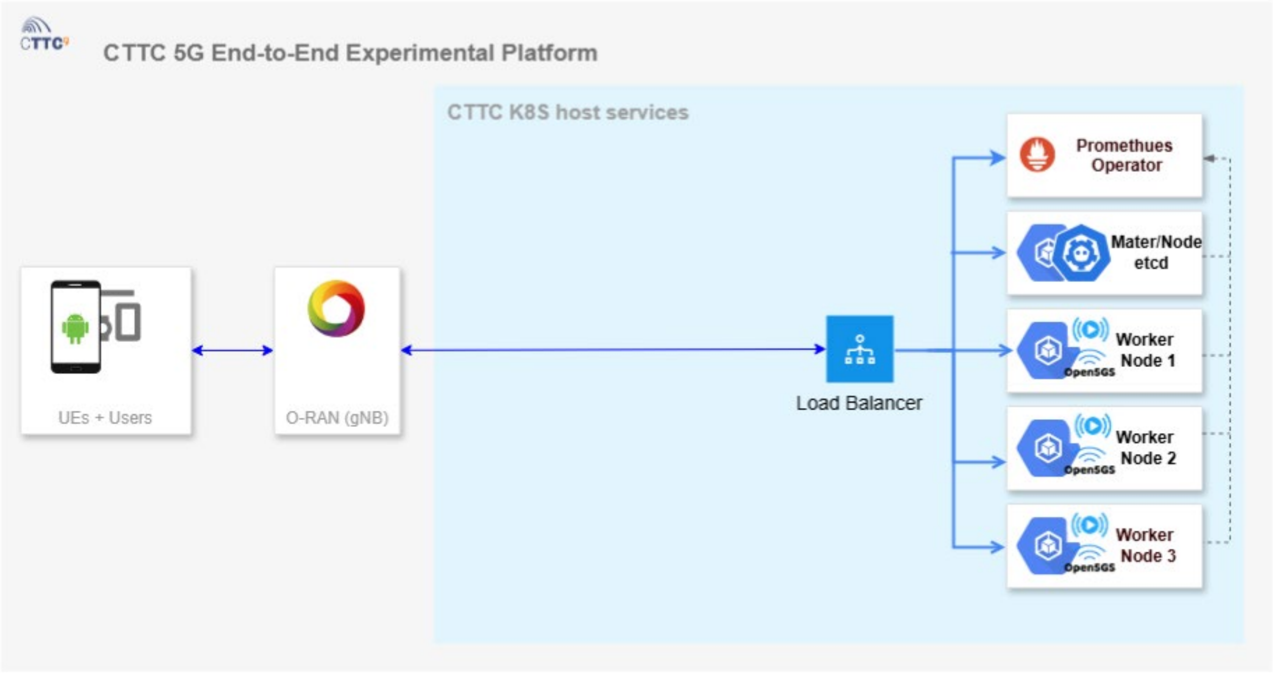


FIGURE 1. RESILIENT UC1 POC1 5G TESTBED ARCHITECTURE

As illustrated in Figure 1, This PoC is implemented and tested in CTTC premises. CTTC deployed a Kubernetes cluster with 1 master and 3 workers nodes. In the Cluster all toolkits such as MetallB loadbalancer, Prometheus and Grafana for monitoring, and persistent volume are deployed. In this k8s, open5gs core without UPF in one namespace, UPF in another name space and video on-demand streaming are implemented and deployed. The 5G core in k8s is routed and connected to the O-RAN for bringing E2E 5G connectivity and CTTC also uses different kinds of smart phone for testing purposes. This k8s cluster hosts all the decentralized intrusion detection and mitigation components.

At the RAN side, R UC1 PoC1 utilizes srsRAN Project to implement the O-RAN functions of the CU/DU, as introduced above in 2.1.1.1. For this PoC, and in collaboration with the 6GBLUR-Joint UNICO I+D 5G project (TSI-063000-2021-57), the CU/DU features have been extended to support slice-aware scheduling. Furthermore, the gNB also provides extended E2SM-KPM and E2SM-RC features, as detailed in Table 3 and Table 4.

TABLE 3. E2 TRAFFIC-LOAD METRICS PROVIDED BY SRSRAN PROJECT IN R UC1 POC1

Category	Name	Description
Data Radio Bearer	DRB.RlcSduTransmittedVolumeUL	Transmitted UL data volume
	DRB. RlcSduTransmittedVolumeDL	Transmitted DL data volume
	DRB.PacketSuccessRateUlgNBUu	UL PDCP SDU success rate
	DRB.PerDataVolumeDLDist.Bin	Incoming DL data success rate per UE
	DRB.PerDataVolumeULDist.Bin	Incoming UL data success rate per UE
Radio Resource Utilization	RRU.PrbDl	DL PRB usage for user-plane traffic
	RRU.PrbUl	UL PRB usage for user-plane traffic
	RRU.PrbTotDl	DL PRB usage for all traffic
	RRU.PrbTotUl	UL PRB sage for all traffic

TABLE 4. E2 RC ACTIONS PROVIDED BY SRSRAN PROJECT IN R UC1 POC1

Category	Name	Description
Radio Resource Allocation Control	Slice-level PRB quota	Enables modifying the resource usage quota for the different RAN users

In the context of the PoC’s final implementation, we have defined the following components and mapping:

- *LADS-Sensor* (flow telemetry) assumes the role of a Monitoring System (MS).
- *LADS-Brain* assumes the role of an Analytics Engine (AE),
- *Mitigator-D* assumes the role of a Decision Engine (DE),
- *Mitigator-A* assumes the role of an actuator (ACT).

LADS-Sensor

The LADS-Sensor plays an essential role in monitoring the network activities in Kubernetes clusters. Its primary purpose is to achieve full visibility of network traffic between worker nodes in a cluster and within pods in each worker node. Additionally, all external traffic - outgoing and incoming traffic

Financiado por la Unión Europea NextGenerationEU · Plan de Recuperación, Transformación y Resiliencia · UNICO I+D · Gobierno de España · cttc Centre Tecnològic de Telecomunicacions de Catalunya

- to cluster's services. The LADS-Sensor generates flow telemetry data along a set of features that are used by the LADS-Brain.

A LADS-Sensor is deployed on each worker node as a DaemonSet[1]. This is a standard practice of Kubernetes to local-node monitoring. Such setting allows the LADS-Sensor to access all virtual network interfaces created on each node, and sniff traffic on those. In addition, the sensor periodically goes through all virtual network interfaces and updates those newly created and discards sniffing from non-existing (outdated) ones. This is an important functionality towards full traffic visibility and automated adaptation to dynamic changes over time on worker nodes.

The flow telemetry of the LADS-Sensor has been specifically extended to represent flows suitable for a Kubernetes computing cluster. Additionally, the LADS-Sensor has been extended to cover flow telemetry for the SCTP and HTTP REST communications, as well as specific to ARP protocol to determine suspicious or conflicting MAC-IP announcements.

LADS-Brain

The LADS-Brain is the cornerstone of the LADS workflow, where a high volume and dimensionality of data from the LADS-Sensors are processed for detection of anomalies. The LADS-Brain uses unsupervised deep learning algorithm for training and prediction modalities.

Training modality, the LADS-Brain uses the flow telemetry from all LADS-Sensors to train a model for the given observed network behavior of the cluster. The threshold is set up during the training phase according to how well the deep learning model has learned the normal traffic data and is used in the prediction modality. We note that the training is always performed on the telemetry data from all nodes, while the monitoring-prediction modality can be performed either in a fully decentralized deployment or in a cluster-centric one, as explained below.

Monitoring modality, also referred to as a prediction modality, the LADS-Brain loads the model from the training phase to observe if there is any deviation in the traffic. These deviations are defined as values of a reconstruction error, i.e., how much a traffic flow resembles (can be reconstructed) according to the trained model. A reconstruction error below a given threshold is considered normal traffic, and above a threshold as anomalous traffic. Additionally, the Brain determines what critical features from the flow telemetry cause the detected anomaly and provides a short report of unusual values that deviate from the observed ones during the training.

Such explainability is very useful for determining types of attacks or intrusions, and potential mitigation actions.

[1] https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

Financiado por la Unión Europea NextGenerationEU Plan de Recuperación, Transformación y Resiliencia UNICO I+D GOBIERNO DE ESPAÑA cttc Centre Tecnològic de Telecomunicacions de Catalunya

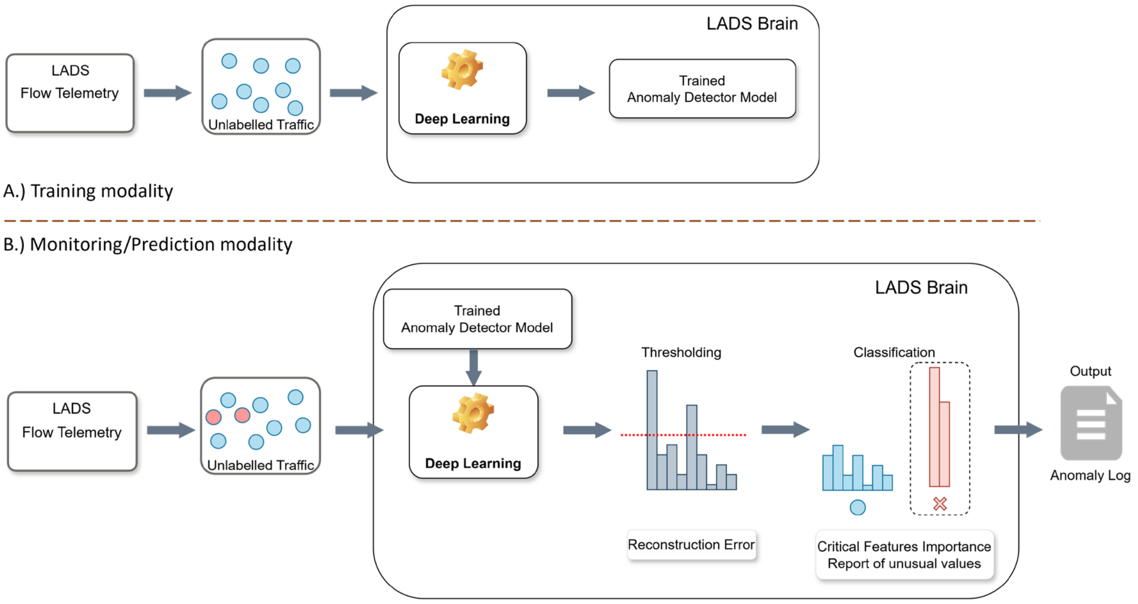


FIGURE 2. LADS WORKFLOW TRAINING AND MONITORING MODALITIES

Figure 2 illustrates the LADS workflow discussed above, highlighting the detection of anomalies (anomalous flows) based on the trained model and threshold value. The output of the LADS-Brain is a log file containing all events of anomaly detection.

Mitigator

The Mitigator component is specifically defined and implemented for the final PoC implementation and demonstration. It has two main roles:

- *Mitigator-D*: Its main role is to *monitor* the event log of the LADS-Brain and *match* a set of rules for each new event added by the Brain. Each set of rules corresponds to a specific asset(s) and type of anomaly that needs mitigation. For instance, if a UPF pod or service receives a high-volume incoming traffic, say more than 500% of packet/s (pps) or bytes/s (bps) than those in training, it is considered as a potential denial of service against the UPF pod/service. In this case, the Mitigator-D triggers the assigned to this set of rules (type of anomaly) a mitigation action.
- *Mitigator-A*: Its main role is to offer a set of named mitigation action and their underlying execution means. In the case of the PoC, the execution means are scripts that interact with the Kubernetes control plane to stop, move, restart pods. For instance, in the case of a potential DoS against the UPF service, Mitigator-A executes the move action of the UPF pod implementing the UPF service to another worker node to ensure continuity of the service.
  We note that the LADS-Brain offers mapping of IP addresses found in anomalies to the corresponding Kubernetes pods or endpoints. It facilitates the application of mitigation.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

Architecture

The architecture of the decentralized intrusion detection and mitigation solution of the 6GDAWN project is shown in Figure 3 and Figure 4.

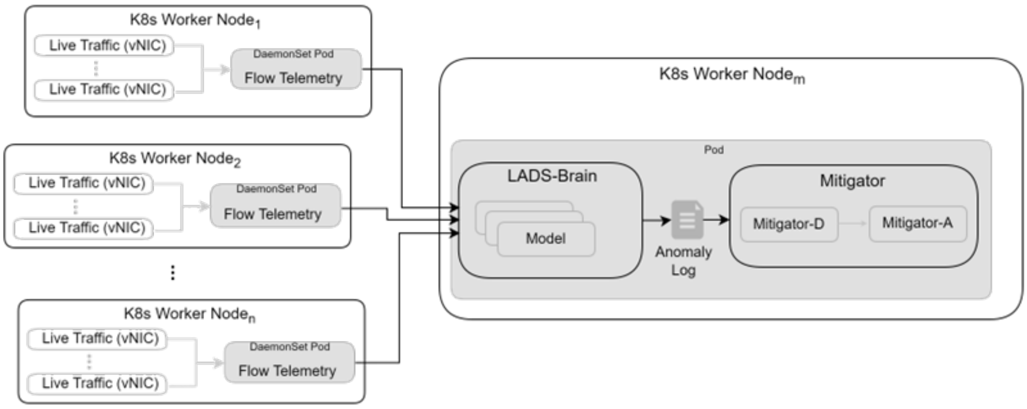


FIGURE 3. DISTRIBUTED MONITORING CLUSTER-CENTRIC DETECTION AND MITIGATION WORKFLOW

The notion of decentralized intrusion detection is realized by (i) decentralization of the LADS-Sensor on each and every node in a Kubernetes cluster including the master node, and (ii) decentralization of the LADS-Brain on one or more nodes in a Kubernetes cluster, or on each node of the cluster.

Figure 3 shows a cluster-centric deployment where a Kubernetes cluster (composed of a set of nodes) is associated with one LADS-Brain instance deployed on one of the nodes of the given cluster. This relation is not a strict one but recommended and preferred in terms of resource optimization for decision-making. In some settings, one may associate multiple Kubernetes clusters to one LADS-Brain, but in such cases, it is needed to dedicate higher compute and memory resources for the node that will host the LADS-Brain. It is recommended to dedicate one or more deep leaning models per cluster.

Furthermore, Figure 3 can be applied by the decentralization of the LADS-Sensor on each and every node in a slice, and the decentralization of the LADS-Brain on one or more slices for a cluster of slices. The LADS Sensor offers visibility of network telemetry per node or segment of a slice, while the LADS Brain manages and assigns deep learning models corresponding to a slice or to a family of protocols in a slice. This allows flexible per customer needs, assignment and processing of network telemetry and machine learning models for decentralized intrusion detection.

Financiado por la Unión Europea NextGenerationEU Plan de Recuperación, Transformación y Resiliencia UNICO I+D GOBIERNO DE ESPAÑA cttc Centre Tecnològic de Telecomunicacions de Catalunya

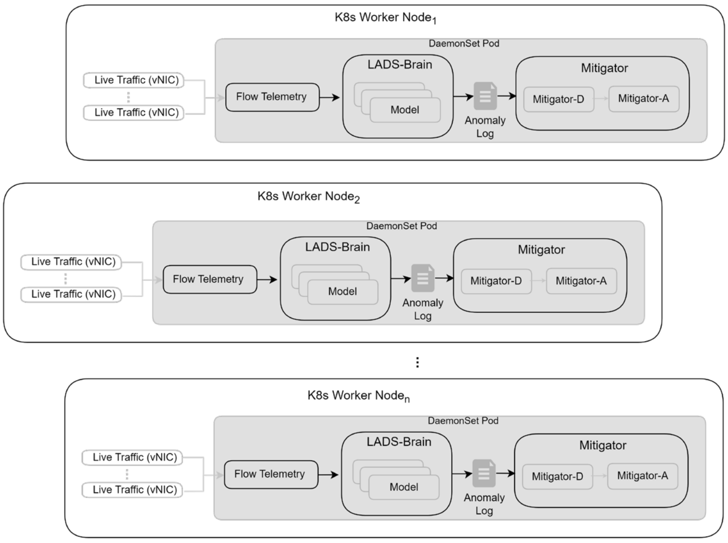


FIGURE 4. DECENTRALISED MONITORING, DETECTION AND MITIGATION WORKFLOW

Given end-user needs, a fully decentralized deployment of the LADS-Brain and Mitigator is supported by the current implementation, where each node in a cluster hosts the Sensor, the LADS-Brain with the trained models, and the Mitigator in a DaemonSet pod. This modality is illustrated in Figure 4. This modality has the benefit of being independent from other nodes local monitoring and detection but requires each node of the cluster to offer the necessary hardware resources CPU and RAM.

In the overall architecture, the LADS-Brain contains one or more trained anomaly detector models (deep learning models), and each model is associated either to a set of LADS-Sensors, or to a family of protocols across all LADS-Sensors, or both to a set of LADS-Sensors and a family of protocols. It is relatively easy to perform such association of Sensors' data to models on the level of the Brain by means of *pre-processing* all flow telemetry arriving at the LADS-Brain into *subsets* of flows, each one corresponding to a set of sensors and/or protocol families.

Thus, each subset of flow telemetry data is used for training, and later the same process for prediction modality. We note that each flow telemetry contains the necessary data to identify which sensor produced the flow telemetry, and what protocol, ports, etc. compose a flow.

We also note that the location of the LADS-Brain is not relevant to which worker node it is deployed as long as the node offers the minimum resources[2] needed for the Brain. There is a requirement in the current implementation on the Mitigator's location – it shall be deployed on the same node where the Brain is deployed. We recall that the Mitigator enforces mitigations against the Kubernetes' control plane, and as such mitigations apply (have a scope) on each node needed.

[2] For the sake of reference, the very minimum resources for the LADS-Brain are 2 CPUs and 2 GB RAM.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

The Mitigator does not need to be on each and every node, but on the node of the Brain. Such a restriction can be relaxed offering the Mitigator on a different node from the Brain.

2.1.1.2 R UC1 PoC1 KPIs Evaluation and Results

We first recall the KPIs defined for Resilient UC1 PoC1. Table 5 shows the KPIs as defined in E3 for the R UC1 PoC1, and their refence points of evaluation defined in the final pilot implementation.

TABLE 5. RESILIENT UC1 POC1 KPIS AND EVALUATION POINTS IN FINAL POC IMPLEMENTATION

KPI	Unit	Type	Definition	Evaluation Points
Attacks from OWASP	Categorical	Attack	At least two types of attacks stemming from OWASP Kubernetes Top Ten.	Selected two types of attacks stemming from Insecure Workload Configurations[3]: - Unauthorized access to host environment - Privilege escalation to inject false packets in host network
Attacks of high impact	Categorical	Attack	At least two types of attacks of high impact such as FDI, MitM, Unauthorized access, DDoS.	- Unauthorized access - DoS
Attacks relevant to the container matrix	Categorical	Attack	At least two types of attacks relevant to the "Containers Matrix" of MITRE ATT&CK framework.	- End-point denial of service[4] - Network denial of service[5]
F1-Score	Numerical	Anomaly Detector	Combines precision and recall when evaluating a classification model. It summarizes the model's ability to be both accurate and comprehensive. A higher F1-Score indicates better model performance.	F1-Score evaluation results for each of the two types of attacks – end-point DoS and network DoS.
Confusion matrix	Numerical	Anomaly Detector	Comparison of the original values against the predicted values. In this case, it compares normal and anomalous flows.	Confusion matrix for each of the two types of attacks – end-point DoS and network DoS.
Time to Detect Threats	Numerical	Anomaly Detector	The time taken to detect an intrusion after it has occurred	Given the network focus of the PoC and the adopted LADS system, this KPI will be evaluated as part of the next one on the Network Threat Detection time. In fact, we will refer to the next KPI as the reference KPI from the two.

[3] https://owasp.org/www-project-kubernetes-top-ten/2022/en/src/K01-insecure-workload-configurations

[4] https://attack.mitre.org/techniques/T1499

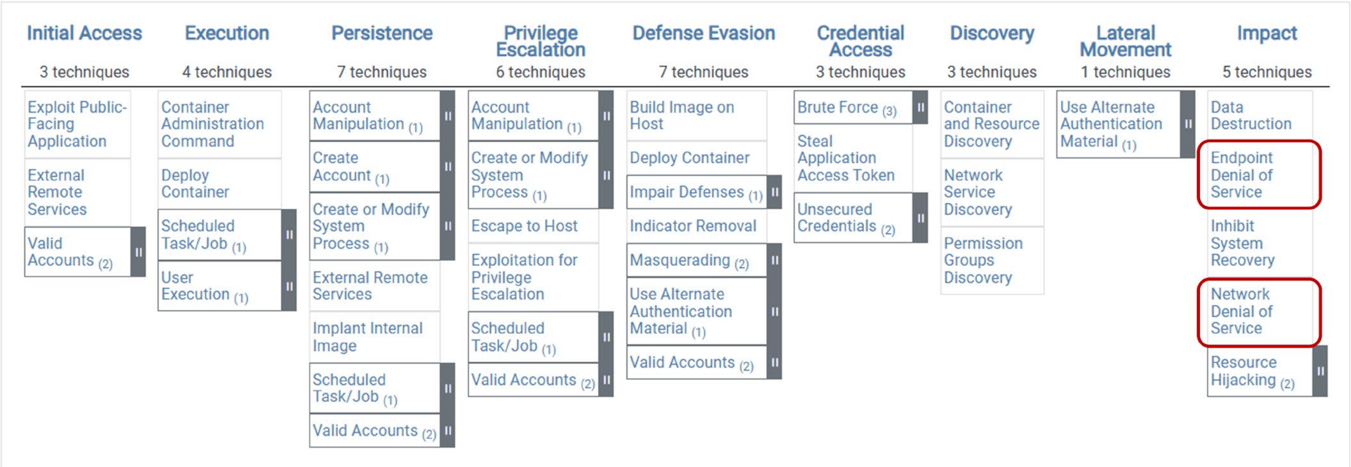
[5] https://attack.mitre.org/techniques/T1498

Network Threat Detection time	Numerical	Anomaly Detector	The time taken to detect an intrusion after it has occurred in network	This KPI is measured from the time the attack (first packet) launched to the time LADS stores in its event log the anomaly.
Increase in detection accuracy	%	Anomaly Detector	Increasing the accuracy in intrusion detection beyond the SOTA	Reference baseline is ⩾**96%** for DoS attacks detection accuracy on network level according to the state-of-the-art[6]. Note, we selected the SOTA based on closest to LADS machine learning approach, Kubernetes environment, and DoS attack type.
Time to Respond to Threats	Numerical	Anomaly Detector	Decreasing the time to response to attack after detection	Measure the time from when LADS outputs an event of detected anomaly (attack) to the time the mitigation is triggered and completed. We note that time of mitigation execution may vary from one execution env to another one, but the KPI is still a very useful reference.

Attack scenarios.

We selected two attacks that lead to high impact on containers and container orchestration systems such as Kubernetes according to the TTPs of the MITRE ATT&CK Containers Matrix[7]:

- *Endpoint Denial of Service* – the VStream endpoint of the testbed service provisioning.
- *Network Denial of Service* –the UPF service of the 5G connectivity testbed.



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
3 techniques	4 techniques	7 techniques	6 techniques	7 techniques	3 techniques	3 techniques	1 techniques	5 techniques
Exploit Public-Facing Application	Container Administration Command	Account Manipulation (1)	Account Manipulation (1)	Build Image on Host	Brute Force (3)	Container and Resource Discovery	Use Alternate Authentication Material (1)	Data Destruction
External Remote Services	Deploy Container	Create Account (1)	Create or Modify System Process (1)	Deploy Container	Steal Application Access Token	Network Service Discovery		Endpoint Denial of Service
Valid Accounts (2)	Scheduled Task/Job (1)	Create or Modify System Process (1)	Escape to Host	Impair Defenses (1)	Unsecured Credentials (2)	Permission Groups Discovery		Inhibit System Recovery
	User Execution (1)	External Remote Services	Exploitation for Privilege Escalation	Indicator Removal				Network Denial of Service
		Implant Internal Image	Scheduled Task/Job (1)	Masquerading (2)				Resource Hijacking (2)
		Scheduled Task/Job (1)	Valid Accounts (2)	Use Alternate Authentication Material (1)				
		Valid Accounts (2)		Valid Accounts (2)				

FIGURE 5: MITRE ATTACK CONTAINERS MATRIX (SOURCE[7]) SCOPE FOR R UC1 POC1

Figure 5 shows MITRE ATT&CK Containers Matrix, and the scope of the attacks' impact.

[6] Chin-Wei Tien, Tse-Yung Huang, Chia-Wei Tien, Ting-Chun Huang, Sy-Yen Kuo, "KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches", *Engineering Reports*, Dec 2019, Journal article, DOI: 10.1002/eng2.12080

[7] https://attack.mitre.org/matrices/enterprise/containers/

We performed the following network attacks:

- *ARP poisoning (blackholing) on the UPF*. Particularly, spoofing the MAC address of the local node gateway on the UPF pod's ARP table. The aim is to DoS (blackhole) the UPF pod's outgoing communications, and consequently the whole UPF service. As a result, the UPF pod directs all outgoing communications to the spoofed MAC address of the gateway IP, which drops all of them. This is an OSI layer 2 cyber-attack.
- *ARP poisoning (blackholing) on the VStream*. Particularly, spoofing the MAC address of the local node gateway on the VStream pod's ARP table. The aim is to DoS (blackhole) the VStream pod's outgoing communications, and consequently the VStream service. As a result, the VStream pod directs all outgoing communications to the spoofed MAC address of the gateway IP, which drops all of them. This is an OSI layer 2 cyber-attack.
- *ICMP ping flood on the UPF and VStream* to show another attack with a similar DoS impact but of a high volumetric nature on the OSI layer 3.

Both types of attacks – ARP poisoning and ICMP ping flood, against the UPF and VStream service are based on *Insecure Workload Configurations* allowing unauthorized access to the host of a node, and *privilege escalation* to execute false packets against the victims' pods on the node.

We demonstrate that both attacks are successfully detected on the network layer and mitigations timely applied to ensure continuity of both services – the UPF and VStream ones.

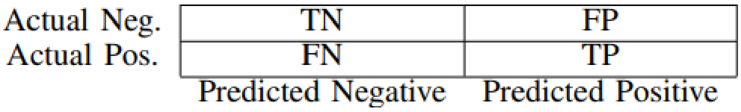
KPI evaluation results.

A default threshold 10 has been placed for all KPIs reported below. For ARP-based DoS (blackholing) attack and ICMP Ping flood attack, we will report the relevant KPIs, such as *F1-Score*, *Confusion matrix = < FP, TP, FN, TN >*, and *Accuracy (⩾96%)*. The Time to Respond to Threats has been measured to **719 ms** (average).

Two types of deep learning models have been used in the experiments – *default* and *host*. The default model englobes the flow telemetry of all protocols in a given environment, while the host one englobes the telemetry of all flows grouped per host/entity/IP in a given environment. They are complementary in scope and visibility.

All attack trials have been tailored against worker2 and worker3 only. Worker1 was used for normal traffic only. We did not use the master node for any worker activity in our experiments (this does not change the applicability or scope of the trials).

All numbers in the column 'support' and in the figures of confusion matrix below show the number of flows. The confusion matrix is characterized by comparing actual values with predicted values. In a binary classification problem, the matrix is described as shown in Figure 6.

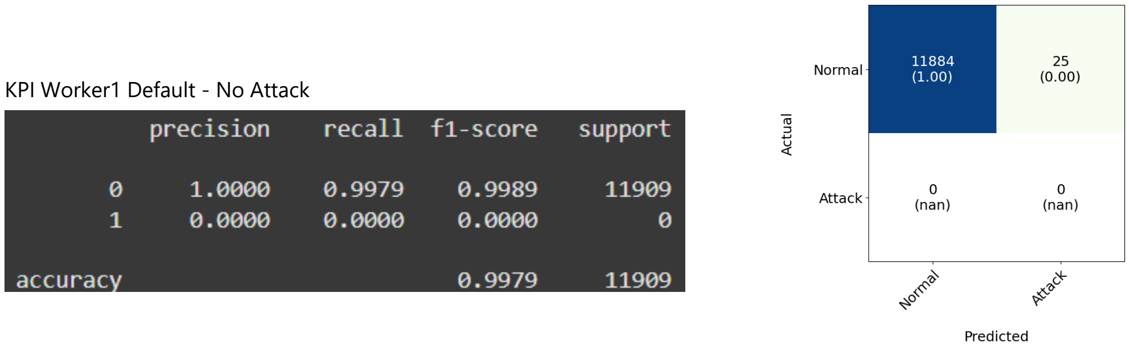


Actual Neg.	TN	FP
Actual Pos.	FN	TP
	Predicted Negative	Predicted Positive

FIGURE 6. CONFUSION MATRIX BINARY CLASSIFICATION

where, TN (True Negatives) are the number of legit flows that have correctly been predicted as legit traffic. FP (False Positives) are the number of legit flows that have wrongly been predicted as attack traffic. FN (False Negatives) are the number of attack flows that have wrongly been predicted as legit traffic. TP (True Positives) are the number of attack flows correctly predicted as attack traffic.

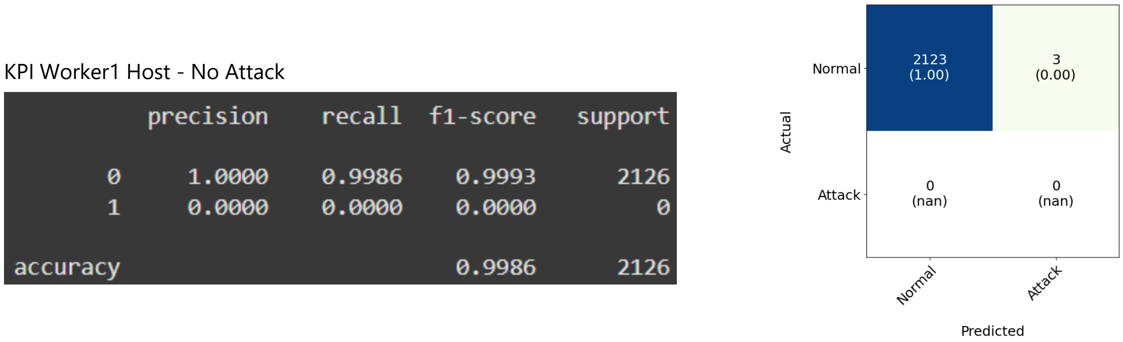
In the following we show the metrics calculated based on the performed attack trials in the testbed and for each of the deep learning models default and host. For selected attacks instances, we show the packets/s distribution over the trial period to better illustrate the attack nature.



KPI Worker1 Default - No Attack

```
              precision    recall  f1-score   support

           0     1.0000    0.9979    0.9989     11909
           1     0.0000    0.0000    0.0000         0

    accuracy                         0.9979     11909
```

FIGURE 7. RESILIENT UC1 POC1 KPI – WORKER1 DEFAULT - NO ATTACK

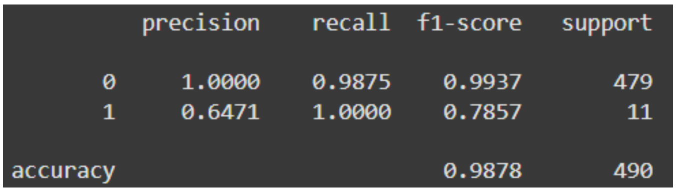


KPI Worker1 Host - No Attack

```
              precision    recall  f1-score   support

           0     1.0000    0.9986    0.9993      2126
           1     0.0000    0.0000    0.0000         0

    accuracy                         0.9986      2126
```

FIGURE 8. RESILIENT UC1 POC1 KPI – WORKER1 HOST - NO ATTACK

KPI Worker3 Default - ARP DoS on UPF (1st trial)



```
              precision    recall  f1-score   support

           0     1.0000    0.9875    0.9937       479
           1     0.6471    1.0000    0.7857        11

    accuracy                         0.9878       490
```

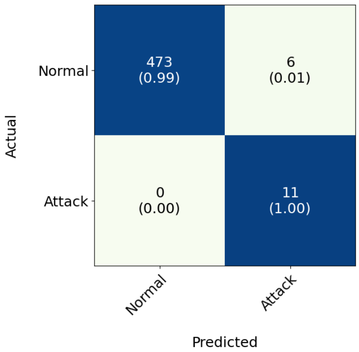
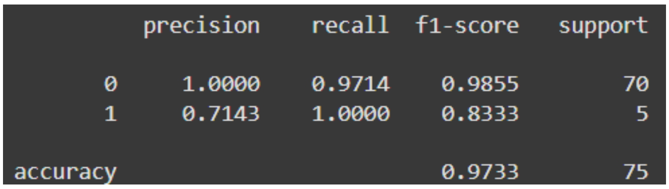


FIGURE 9. RESILIENT UC1 POC1 KPI – WORKER3 DEFAULT - ARP DOS ON UPF (1ST TRIAL)

KPI Worker3 Host - ARP DoS on UPF (1st trial)



```
              precision    recall  f1-score   support

           0     1.0000    0.9714    0.9855        70
           1     0.7143    1.0000    0.8333         5

    accuracy                         0.9733        75
```

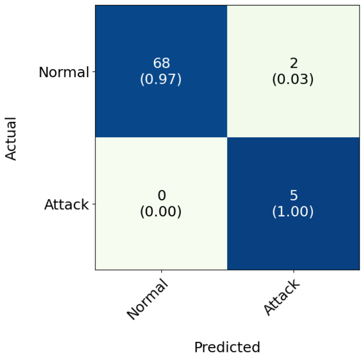
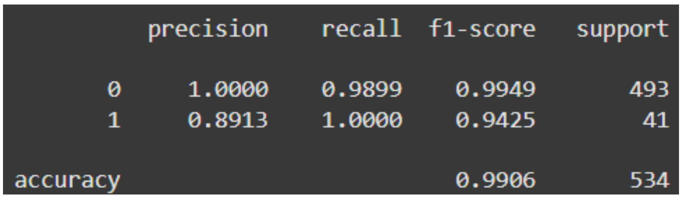


FIGURE 10. RESILIENT UC1 POC1 KPI – WORKER3 HOST - ARP DOS ON UPF (1ST TRIAL)

KPI Worker3 Default - ARP DoS on UPF (2nd trial)



```
              precision    recall  f1-score   support

           0     1.0000    0.9899    0.9949       493
           1     0.8913    1.0000    0.9425        41

    accuracy                         0.9906       534
```

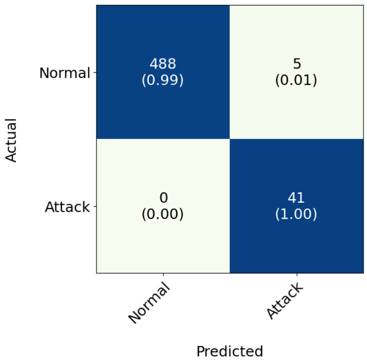
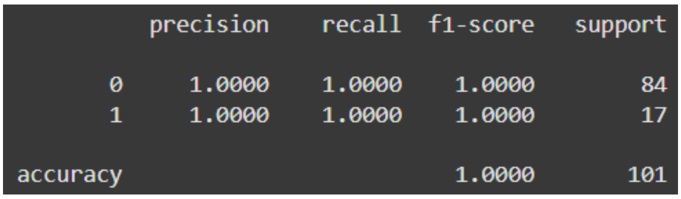


FIGURE 11. RESILIENT UC1 POC1 KPI – WORKER3 DEFAULT - ARP DOS ON UPF (2ND TRIAL)

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

KPI Worker3 Host - ARP DoS on UPF (2nd trial)



```
              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000        84
           1     1.0000    1.0000    1.0000        17

    accuracy                         1.0000       101
```

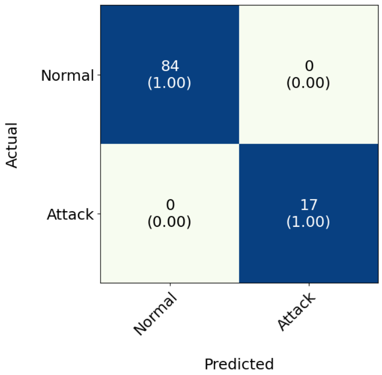
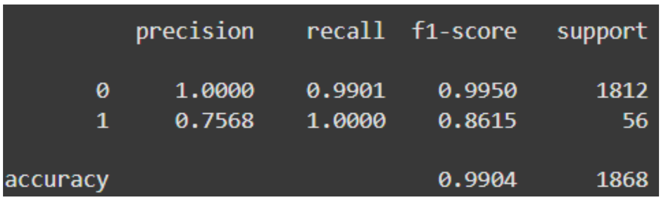
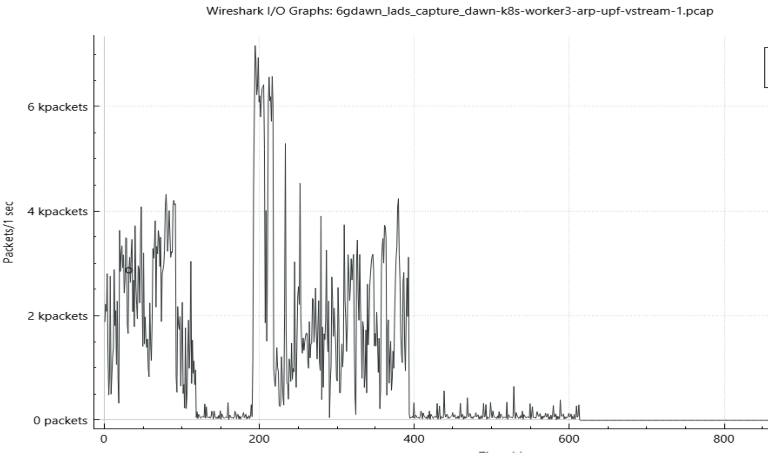


FIGURE 12. RESILIENT UC1 POC1 KPI – WORKER3 HOST - ARP DOS ON UPF (2ND TRIAL)

KPI Worker3 Default - ARP DoS on UPF & VStream



```
              precision    recall  f1-score   support

           0     1.0000    0.9901    0.9950      1812
           1     0.7568    1.0000    0.8615        56

accuracy                             0.9904      1868
```



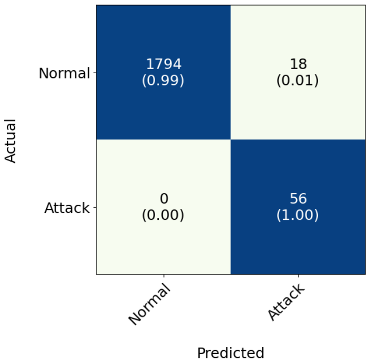
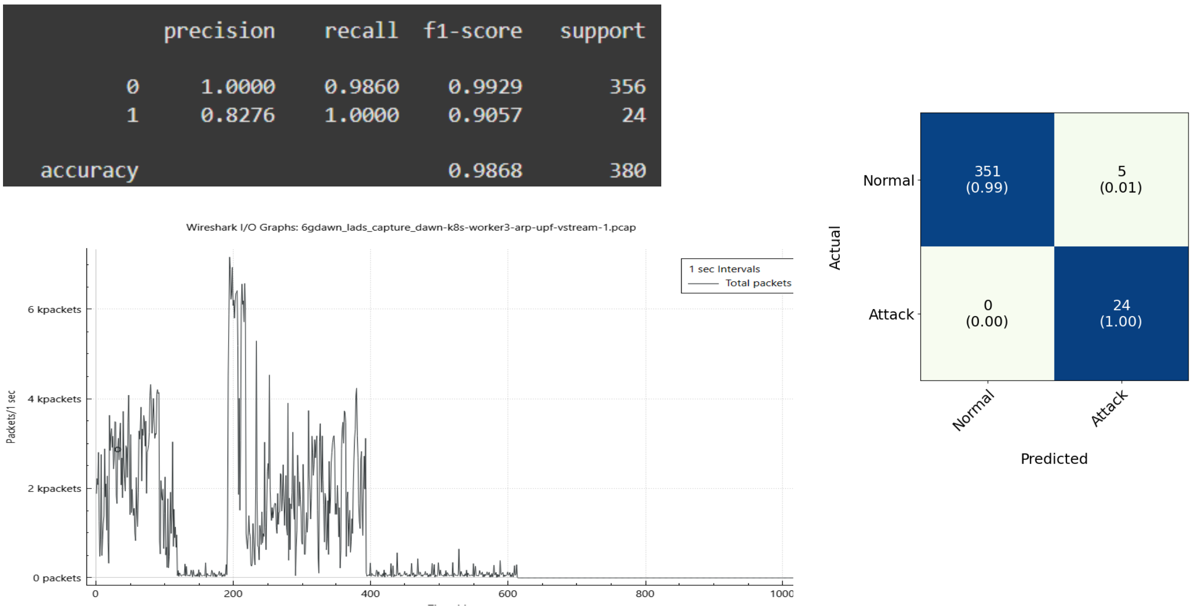


FIGURE 13. RESILIENT UC1 POC1 KPI – WORKER3 DEFAULT - ARP DOS ON UPF & VSTREAM

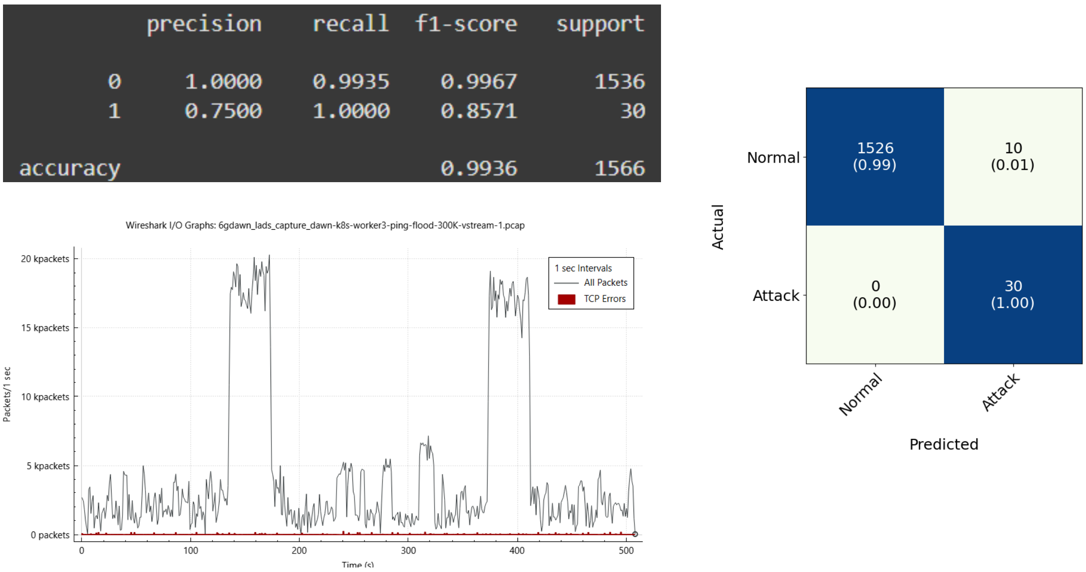
KPI Worker3 Host - ARP DoS on UPF & VStream



```
              precision    recall  f1-score   support

           0     1.0000    0.9860    0.9929       356
           1     0.8276    1.0000    0.9057        24

    accuracy                         0.9868       380
```

FIGURE 14. RESILIENT UC1 POC1 KPI – WORKER3 HOST - ARP DOS ON UPF & VSTREAM

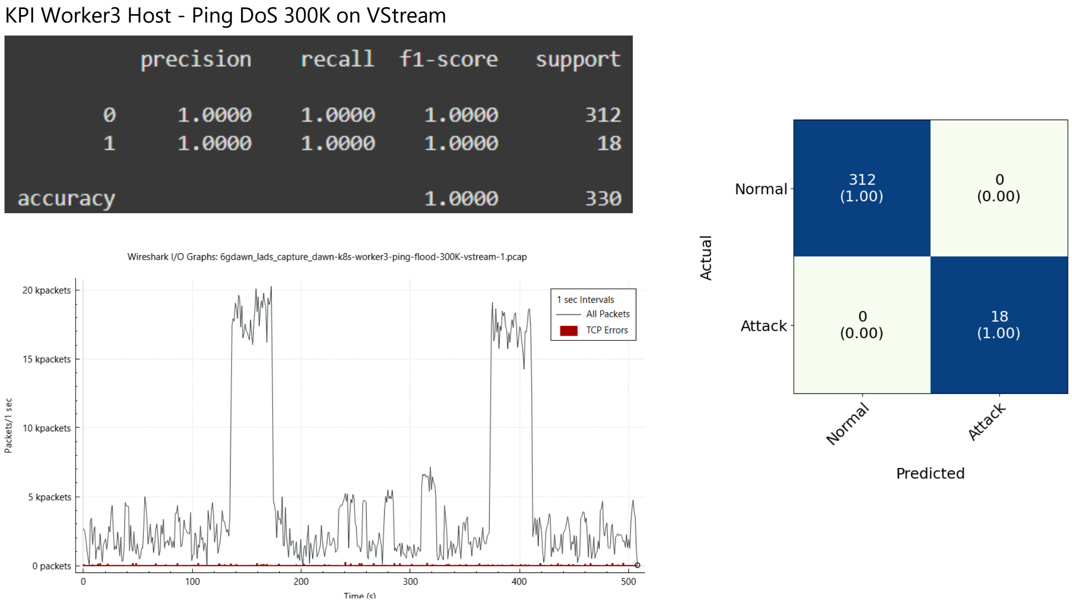
KPI Worker3 Default - Ping DoS 300K on VStream



```
              precision    recall  f1-score   support

           0     1.0000    0.9935    0.9967      1536
           1     0.7500    1.0000    0.8571        30

    accuracy                         0.9936      1566
```

FIGURE 15. RESILIENT UC1 POC1 KPI – WORKER3 DEFAULT - PING DOS 300K ON VSTREAM

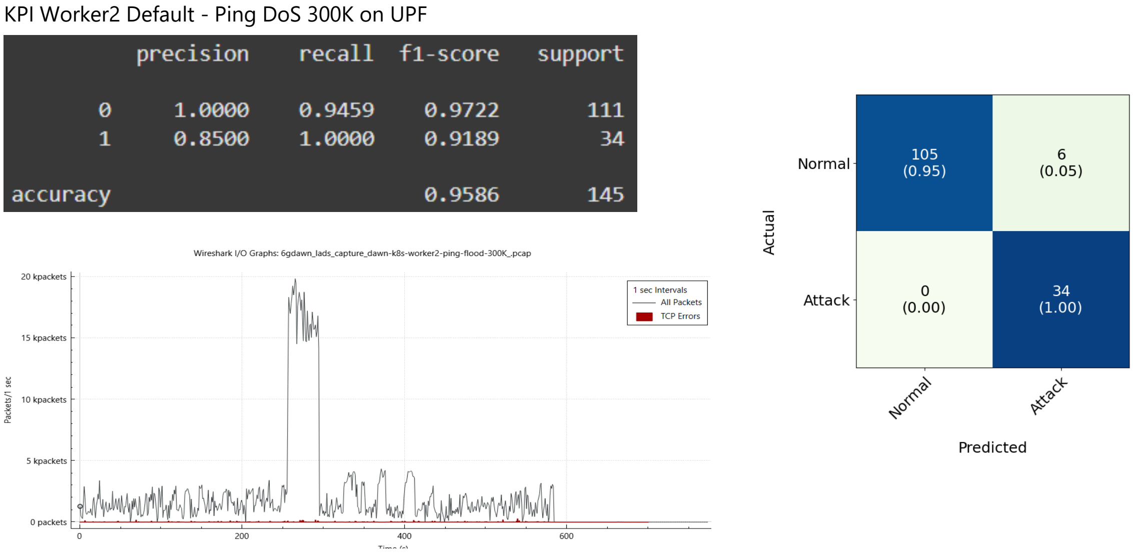
Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya



KPI Worker3 Host - Ping DoS 300K on VStream

```
              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       312
           1     1.0000    1.0000    1.0000        18

    accuracy                         1.0000       330
```

FIGURE 16. RESILIENT UC1 POC1 KPI – WORKER3 HOST - PING DOS 300K ON VSTREAM



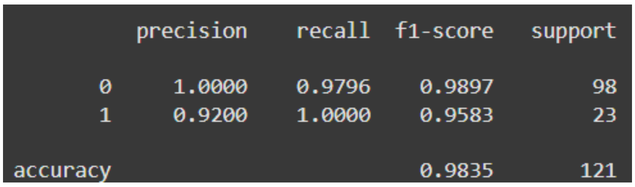
KPI Worker2 Default - Ping DoS 300K on UPF

```
              precision    recall  f1-score   support

           0     1.0000    0.9459    0.9722       111
           1     0.8500    1.0000    0.9189        34

    accuracy                         0.9586       145
```

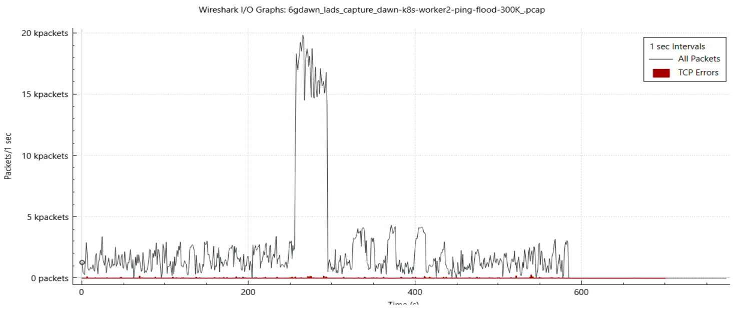
FIGURE 17. RESILIENT UC1 POC1 KPI - WORKER2 DEFAULT - PING DOS 300K ON UPF

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

KPI Worker2 Host - Ping DoS 300K on UPF



```
              precision    recall  f1-score   support

           0     1.0000    0.9796    0.9897        98
           1     0.9200    1.0000    0.9583        23

    accuracy                         0.9835       121
```



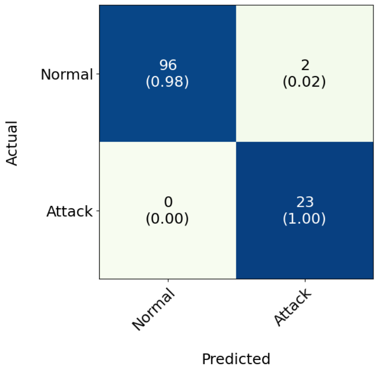


FIGURE 18. RESILIENT UC1 POC1 KPI – WORKER2 HOST - PING DOS 300K ON UPF

2.1.1.3 Achievements and lessons learned

The results for the different attacks shown above are promising for the LADS system. In all attack cases, an **accuracy over 96%** was achieved, with an **F1-Score for legitimate** cases (shown as row 0) exceeding **97%**, and an **F1-Score for anomalous** cases (shown as row 1) on average **90%**. Furthermore, the confusion matrices confirm the good results obtained. In some cases, there are some false positive errors in predicting legitimate traffic as anomalous traffic, but this is an acceptable error. Given that the opposite error, misclassified anomalous traffic as normal traffic, is the most critical error for anomaly detection systems, where **LADS showed detection of all attack instances** with no failure (no false negatives).

We confirm the very fruitful work, collaboration and results achieved in the project and above all in the domain of decentralized intrusion detection for Kubernetes.

We summarize the main achievements:

- Network-anomaly-detection-based assurance of the healthy status of a Kubernetes cluster,
- Full-scale and visibility of network traffic in a Kubernetes cluster,
- Enriched and consistent flow telemetry over highly volatile and dynamic IP addresses,
- Multi-model parallel processing for anomaly detection to scale to the volume and velocity of flow telemetry data produced but the different sensors.

We summarize the main lessons learnt:

- Kubernetes clusters are challenging environments for anomaly-based network intrusion detection due to their highly dynamic nature of containers (pod) deployment over cluster lifetime.

- Kubernetes clusters require suitable adaptation and consistent flow telemetry over the highly dynamic and volatile IP addresses association to pods and services.
- Kubernetes clusters require a suitable training procedure for the machine learning models. Given their dynamic nature and network behavior over time, it is necessary to suitably set up the threshold for anomaly detection per model being trained, and above all to automatically identify the need to collect more training data to minimize false positives and avoid making the threshold too high (permissive).

  The threshold can be more permissive or less permissive depending on the value and volume of data. For instance, if during a week of traffic data collected for training, there are only two or three administrative accesses to a cluster, these will represent an important to model and learn administrative network behavior but will make the deep learning algorithm difficult to learn the pattern (the value) of such behavior. This in turn would require setting up in a higher threshold value to minimize potential false positives of administrative access but will be permissive for other anomalies.

  Alternatively, it is desirable to automate and collect (or synthetically generate) more volume of such administrative access for an extended training and make the deep learning model learn such behavior allowing lower threshold for anomaly detection.

2.1.2 RESILIENT UC1 PoC2 – Trustworthy Federated Learning enabled through smart contracts and Blockchain

The overall goal of this PoC is to improve the trustworthiness of FL model employed 6G DAWN architecture. This PoC seeks to enhance trustworthiness which is achieved through transparency and immutability. The Analytical Engine (AE) serves as the FL client, performing local model training and updates at edge sites. The AE collects data from the Monitoring System (MS), analyzes metrics, and predicts resource requirements, such as CPU load, ensuring accurate predictions with minimal error, such as low Normalized Mean Squared Error (NMSE). On the other hand, the Decision Engine (DE) acts as the FL server or central aggregator. It manages the exchange of model weights between FL clients (AEs) and aggregates these weights to update the global model. This global model is then shared with the AEs for further refinement, creating an iterative learning process. This setup, implemented within a cloud-native infrastructure using Kubernetes and Docker containers, ensures scalability and flexibility in managing distributed learning tasks across multiple sites.

In FL by incorporating a blockchain-based system for meticulous logging of all client-aggregator interactions. At its core, this system employs a smart contract-driven reputation mechanism to monitor and assess the performance of each client rigorously. This assessment results in a reputation score, pivotal for decision-making (where 90% of the best performing AE can be selected for the global model) within the FL framework presented in Figure 19. This score encapsulates each client's historical input and dependability shaping their future contributions to the global model refinement. By integrating this mechanism, the PoC aspires to cultivate an FL environment that is not only transparent and trustworthy but also ensures a balanced and integrity-focused collective learning process.

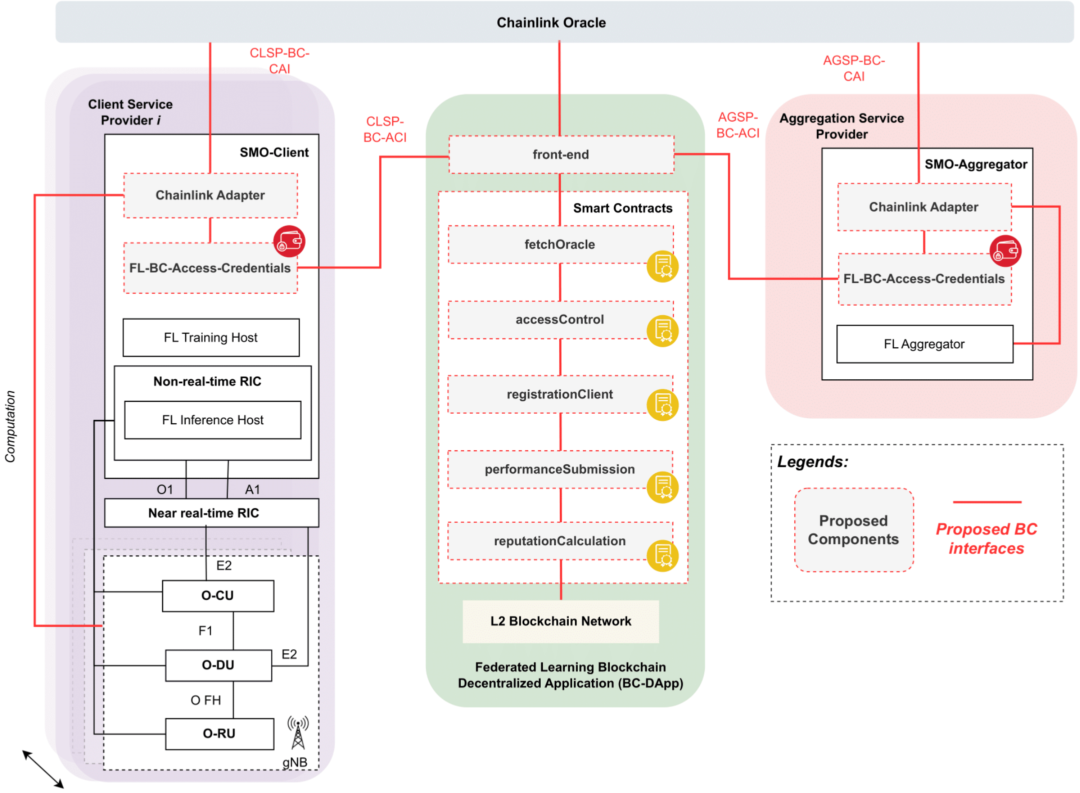


FIGURE 19. PROPOSED FRAMEWORK: FUNCTIONAL BLOCKCHAIN-ENABLED O-RAN ARCHITECTURE FOR TRUSTWORTHY FL USING SMART CONTRACTSPOC IMPLEMENTATION DETAILS

We have implemented our smart contracts using Kubuntu 22.04.3 LTS, complemented by 32 GB DDR4 RAM to enhance processing capabilities. The development framework used is Hardhat 2.22.4, which facilitates the development of Ethereum-based applications, and the smart contracts are developed using Solidity v0.8.0 to ensure secure and efficient coding. These details are presented in Table 6.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

TABLE 6. SIMULATION SYSTEM FOR FL DAPP

Components	Specification
Operating system	Kubuntu 22.04.3 LTS
Memory (RAM)	32 GB DDR4
Blockchain testnet	Amoy testnet
Development framework	Hardhat 2.22.4
Smart contract language	Solidity
Node Configuration	Alchemy node
Number of accounts	51 Externally Owned Accounts (EOA) and their private keys

For our simulations, we utilize the Polygon testnet Amoy, a Layer 2 (L2) scaling solution for Ethereum, which operates on top of the Ethereum mainnet to offer faster transaction speeds and lower gas costs. This test environment mirrors the conditions of the Polygon mainnet, allowing for a realistic assessment of network responsiveness and performance. The choice of Amoy is strategic, enabling extensive testing of smart contracts before deployment in live settings. The simulation involves 51 External Owned Accounts (EOAs) 50 FL clients and 1 server — to simulate different types of interactions within the network. These interactions are demonstrated through three phases of operation: registration of clients, submission of NMSE value, and calculation of reputation scores. Each test was run over 10 iterations to ensure robustness. An essential component of our setup is the *fetchOracle* from Chainlink[8], an oracle contract that, along with a custom external adapter, enables the integration of blockchain smart contracts with external data sources. This setup allows a Chainlink node to retrieve client IDs and their associated NMSE from a predefined mock API, format this data appropriately, and feed it into the smart contract. This process is critical for ensuring that the smart contract can operate with external data in a simulated environment, which is ideal for development and testing.

2.1.2.1 R UC1 PoC2 KPIs Evaluation and Results

Previously in E3 and E4 we presented the KPIs that will be presented for this PoC. Such as Gas Used for each function within this DApp and Latency measurement for these functions described in Table 7.

[8] https://chain.link

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

TABLE 7. SUMMARY OF MAIN SMART CONTRACTS AND THEIR KEY FUNCTIONS

Proposed Contract	Main Function	Functionality of Main Function
registrationClient.sol	registerAsClient()	Registers each CLSP as a client
performanceSubmission.sol	submitNMSE()	Submits weight for a specific round.
reputationCalculation.sol	updateScores()	Updates reputation scores based on provided NMSE values.
	selectTopPerformers()	Selects the top 90% performers based on reputation scores

Gas Used:

Ethereum facilitates the execution of smart contracts through its Ethereum Virtual Machine (EVM), a sophisticated virtual computing environment designed to interpret a low-level, stack-based bytecode language. This language enables the EVM to perform a wide range of computational tasks essential for processing smart contracts. The core of EVM functionality lies in its opcodes, which are fundamental instructions that govern the operations of the virtual machine. Function calls within smart contracts play a pivotal role in their execution, encompassing activities such as computations, data retrievals, and conditional evaluations. Depending on the nature of these operations, a function call may trigger various opcodes that either read or modify the blockchain's state, thereby shaping how the contract interacts with stored data.

A particularly significant opcode in this context is SSTORE (Storage Store), which is responsible for storing data in the blockchain's state. This operation is critical as it involves writing to the ledger, resulting in a permanent change to the contract's state. Given its impact, SSTORE is recognized as a gas-intensive operation, with its high cost reflecting the computational effort required and the consensus process needed across the network to validate the change.

Every transaction that leverages these opcodes incurs gas, a unit measuring the computational workload needed for execution. As part of our testing process, the completion and confirmation of a transaction produce a transaction receipt, which is typically accessed through a statement such as const gasUsed = receipt.gasUsed;. This receipt serves as a comprehensive log of the transaction, capturing key details like the execution outcome and any events triggered during its course. Of particular importance is the gasUsed metric, which quantifies the actual gas consumed. This measurement is essential for evaluating the computational resources utilized and provides valuable insights into the complexity of the smart contract functions deployed within our FL DApp framework.

Transaction Latency:

Latency is a key metric for assessing the performance of blockchain systems. It measures the time taken for a transaction to move from submission to full confirmation on the network. This metric is essential for evaluating the network’s responsiveness and overall efficiency, especially during the execution of smart contracts.

In our tests, latency is measured as the time difference between two key events: the timestamp of the block that confirms the transaction (T_confirmed) and the timestamp when the transaction was initially submitted (T_submitted). The formula used is:

Latency (L) = T_confirmed - T_submitted

This calculation provides a direct measure of the time required for a transaction to be included in a block and fully confirmed. For our analysis, we used the Amoy testnet, a network that closely mirrors the Polygon mainnet, to simulate transaction processing under controlled conditions. By utilizing blockchain-native timestamps, we can accurately measure the network's response time.

To examine factors influencing latency, we conducted tests involving sequential transactions, such as registering CLSPs, and concurrent transactions, such as NMSE submissions by multiple CLSPs. These tests, summarized in the main functions of our FL DApp smart contracts, simulate interactions from multiple accounts. This allows us to evaluate how the network handles varying transaction loads and their effect on overall latency.

Gas Consumption:

TABLE 8. GAS USED DURING INITIALIZATION AND POST-INITIALIZATION PHASES FOR EACH FUNCTION

Function	Phase	Gas Used
registerAsClient()	Initialization	43,464
submitNMSE()	Initialization	113,394
	Post-Initialization	96,294
updateScore()	Initialization	1,333,405
	Post-Initialization	458,481, 458,493, 458,505, 458,517, 458,529, 458,541, 458,553, 458,565, 458,577 (difference of 12)

Gas Usage for Client Registration

The smart contract utilizes the function *registerAsClient()* to enroll clients, consistently consuming 43,464 gas for each transaction. This operation's consistency in gas usage is attributed to the Ethereum Virtual Machine's (EVM) handling of low-level operations, particularly the SSTORE opcode, which facilitates data storage on the blockchain. This opcode is notably expensive as it alters the blockchain's persistent state, necessitating replication across all network nodes to maintain decentralized ledger integrity. In the context of the *registerAsClient()* function, as outlined in Table 8 each invocation updates the blockchain's state by adding a new client's address to the registeredClients mapping. When a new address is recorded, the associated storage slot transitions from a default zero state to a non-zero value, incurring a high gas cost through this state alteration, thereby registering the client on the blockchain.

Gas Consumption for NMSE Submission

As documented in interacting with the *submitNMSE()* function within the *performanceSubmission.sol* smart contract reveals significant gas consumption nuances. The initial submission by a client in a given round ("initialization" phase) consumes 113,394 gas, attributed to setting up several storage structures. This includes initializing an array for storing client addresses and mapping client weights. The initialization of these storage structures incurs higher gas costs as they transition from unused (zero) to used (non-zero) states, which is more gas-intensive. Subsequent submissions within the same round decrease to 96,294 gas, as outlined in Table 8 due to the pre-established primary storage structures, thus requiring less initialization. It's important to note that each client submission is treated as a separate transaction, with storage slots considered "cold" at each access, which maintains higher costs due to repeated initialization across transactions.

Analyzing Gas Usage for Reputation Score Updates

In examining the *reputationCalculation.sol* smart contract, distinct patterns in gas usage emerge, particularly with the *updateScores()* function. The first call of this function in a round involves significant computational effort, consuming 1,333,405 gas as indicated in Table 8. This high cost is due to initializing the storage with NMSEs and updating all 50 clients' reputation scores from zero to non-zero. Subsequent rounds stabilize at around 458,481 to 458,577 gas, reflecting the reduced computational demand for updating existing data entries. Each additional non-zero byte in the NMSE values submitted increases the transaction gas cost slightly, by approximately 12 gas per byte. This increment is a result of the higher costs associated with processing non-zero data compared to zero data in the EVM.

Analysis of Client Registration Latency

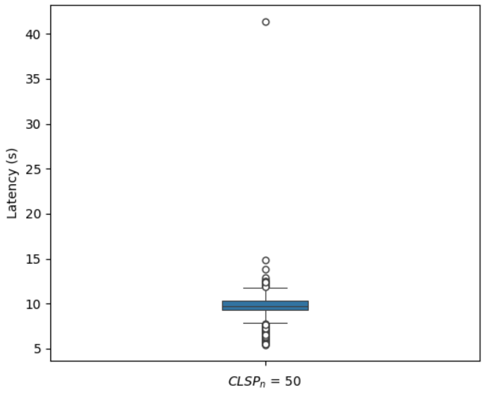


FIGURE 20. LATENCY FOR REGISTRATION RANGING FROM 1-50 CLIENTS

The process for client registration within the FL DApp, managed through the *registrationClient.sol* contract, requires the sequential enrollment of 50 clients via the *registerAsClient()* function, as detailed in referenced works. The latency data, visualized in the boxplot in Table 8, predominantly shows closely grouped registration times with a median near 10 seconds, indicating a compact interquartile range (IQR). This grouping suggests efficient processing, largely due to Polygon's Layer 2 technology, which facilitates batch processing off the main Ethereum network, thus alleviating congestion and minimizing latency.

An exceptional latency occurrence, noted at 41.424 seconds, stands out as an anomaly. Analysis links this outlier to block number 7908352, which not only processed standard smart contract interactions but also generated two internal transactions of the 'create' type. These internal events are not visible in the main transaction list but are instead captured within the transaction receipts as part of embedded smart contract operations. Such occurrences can significantly influence latency, especially under conditions of network strain. Although most registrations are processed expediently, a minority face significant delays due to the complexity and nature of transactions within a block. Despite the general efficiency promoted by Polygon's modified Proof of Stake (PoS) mechanism, the system can still experience slowdowns during periods of heavy transaction loads or intricate transactional activities, which are reflected in the observed latency variations as detailed in the boxplot.

Analysis of NMSE Submission Latency for Clients

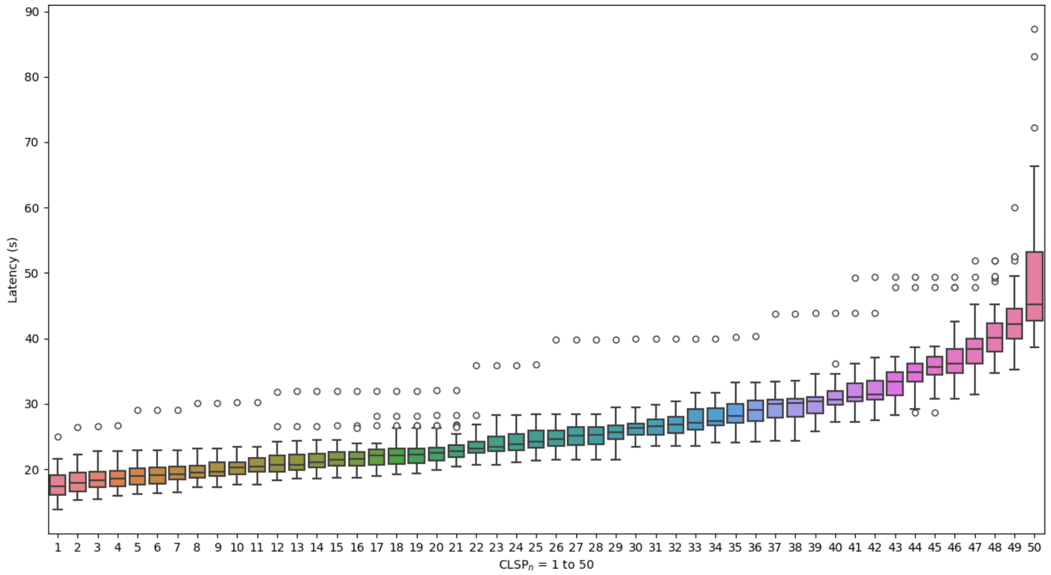


FIGURE 21. LATENCY FOR SUBMITNMSE() FOR CLIENTS RANGING FROM 1 TO 50 DURING FL ROUNDS

This section examines the latency involved in the *submitNMSE()* function, where each client from the FL DApp submits a NMSE value calculated off-chain. These values are incorporated into the *performanceSubmission.sol* smart contract, which handles their submission. Each of the registered clients proceeds to submit their NMSE values for 50 rounds simultaneously through the *submitNMSE()* function.

In Figure 21, we analyze the latency experienced during the transmission of NMSE values across multiple transaction rounds for fifty clients, denoted as $CLSP_1$ to $CLSP_{50}$.

The collective boxplot in Figure 21 displays latency across the 50 clients for each of the 50 submission rounds. Each boxplot corresponds to an individual account $CLSP_i$, illustrating the distribution and central trends of latency times. Although there is a visible trend of increasing latency from $CLSP_1$ to $CLSP_{50}$ it is crucial to highlight that these NMSE submissions are conducted in parallel. The term "simultaneously" here implies that all clients are engaging in the transmission of their data to the server at the same time via the smart contract, activated by the *submitN*$_{MSE}$*()* function.

Latency Analysis for Reputation Score Calculation

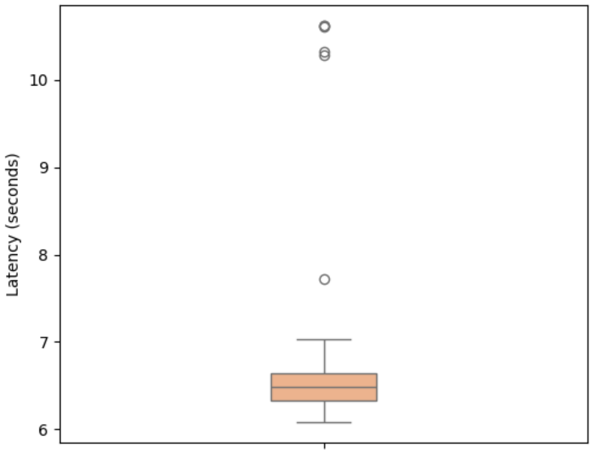


FIGURE 22. BOX PLOT FOR 50 ROUNDS OF REPUTATION SCORE CALCULATION

Figure 22 displays a boxplot providing a detailed look into the latency performance of our blockchain-based reputation management system. The boxplot shows that latency typically centers around 6.5 seconds, with most observations lying between 6.2 and just above 7 seconds. However, there are a few notable outliers extending up to around 9 and 10 seconds, indicating occasional peaks in latency.

The data from the supports these observations, with the average latency noted at approximately 6.7 seconds and the median closely matching the 6.5-second mark. The 95th percentile is observed at about 7.6 seconds, suggesting that most operations are executed under this time, pointing to a generally reliable performance with some rare delays.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

These sporadic latency spikes are often the result of intensive computational tasks, particularly the concurrent processes of calculating reputation scores and selecting top performers. To optimize the efficiency of these operations, we bundle these computations into a single transaction at the conclusion of each cycle. This method minimizes the frequency of on-chain updates, thereby reducing the cumulative burden of gas costs and transaction confirmation delays. The practice of grouping transactions is in line with efficient blockchain management strategies seen in both Ethereum and Polygon ecosystems.

For instance, successful implementations like the 1inch protocol have shown that such aggregation of tasks can decrease total gas expenditures by 20-30%, effectively reducing the cost per individual operation and potentially hastening transaction confirmation by alleviating network congestion.

In addition to cost savings, this approach to consolidating tasks can yield more uniform latency figures. A reduction of 15-25% in median latency has been documented when comparing this bundled processing approach to executing each task separately. This synchronization of reputation scoring and top-performer selection not only ensures that rankings are based on the latest data for better accuracy and integrity but also simplifies the update process. By limiting the complexity and frequency of state modifications, we reduce the potential for errors or discrepancies, thus enhancing the overall reliability and robustness of the reputation management system.

2.1.2.2 Achievements and lessons learned

We have detailed the architectural framework and the blockchain implementation for integrating FL in O-RAN environments. Our strategy utilizes Polygon's Layer 2 solutions to develop a DApp designed for managing and validating machine learning model training and data exchanges across a multi-vendor landscape to support trusted collaborative learning.

In our implementation, we focus on several critical smart contract functions, including registerAsClient(), submitNMSE(), and updateScores(). For each of these functions, we examine and report on various blockchain parameters such as the average, standard deviation, and median of block size, total gas used by the block, number of transactions, gas prices, and latency. This analysis provides insights into the efficiency and effectiveness of our blockchain approach in handling these operations within the FL framework.

While Polygon effectively reduces latency through its off-chain transaction processing capabilities, our findings indicate certain challenges in scenarios of high concurrency. These observations emphasize the necessity for ongoing enhancements to blockchain infrastructure to meet the rigorous demands of Federated Learning (FL). Overall, they prompt critical questions about the deployment of FL clients, their trustworthiness within the system, and strategies to enhance their reliability. A promising solution to these challenges is the adoption of Zero-Knowledge Machine Learning (ZK-ML) methods, which ensure accurate execution of algorithms by system components. ZK-ML principles strengthen FL by enabling decentralized and collaborative model training without the need for direct data sharing, thus ensuring consistent execution and enhanced security against threats. A significant application of this method is the ezkl library, which transforms TensorFlow or PyTorch computational graphs into zero-knowledge proofs (ZK-SNARK circuits). This allows FL clients to verify computations on private data securely. While our PoC touches on the potential future applications of ZK-ML, an in-depth exploration of this topic is outside the scope. Finally, in FL environments characterized by high concurrency and trust challenges, a Service Orchestrator augmented with Grover Search Algorithm can optimize the selection and scheduling of trustworthy FL clients by efficiently searching through the vast space of potential configurations and participant nodes. Grover Search[9], known for its quadratic speedup in unstructured search problems, can accelerate decision-making processes in selecting optimal client sets or verifying ZK-ML proof integrity across multiple nodes.

2.2 Use case RESILIENT Network Resiliency vs QoS and Energy Efficiency

2.2.1 R UC2 PoC1 - Detecting and Isolating Energy Consumption-Heavy Users

The complexity and scale of Non-Public Networks (NPN) and emerging 6G networks pose significant challenges in maintaining optimal system performance. Anomalies in network behavior can indicate issues such as hardware malfunctions, configuration errors, or potential security threats. If left undetected, these anomalies could degrade network performance, cause service disruptions, and introduce security vulnerabilities. Therefore, a robust framework is essential for efficiently detecting and analyzing anomalies within NPNs to enhance operational efficiency, reduce downtime, and improve the reliability of 6G networks. The initiative to "Detect and Isolate Energy Consumption-Heavy Users in Non-Public Networks (NPN) with Digital Twin" aims to enhance network resiliency through a semi-closed loop system. This system integrates several components: real-time monitoring of energy consumption & traffic patterns, anomaly detection analytics, user isolation using network APIs, and policy creation for behavior management. The implementation leverages the 6G DAWN architecture framework, a Network Digital Twin Platform, and an Application Function to detect and manage energy-related anomalies.

The overall goal of this PoC is to improve the NPN Network Resiliency by deploying a semi-close loop consisting of the following components:

- Monitoring: for real-time energy consumption and performance data
- Analytics: for performing anomaly detection (for heavy energy users) and alerting the relevant services

[9] Grover, L. K., *"Quantum Mechanics Helps in Searching for a Needle in a Haystack,"* Physical Review Letters, vol. 79, no. 2, pp. 325–328, 1997.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

- Isolation: leveraging on Network API procedures for user allocation to quarantine slices
- Policy: creation of rules for managing the inspected behavior

This semi-close loop is achieved via the different models involved in the operation of the 6G DAWN AE, implemented via the NDT, which can make predictions. More concretely, in this PoC, it will be demonstrated how the 6G DAWN framework can enable the detection and handling of anomalous energy-related events to reduce their negative impact on the network performance and availability.

2.2.1.1 PoC Implementation Details

2.2.1.1.1 Architecture and Components

The primary elements of this PoC architecture are:

- **NPN System** is deployed at the network edge on the customer premises, in this case at the CTTC Lab in Castelldefels, Barcelona, and is composed of the RAN and UPF.
- **Public Network (PN)** is deployed at the 5TONIC Lab in Leganes, Madrid. It houses the control plane network functions and UPF for central eMBB slice services integrated with the NPN.
- **NPN Network Digital Twin (NDT) Platform** is a key enabler that serves as a digital replica tightly coupled with the deployed physical NPN system. Digital twins open a virtual world of possibilities—a safe, simulated testing environment where you can explore 'what-if' scenarios to your heart’s (or training model’s) content, with no risk to the real-world counterpart. The NDT platform leverages network domain knowledge (theoretical model), automated measurement data (empirical model), and field/operations data (trained model). The NDT platform is developed to provide a Monitoring Engine service for monitoring real-time energy consumption and performance data of the NPN system. It also provides Analytics Engine services for performing anomaly detection (for heavy energy users) and alerting relevant services for interested Application Functions. Additionally, it provides Isolation services for moving specific subscribers to quarantine slices to reduce the impact on the network.
- **Network Exposure Function (NEF)** interacts with NDT platform, PCF and other core networks functions to expose their services to AFs to support various vertical use cases.
- **Application Function (AF)**, a crucial facilitator for implementing vertical use cases. It influences PNI-NPN characteristics, such as energy consumption optimization, by interacting with the Network Digital Twin via NEF. Notably, PNI-NPN covers multiple technological domains (RAN, Edge, and Core), enabling the Application Function to influence various technological domains.

Financiado por la Unión Europea NextGenerationEU · Plan de Recuperación, Transformación y Resiliencia · UNICO I+D · GOBIERNO DE ESPAÑA · cttc Centre Tecnològic de Telecomunicacions de Catalunya

PNI-NPN Architecture – 5Tonic

- Control Plane + User Plane (Public slice) @ Central Office
- User Plane (Private slices) + Radio Access + vAPPs @ Far Edge

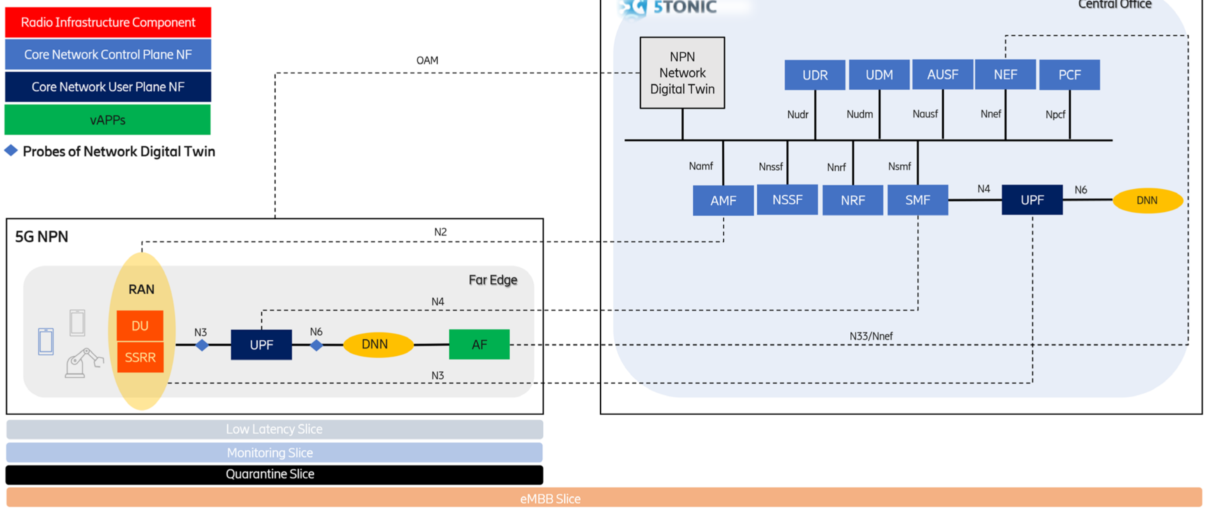


FIGURE 23 PNI-NPN

Network Slicing

Ericsson has developed and implemented a Network-slicing blueprint consisting of 5 slices with the following characteristics.

- **URLLC Slice**: intended for use cases where data timeliness is the relevant quality parameter. The concept, for this type of slice, focuses on a combination of low latency (single-digit milliseconds) and high reliability (prioritized traffic), with services being provided in-house by the Enterprise customer (with a local UPF + local application server (AS)).
- **eMBB Local Slice**: in general, intended for use cases related to human-centric and enhanced access to multimedia content, services, and data with selected balance of speed and capacity. Here the defining quality parameter is the data rate. In the NPN scenario, this slice also keeps a local user plane.
- **eMBB Central Slice (Internet)**: same performance profile as the Local slice. However, the concept focuses on having both, Control plane and User Plane rely on NFs located in the CSP central DC.
- **Quarantine Slice**: after anomaly detection (i.e., high energy consumption users), the network/AF might decide to reallocate subscriber for further analysis.
- **Monitoring Slice**: for CSP use only, i.e., internal performance testing.

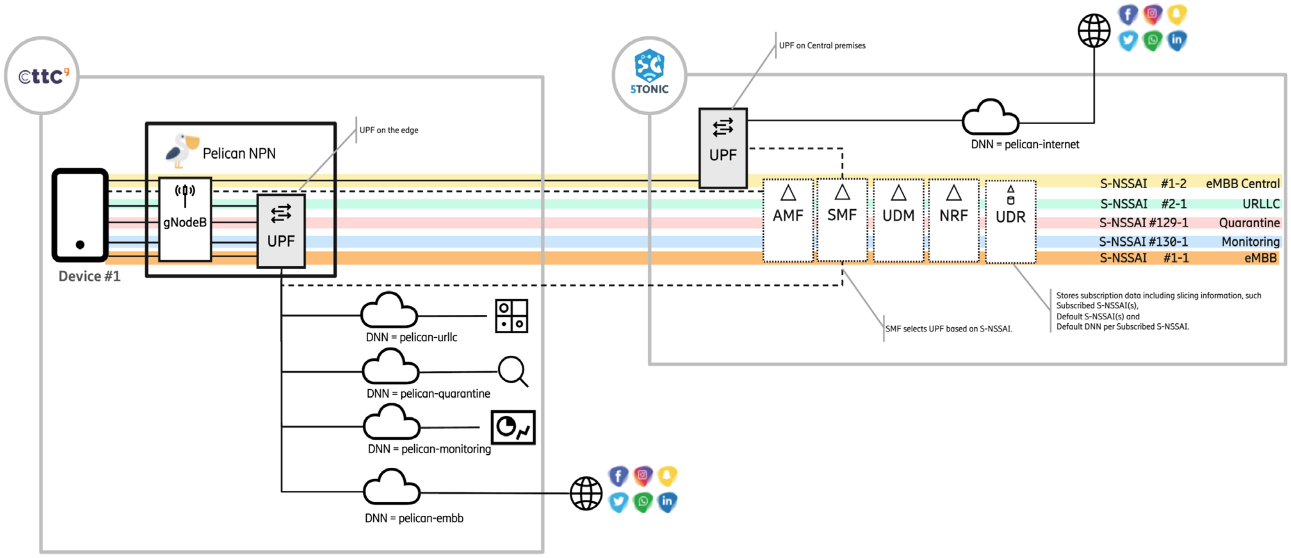

cttc
Pelican NPN
UPF on the edge
gNodeB
UPF
Device #1
DNN = pelican-urllc
DNN = pelican-quarantine
DNN = pelican-monitoring
DNN = pelican-embb
5TONIC
UPF on Central premises
UPF
DNN = pelican-internet
AMF
SMF
UDM
NRF
UDR
S-NSSAI #1-2 eMBB Central
S-NSSAI #2-1 URLLC
S-NSSAI #129-1 Quarantine
S-NSSAI #130-1 Monitoring
S-NSSAI #1-1 eMBB
SMF selects UPF based on S-NSSAI.
Stores subscription data including slicing information, such Subscribed S-NSSAI(s), Default S-NSSAI(s) and Default DNN per Subscribed S-NSSAI.

FIGURE 24 PNI-NPN SLICE SETUP

2.2.1.1.2 PoC overview

The overall functionality of the PoC is shown in Figure 25 as a sequence diagram. Here is a summary of the steps involved and depicted in this diagram:

- **Initial Setup**: the NPN is running with a specific configuration and a defined traffic model, accommodating both URLLC and eMBB slices, in addition to the 3 other slices (eMBB central, Quarantine and Monitoring as per section 2.2.1.1.1).
- **Data Collection**: network probes continuously collect data, and the NDT Platform (Monitoring System & Analytics Engine) derives KPIs from this data. Application probes / monitoring systems also collect data, which is analyzed and used by the AF to derive behavior baselines.
- **Anomalous User Detection Initiation**: an anomalous User Equipment (UE) connected to the network is determined by the Network anomaly detection system of the NPN to be responsible for disturbing the Key Performance Indicators (KPIs) of the NPN. These findings are exposed to the AF, which initiates a more in-depth auditing process. Alternatively, the AF may continuously audit the traffic and monitor possible anomalous behavior.
- **Anomaly Decision**: the AF uses multiple domain-level information collected previously (like traffic flow details and context data) to analyze and decide on the presence of an anomaly via its application-aware anomaly detection system (see section 2.2.1.2.1).
- **Isolation Process**: if an anomaly is confirmed and pinpointed to a specific UE or set of UEs, the AF makes a decision to isolate them. The isolation command is sent through the network components, including NEF, NDT Platform, and NPN. The anomalous UE(s) are then quarantined and moved to a quarantine slice for further investigation.
- **Post-Isolation Analysis**: the AF uses analytics information to monitor and confirm improvements in KPIs following the isolation of the anomalous user. Upon manual or

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

automatic correction of the root causes of this anomalous behavior, or upon the determination that the behavior does fall under expected patterns after further investigation, the AF may decide to remove the policy and move the UE / set of UEs back to their default slice.

This process leverages digital twin technology to simulate and analyze network behavior, enabling proactive management of network resources by identifying and isolating users who consume excessive energy and disrupt network performance. The implementation details of the most relevant phases are further detailed in the sections below.

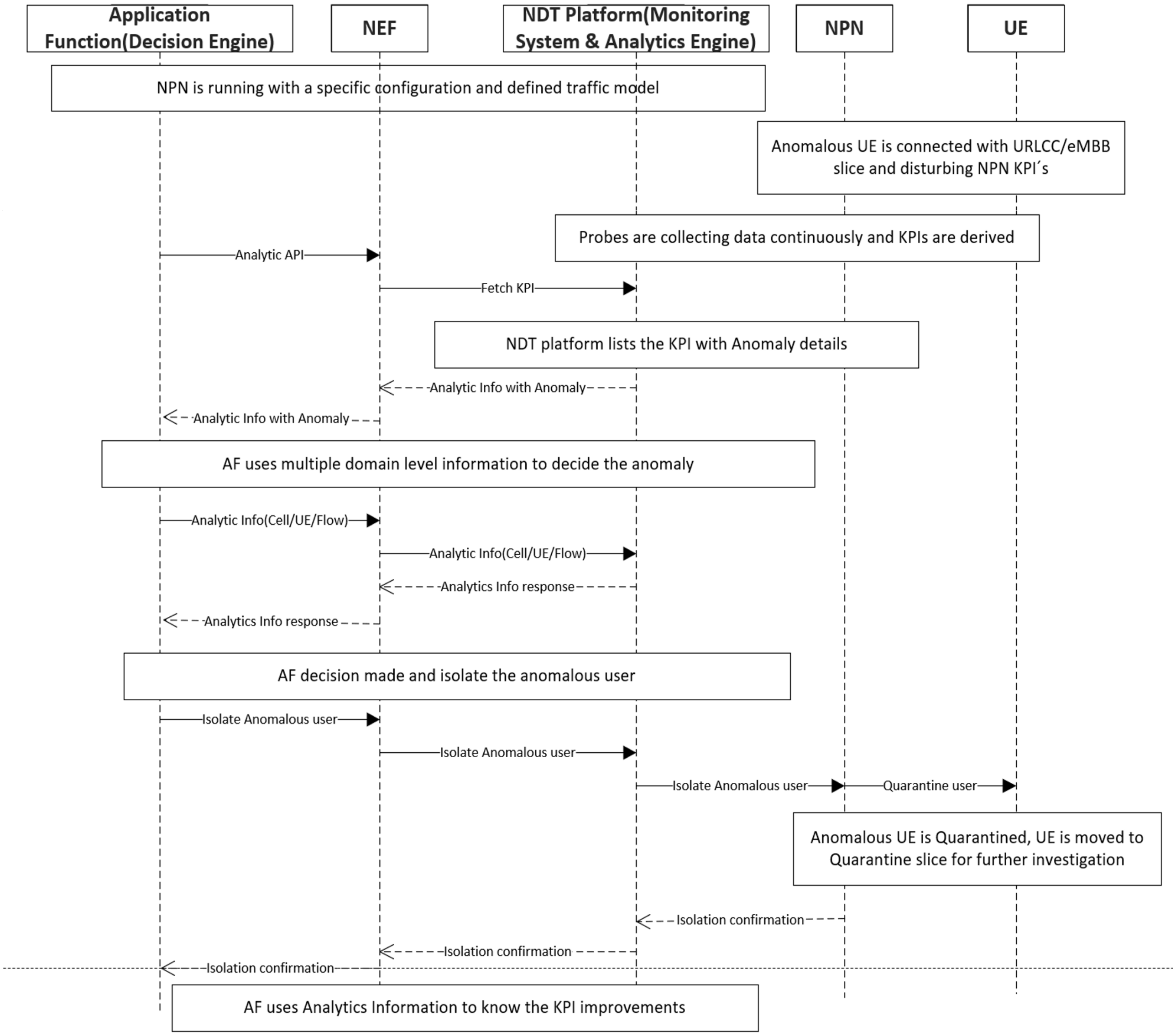

Application Function(Decision Engine)
NEF
NDT Platform(Monitoring System & Analytics Engine)
NPN
UE
NPN is running with a specific configuration and defined traffic model
Anomalous UE is connected with URLCC/eMBB slice and disturbing NPN KPI´s
Probes are collecting data continuously and KPIs are derived
Analytic API
Fetch KPI
NDT platform lists the KPI with Anomaly details
Analytic Info with Anomaly
Analytic Info with Anomaly
AF uses multiple domain level information to decide the anomaly
Analytic Info(Cell/UE/Flow)
Analytic Info(Cell/UE/Flow)
Analytics Info response
Analytics Info response
AF decision made and isolate the anomalous user
Isolate Anomalous user
Isolate Anomalous user
Isolate Anomalous user
Quarantine user
Anomalous UE is Quarantined, UE is moved to Quarantine slice for further investigation
Isolation confirmation
Isolation confirmation
Isolation confirmation
AF uses Analytics Information to know the KPI improvements

FIGURE 25. POC SEQUENCE DIAGRAM

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

2.2.1.1.3 Network Anomaly Detection System

Model

Based on the data collected and processed by the NDT from the different domain knowledge areas, the network-anomaly detection system is capable of reporting unexpected behavior to subscribers (or more specifically, to external functions that retrieve and parse the available analysis reports produced by the NDT via the NEF interface). Currently, the anomaly reporting algorithm for traffic-related KPIs has been implemented via Z-scoring for given KPIs. By standardizing monitored data relative to its mean and standard deviation, z-scores can provide a normalized metric that highlights deviations from typical behavior for both traffic-related KPIs (such as RAN Downlink Throughput or RAN Uplink Throughput) and energy-related KPIs (such as RAN Energy Consumption or RAN Energy Consumption Per Byte).

The advantage of z-scores over other alternative models is that they are easy to compute and interpret, making them ideal for real-time anomaly detection in complex, dynamic environments like cellular networks, where energy consumption patterns can vary across different components and use cases. By leveraging z-scoring, network operators can proactively address energy anomalies, optimize resource allocation, and ensure sustainable network operation. Furthermore, more detailed analysis of reported anomalies, using domain knowledge, can be delegated to the application-aware anomaly detection system.

Reporting format

For those KPIs that support the reporting of anomalies via z-scores, an additional field is added to their schema. Apart from providing the average and standard deviation a 'potentialAnomalies' field is also included. The 'potentialAnomalies' field is a map with the following key-value pair:

- Key: the exact date of the anomaly.
- Value: the anomaly throughput value.

This map, encoded as a list of tuples as shown in Figure 25, represents the anomalies in the associated KPIs in the time frame selected in the request with their exact date when they happened. The formula used to calculate these anomalies is:

$$Anomaly = mean - (Z \cdot sttdev)$$

- Z: a factor to define the window where all traffic outside is considered anomaly. The factor used is Z = 2.
- stddev: the standard deviation for that period defined in the request.
- mean: the mean throughput for that period defined in the request.

In summary, any KPI value that falls outside the range of two standard deviations from the mean is considered an anomaly.

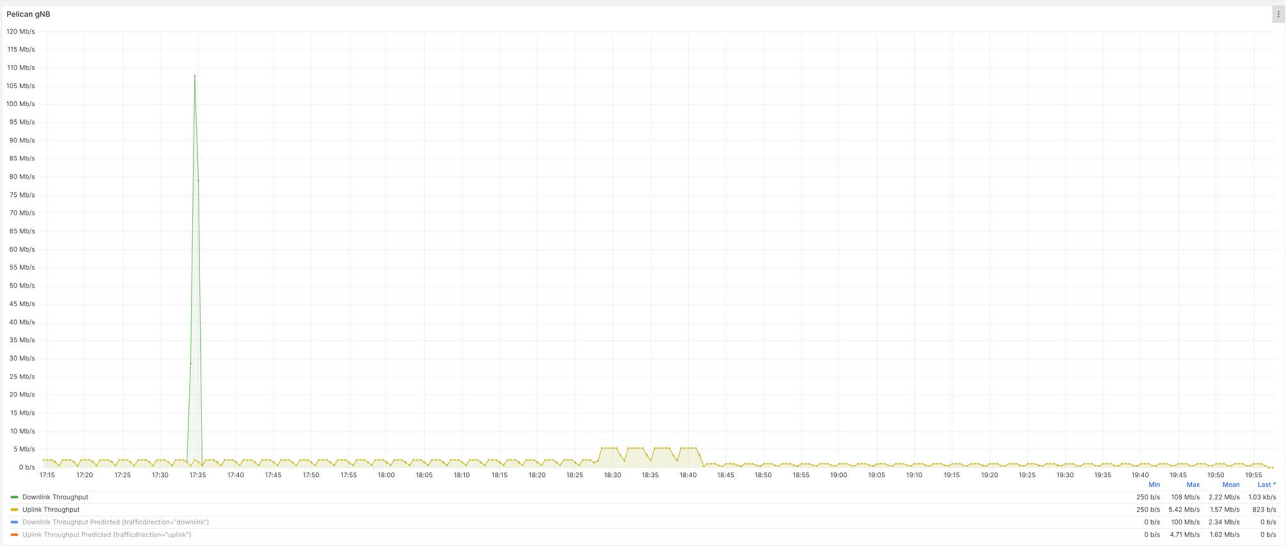
Financiado por la Unión Europea NextGenerationEU · Plan de Recuperación, Transformación y Resiliencia · UNICO I+D · Gobierno de España · cttc Centre Tecnològic de Telecomunicacions de Catalunya

Example use case

For better understanding the behavior and characteristics of the implemented network anomaly detection system, let us consider an illustrative practical scenario. Here, we will use the 'RanDownlinkThroughput' KPI as an example due to it being much easier to visualize with respect to energy-related KPIs, which need to be aggregated and processed over significantly longer periods of time to produce meaningful results.

The NPN first collects data based on an exhaustive measurement campaign. The experiment datasets generated (testing over 20 distinct configs and 130 different traffic models, with almost 40K KPI measurements per KPI being retrieved) are automatically collected and fed to ML models to update the training and prediction datasets.

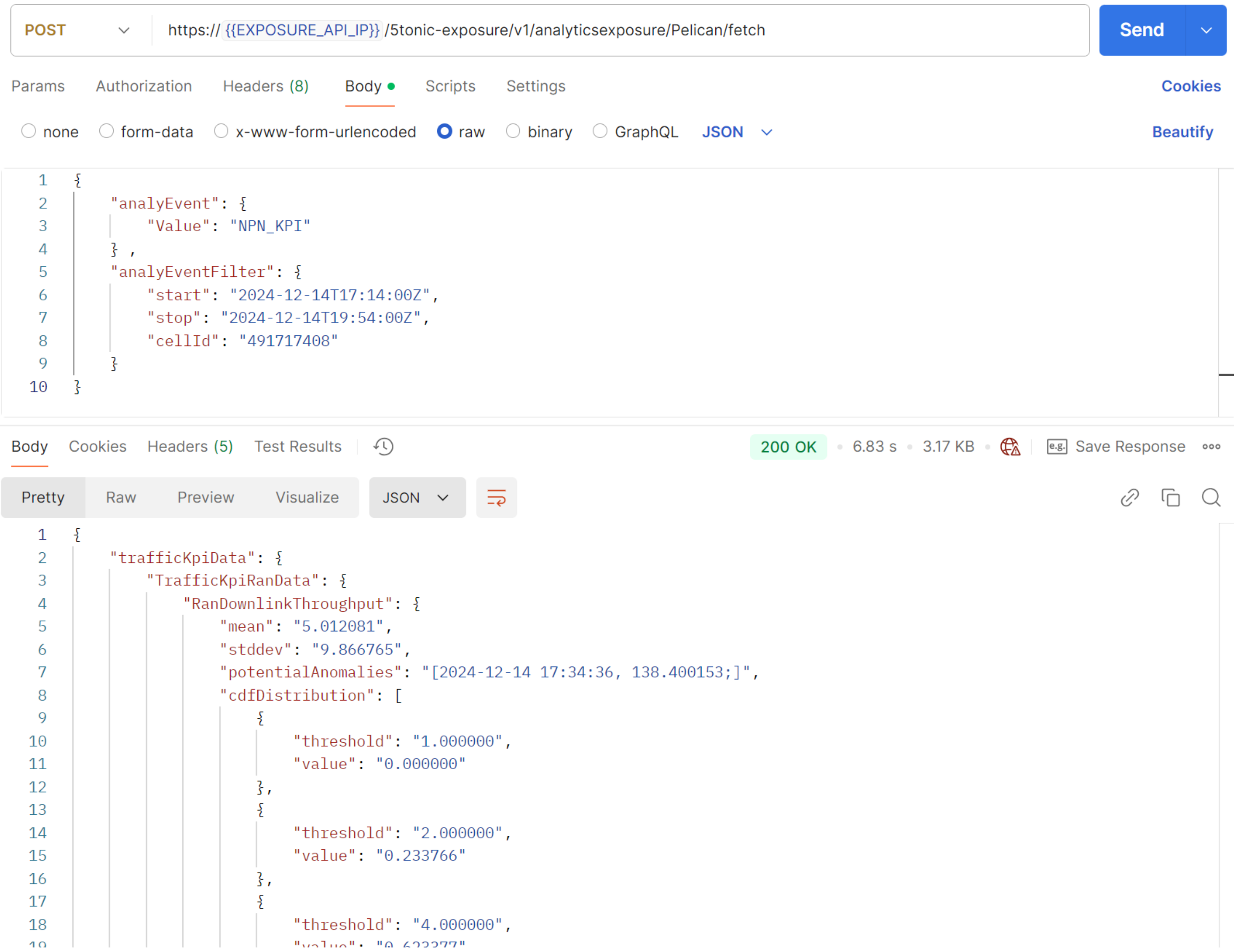
Using this information as the baseline, when the NPN detects unusual network behavior (where a high, unexpected rise of traffic in the Downlink channel – one not previously seen as normal within the train data – can be seen around the 17:35 timestamp) it registers internally this event, marking it as anomalous.



	Min	Max	Mean	Last *
🟩 Downlink Throughput	250 b/s	108 Mb/s	2.22 Mb/s	1.03 kb/s
🟨 Uplink Throughput	250 b/s	5.42 Mb/s	1.57 Mb/s	823 b/s
🟦 Downlink Throughput Predicted {trafficdirection="downlink"}	0 b/s	100 Mb/s	2.34 Mb/s	0 b/s
🟧 Uplink Throughput Predicted {trafficdirection="uplink"}	0 b/s	4.71 Mb/s	1.62 Mb/s	0 b/s

FIGURE 26. ANOMALOUS BEHAVIOUR

Then, external applications (such as the application-based anomaly detection system), can periodically (or based on given events) retrieve this information via the use of Analytics API (implemented as an extension of the NEF interface). This information could be retrieved at a cell-level (as shown in Figure 26), where the anomaly can be seen included as part of the 'potentialAnomalies' field, with timestamp 17:34:36) or even at a UE-level, that is, all anomalies related with a given user specified via the tgtUe filter field (as shown in Figure 27)



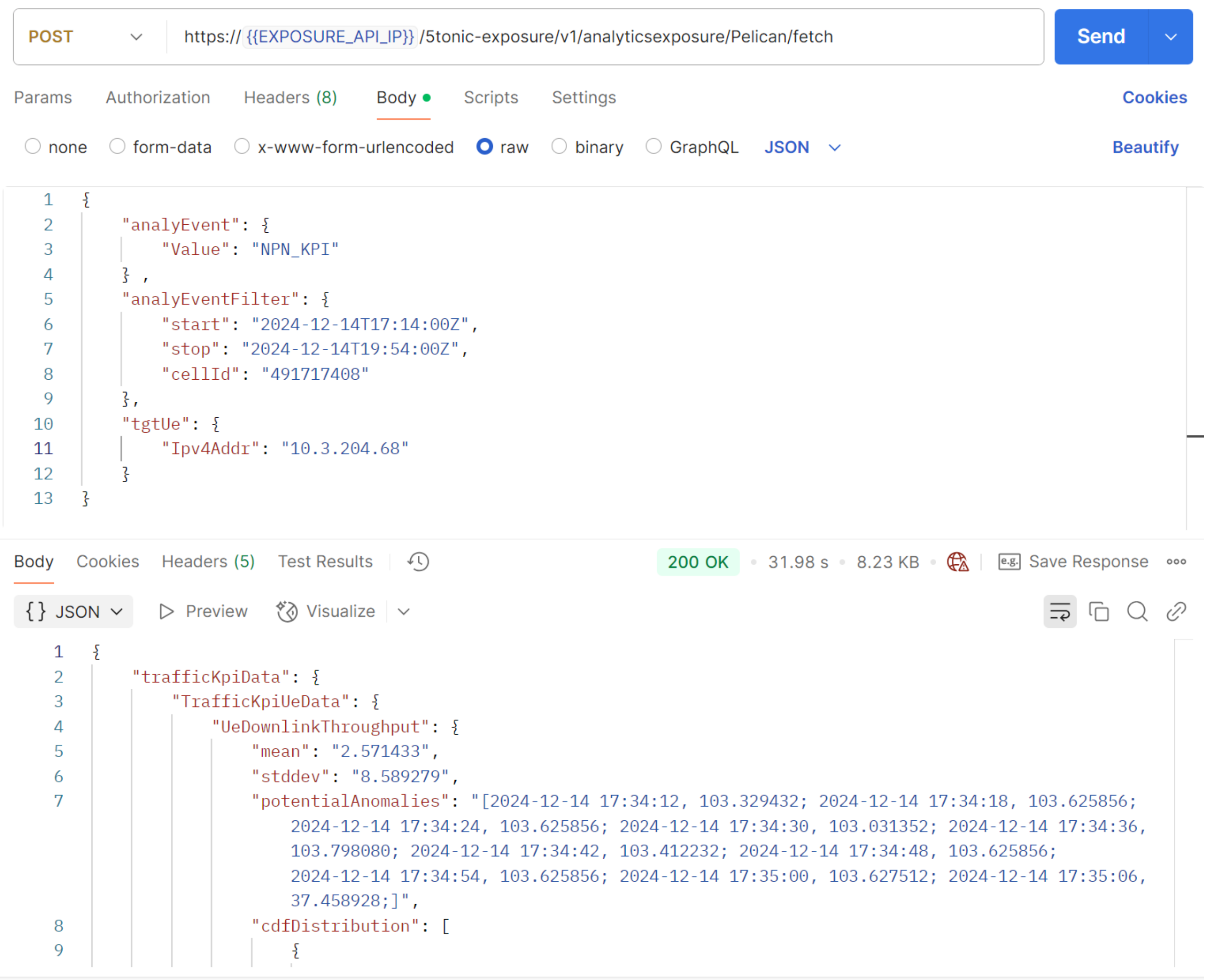
POST https:// {{EXPOSURE_API_IP}} /5tonic-exposure/v1/analyticsexposure/Pelican/fetch

```
{
    "analyEvent": {
        "Value": "NPN_KPI"
    } ,
    "analyEventFilter": {
        "start": "2024-12-14T17:14:00Z",
        "stop": "2024-12-14T19:54:00Z",
        "cellId": "491717408"
    }
}
```

200 OK • 6.83 s • 3.17 KB

```
{
    "trafficKpiData": {
        "TrafficKpiRanData": {
            "RanDownlinkThroughput": {
                "mean": "5.012081",
                "stddev": "9.866765",
                "potentialAnomalies": "[2024-12-14 17:34:36, 138.400153;]",
                "cdfDistribution": [
                    {
                        "threshold": "1.000000",
                        "value": "0.000000"
                    },
                    {
                        "threshold": "2.000000",
                        "value": "0.233766"
                    },
                    {
                        "threshold": "4.000000",
                        "value": "0.623377"
```

FIGURE 27. ANOMALY DETECTION AT THE CELL LEVEL



POST https:// {{EXPOSURE_API_IP}} /5tonic-exposure/v1/analyticsexposure/Pelican/fetch

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "analyEvent": {
        "Value": "NPN_KPI"
    } ,
    "analyEventFilter": {
        "start": "2024-12-14T17:14:00Z",
        "stop": "2024-12-14T19:54:00Z",
        "cellId": "491717408"
    },
    "tgtUe": {
        "Ipv4Addr": "10.3.204.68"
    }
}
```

200 OK • 31.98 s • 8.23 KB

{} JSON

```
{
    "trafficKpiData": {
        "TrafficKpiUeData": {
            "UeDownlinkThroughput": {
                "mean": "2.571433",
                "stddev": "8.589279",
                "potentialAnomalies": "[2024-12-14 17:34:12, 103.329432; 2024-12-14 17:34:18, 103.625856;
                    2024-12-14 17:34:24, 103.625856; 2024-12-14 17:34:30, 103.031352; 2024-12-14 17:34:36,
                    103.798080; 2024-12-14 17:34:42, 103.412232; 2024-12-14 17:34:48, 103.625856;
                    2024-12-14 17:34:54, 103.625856; 2024-12-14 17:35:00, 103.627512; 2024-12-14 17:35:06,
                    37.458928;]",
                "cdfDistribution": [
                    {
```

FIGURE 28. ANOMALY DETECTION AT THE SUBSCRIBER LEVEL

2.2.1.1.4 Application-aware anomaly detection system

The application-aware anomaly detection system's role is to use the service operator's domain knowledge concerning the services managed by it to detect possible offenders once the operator has been alerted of anomalous KPI values (such as energy consumption at different infrastructure components / levels) observed by the NPN at a large-scale / global level.

Network model

Developing a robust traffic model for communication among public safety agents during emergency situations presents significant challenges, particularly in sourcing datasets. The availability of relevant and comprehensive data is often constrained by several factors, including privacy concerns, the sensitive nature of emergency operations, the fragmented systems used by different agencies and the lack of sophisticated monitoring systems deployed by these agencies to produce this kind of insights. Public safety communications data, such as logs from dispatch systems or radio traffic patterns during crises, is rarely centralized or openly shared.

Most existing datasets come from limited sources, such as simulated scenarios, specific agency studies, or proprietary research initiatives. For example, resources like NIST's Public Safety Communications Research (PSCR) usability survey [10](which includes, among other things, how often agents from several agencies use devices) and FCC's reports on the telecommunications infrastructure and use provide some insights, but they are too high level, not providing enough detailed usage logs to train and test an anomaly model.

Therefore, due to the lack of available datasets, the designed model has been trained and evaluated using synthetic data (see section 'Dataset generation' for implementation details), selecting an unsupervised learning algorithm as the ML model to combat the lack of labeled data. Furthermore, the synthetic data has been based on the technical case study performed on the communications systems by FCC during the handling of a real-life incident-response, where the FCC recorded communications usage and performance metrics, together with the estimation of specific traffic / application services from the agents in emergency situations as provided by the NYCDIT[11].The results of this analysis, considering video quality transmitted at 512kbps streams and a density of 21 agents per sector, is summarized in Table 9

TABLE 9. TRAFFIC MODEL

Type of device	Agents with device (%)	Total devices	UL (kbps/device)	DL (kbps/device)	Time transmitting	Time receiving
Mobile video camera	25%	5	256	12	10%	5%
Data file transfer CAD/GIS	87%	18	50	300	15%	5%
VoIP	100%	21	27	27	5%	15%
Secure file transfer	12%	3	93	93	5%	5%
EMS patient tracking	6%	1	30	50	10%	5%
EMS data transfer	6%	1	20	25	25%	5%
EMS Internet access	6%	1	10	90	10%	5%

[10] https://publicsafety.nist.gov/survey.html

[11] New York City Filing, FCC Docket 07-114, New York City Department of Information and Technology (NYCDIT), (Nov. 17, 2009)

Command unit DL video	-	6	0	512	0%	100%
Command unit UL video	-	2	512	0	100%	0%

Anomaly detection system

For detecting traffic-related anomalies, some of which may also induce unexpected declines in energy efficiency, a One-Class Support Vector Machine (OCSVM) based algorithm has been implemented for the main detection system due to the limited availability of labelled network data for the type of public protection and disaster relief scenarios analyzed in the previous section, and due to the high variability of the expected traffic patterns in these scenarios. OCSVM, with its kernel-based approach, can model nonlinear relationships in such high-dimensional data effectively, and showed better results in the employed datasets with respect to other tested models such as k-means. The adaptive nature of OCSVM allows it to adjust to shifts in network traffic patterns over time while maintaining a baseline for anomaly detection, which is key for successfully adapting to pattern changes in traffic flows due to the introduction of new services or of new response procedures. Finally, its computational efficiency ensures that it can scale to large datasets / traffic flows, while still being able to detect anomalies in near real-time, a key requirement for rapidly responding to possible anomalies and minimizing their impact.

More concretely, the detection system has been designed as a two-step algorithm composed by an initial Random Forest classifier, which classifies the observed traffic flows in different service / application types, and then service-specific OSVM models trained for each of the previous service classes which categorize the traffic as either anomalous or non-anomalous as shown in Figure 29. In the first step, the classifier categorizes the traffic flows into the different traffic services identified in Table 9 based on their characteristics. Using the dataset generated with the traffic simulator (as described in the next section), this classification was performed based on a register mapping traffic endpoints to the different service types. In a production system, this could be done in a similar way using an actively maintained registry, or by using the traffic characteristics (source/destination IP(s), ports, transport protocol, traffic patterns, etc.) to perform this classification based on heuristics or a trained model. Then, in the second step, the per-service OSVM models process each of the traffic subsets belonging to the same class and predict, based on the trained data, if any of the involved devices is exhibiting abnormal behavior.

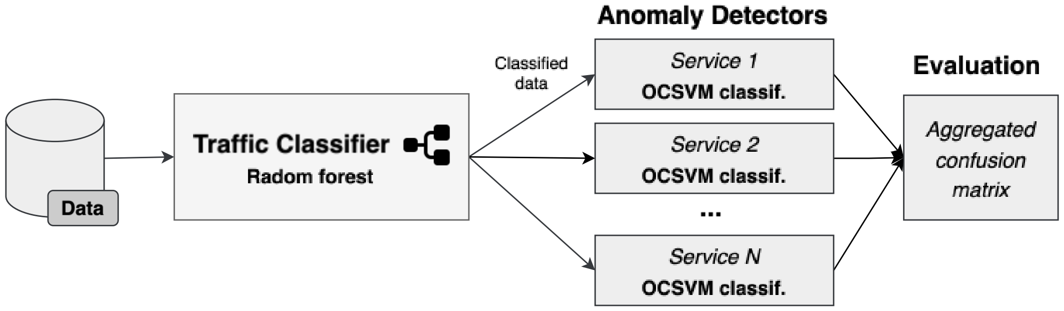


FIGURE 29. MODEL PIPELINE

Dataset generation

As previously mentioned, due mainly to the lack of labeled data describing baseline and anomalous traffic in real-life PPDR scenarios, synthetic traffic generation has been used to validate the performance of the developed model. It also enables scalability and provides flexibility in the creation of controlled scenarios to test multiple scenarios, additionally allowing for the emulation of devices.

To recreate complex and more realistic scenarios, where a large number of devices are involved, the simulator can generate real traffic at both the physical endpoints of the testbed (the end-devices connected to the B5G NPN via the CPEs at CTTC) and at container-based emulated network devices deployed within other servers connected to the network. More concretely, the emulated devices used to recreate the traffic model of the previously analyzed PPDR scenario were deployed in the Application Server on the NPN, which had connectivity with the other physical end-devices via the testbed's wireless B5G interface.

Using a parameter-based configuration, the implemented traffic simulator is capable of simulating different types of traffic (e.g. TCP, UDP) over several intervals by adjusting the upload and download speeds, as well as the transmitting and receiving times and the different possible communication matrix between devices. These adjustments are calculated by applying a random deviation to the baseline speeds and times provided in the configuration. The total upload and download traffic values are then computed by multiplying the adjusted speeds by the time spent transmitting or receiving data.

After generating normal traffic data, the inject_anomalies method within the traffic simulator is then used to introduce irregularities into the dataset. This is done probabilistically, whereas, for example, 10% of the entries are altered to simulate abnormal behavior. Anomalies can occur at the upload and download speeds, as well as the transmitting and receiving times or in the traffic patterns.

Model implementation and evaluation

The final implemented application-aware anomaly detection system starts by loading the configuration file and the dataset. The configuration file, which was mentioned previously and is generated via the *DatasetGenerator*, contains information about the types of devices and their traffic characteristics. The dataset is loaded from a data file and processed to facilitate access to important columns.

- Renaming Columns**:** the columns in the configuration file are renamed for easier reference, using labels such as '*UL_speed*', '*DL_speed*', '*transmitting_time*', and '*receiving_time*'.
- Dataset Factorization**:** to handle categorical data, the device_type column, which holds the type of device (e.g., mobile, VoIP), is factorized into numerical values. This step is essential since machine learning algorithms like OCSVM require numeric inputs.
- Feature Selection: the features used for training the model include relevant traffic characteristics such as the device type, transmitting and receiving times, and upload/download speeds. These are extracted into the feature matrix, while the labels indicating whether a data point is "Normal" or "Anomalous" are stored in output vector "y".

The generated dataset is then split into two subsets:

- Training set (80%): used to train the model. Importantly, only the samples labeled as "Normal" in *y_train* are used to train the OCSVM model. This is a typical approach for anomaly detection since OCSVM is a semi-supervised model that learns from normal data.
- Test set (20%): reserved for testing and evaluating the model's ability to detect anomalies.

Once the data is preprocessed, each of the OCSVM models are trained via the *fit* method on the training data that contains only "Normal" samples. During this training phase, each model attempt to learn the boundary that encloses most of the normal data points, enabling the model to later classify new data points as either "Normal" or "Anomalous". The following kernel and relevant hyperparameters:

- Kernel: the OCSVM uses a Radial Basis Function (RBF) kernel (kernel='rbf') to map the input features into a higher-dimensional space, which helps in separating normal data points from potential anomalies.
- Gamma: set to 'auto', it adjusts the influence of each data point in the model.
- Nu: the parameter nu=0.01 is crucial as it defines the expected proportion of anomalies in the dataset. A smaller nu value tells the model to expect fewer anomalies.

Finally, the trained two-step model was used to (i) evaluate its performance on the test dataset and to (ii) validate its capabilities with real traffic in the 6GDAWN testbed, integrating its predictions with the policy enforcement and the monitoring and visualization dashboards.

Federated Learning

To achieve better privacy preservation, improved scalability and reduced bandwidth usage in distributed environments (where multiple Public Safety Agencies and other organizations from different geographical areas and with disparate privacy requirements may be involved), an evolved architecture has also been devised as part of R UC2 PoC1.

This enhanced architecture relies on the use of Federated Learning (FL) to train local models at each Domain, aggregating these updates to build a global anomaly detection model that captures shared patterns across the different agencies. This common model could be built within the AE of the IDMO (Inter-Domain Manager and Orchestrator) as per the 6G-DAWN architecture defined in E3, which would then send the updated global model to the edge devices for fine-tuning, ensuring all local AE instances at each of the different Domain can detect network-wide anomalies while retaining the ability to spot localized ones.

The global model would consist of a unique service classifier and individual OCSVMs (one for each of the existent network services / applications across the whole set of Domains in the network), which would be trained with the domain-level models' update data. More specifically, the local models would send the following information to be aggregated at the inter-domain level for each of the OCSVM-based classifiers:

- Support Vector Points: the compressed set of data points defining the local decision boundary (between anomalous and normal traffic).
- Kernel information: Kernel function type (such as RBF or polynomial) and associated hyperparameters (e.g. Gamma=auto and nu=0.01 in the implemented model).
- Anomaly Scores: statistical summaries or metrics derived from local training to inform the server of data trends.

This information would then be aggregated using support vector merging, where the individual vectors would be processed to remove overlapping or redundant ones and adjust their weights based on their relevance. This relevance would be based on the data volume at each Domain and service with respect to the rest, on the frequency of appearance and on the confidence in local models (where more advanced models developed by certain agencies with higher accuracy could have higher weights). Compared with other aggregation techniques such as Federated Averaging (FedAvg), this approach ensures the global model retains the critical boundaries of service-specific models at each domain (without altering them) and would improve the interpretability of the global model, where its decision boundary can be easily interpreted and visualized in terms of the merged local support vectors (a critical feature for the development of a joint model where multiple agencies with different security/resiliency criteria and requirements must agree to a common decision boundary).

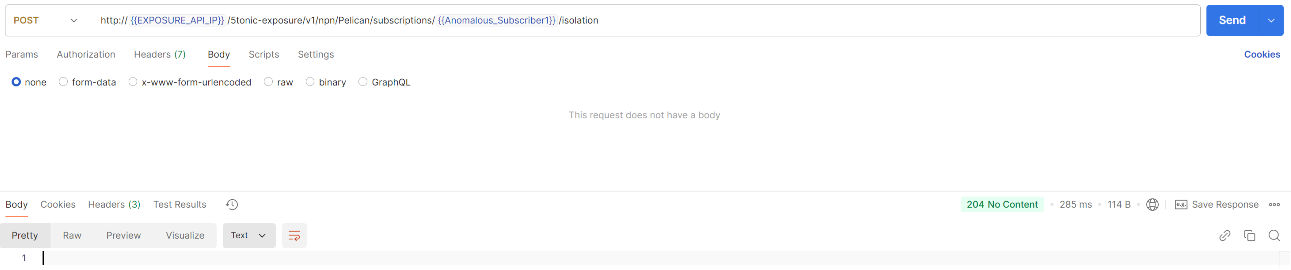
Regarding the location / abstraction level of the domain-specific models with regards to the 6G-DAWN architecture, these would be maintained by the AE at the Infrastructure layer (e.g. within the Core/Transport technology domain as an external AF or as a NWDAF model; or within the Cloud technology domain) and aggregated / routed towards the upper level of the Management and Orchestration Layer via the DMOs (Domain Managers and Orchestrators).

Finally, the use of Federated Learning to create a global anomaly-detection model is particularly relevant for the analyzed use case due to the unique requirements and constraints regarding data privacy and security of different public safety agencies (e.g. police, firefighters, EMS) due to their unique operational and legal contexts. The information logged and used to train the devised anomaly detection system (such as communication flows between agents and/or users and their location, which may be inferred from the collected data) may be protected under special regional and/or agency-specific regulations such as GDPR, LOPDGDD or LOPS in Spain. As already mentioned briefly, this enhanced architecture is also key to achieve scalability, where otherwise large amount of data (detailed and granular data logs of traffic and other infrastructure-related data) would need to be exchanged between DMOs/IDMOs over large distances for the creation of a global model. Additionally, the use of FL can improve the interpretability of the model and enable faster adaptation to changing network conditions, supported services and response procedures.

2.2.1.1.5 Policy enforcement

This component, implemented as service within the network Core and exposed to the AF via the NEF interface, allows external services to, among other things, isolate certain users behaving anomalously under their own defined parameters and criteria. As first outlined in deliverable E3, due to the fact that the 3GPP NEF interface (3GPP TS 23.503[12]) does not support an interface for making policy changes to the PCF (allowing external applications and services to perform these changes), a custom extension has been defined. The implemented NEF extension therefore allows external applications to isolate certain users (i.e. UEs) in the network via the following parameters:

- **npnID**: ID of the NPN to which the given UE is connected.
- **subscriptionID**: the ID associated to the given UE, that is, the IMSI assigned to it.


POST http:// {{EXPOSURE_API_IP}} /5tonic-exposure/v1/npn/Pelican/subscriptions/ {{Anomalous_Subscriber1}} /isolation

Body: ● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

204 No Content • 285 ms • 114 B

FIGURE 30. ISOLATION API

Upon the reception of the request initiated by the AF, the Core will re-map the specified UE (based on the npnID and subscriptionID specified by the AF) to the Quarantine slice (S-NSSAI #129-1 in the testbed) as shown in Figure 30 (note that the depicted elements, such as the gNodeB, the UPF or the AMF, correspond to the same elements before the isolation – only the assigned slice changes).

[12] https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3334

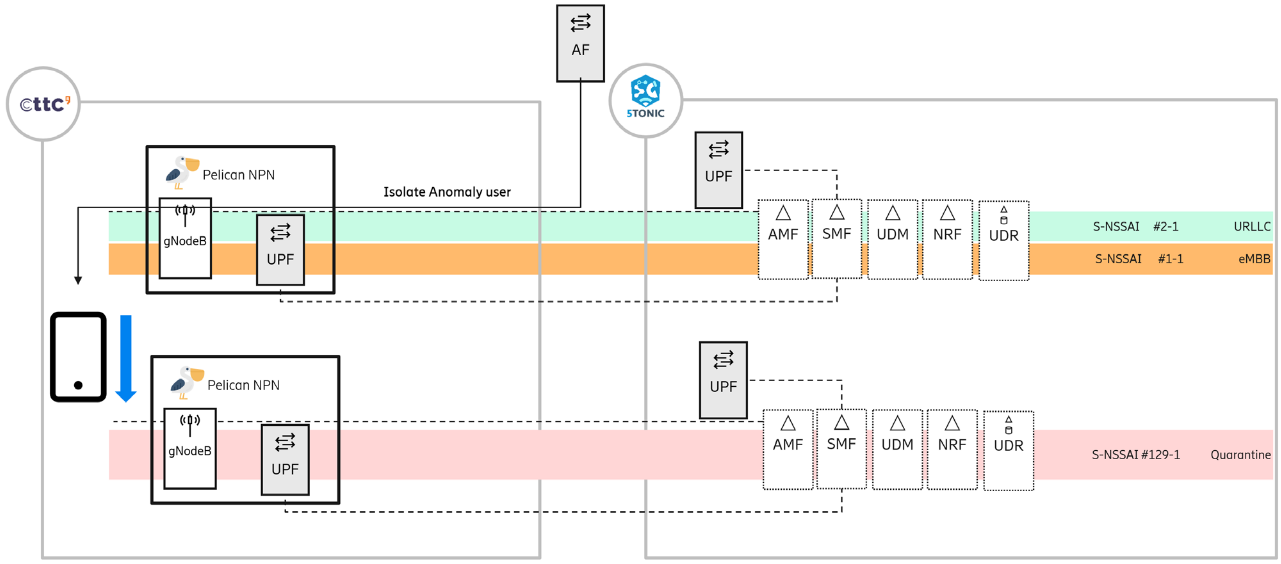


FIGURE 31. ISOLATION IN ACTION

This alternative was chosen to isolate anomalous users over other possible approaches, such as the use of URSP rules because it enables (i) increased, centralized control, (ii) the enforcement of the redirection without requiring cooperation from the UEs, (iii) a higher scalability, being able to isolate multiple UEs at once without having to update and interact with all the individual UEs and (iv) it enables real-time enforcement, improving the response time to anomalous behavior and minimizing its impact as much as possible.

The alternative approach, consisting in the installation of URSP rules at the anomalous UE, that was initially consider as a possible way of isolating the UEs, is shown in Figure 32. This would be achieved

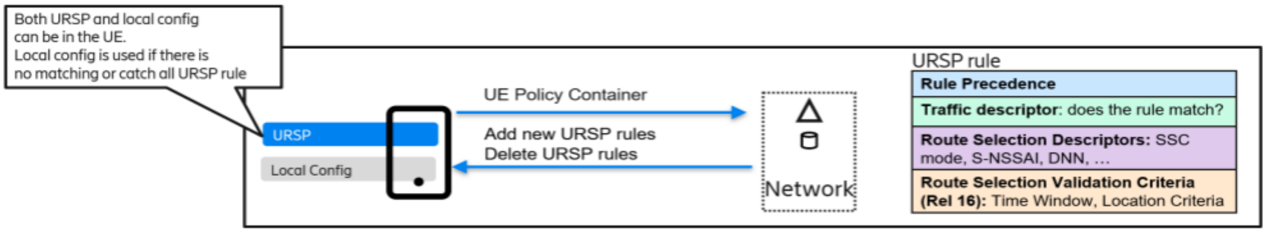

Both URSP and local config can be in the UE.
Local config is used if there is no matching or catch all URSP rule
URSP
Local Config
UE Policy Container
Add new URSP rules
Delete URSP rules
Network
URSP rule
Rule Precedence
Traffic descriptor: does the rule match?
Route Selection Descriptors: SSC mode, S-NSSAI, DNN, …
Route Selection Validation Criteria (Rel 16): Time Window, Location Criteria

FIGURE 32. URSP CONFIGURATION AT THE UE

2.2.1.2 R UC2 PoC1 KPIs Evaluation and Results

The set of KPIs originally defined as part of E3 are summarized in Table 10

TABLE 10. RESILIENT UC2 KPIS

KPI	Unit	Type	Definition

NPN Resiliency Improvement – Energy Optimization	%	Resiliency	By means of isolating high-energy consuming (anomalous) users into the quarantine slice, overall energy consumption of the main service slice will be decreased. Thus, overall NPN resiliency will be increased.
NPN Resiliency Improvement – Resource Availability	%	Resiliency	By means of isolating high-energy consuming (anomalous) users into the quarantine slice, the network resources previously used/wasted by anomalous users will be available to be used by remaining existing users of the NPN. Thus, overall NPN resiliency will be increased.
True Positive Rate (*)	%	Anomaly Detection Effectiveness	This KPI is used to assess how well the anomalous user detection algorithm on NDT/AF is operating.
Precision (*)	%	Anomaly Detection Effectiveness	This KPI is used to assess how well the anomalous user detection algorithm on NDT/AF is operating.

2.2.1.2.1 Application-aware anomaly detection system

In terms of the application-aware anomaly detection system, the evaluation of how the designed system contributes towards the detection and its accuracy/precision is challenging due to the limited access to labeled or raw data for reference as already mentioned in section 2.2.1.2.1. However, based on the synthetic data generated by the data generator as per the traffic model previously defined (based on existing reports about traffic patterns in PPDR scenarios), it is possible to validate the model, assessing its viability and approximating its expected performance. Based on this generated data, the model’s performance in relation to the previously defined KPIs is studying bellow for both the initial classification model and the 2-step one.

Basic OCSVM implementation

The results from the first iteration of the anomaly detection system, based on a single OCSVM global model for all traffic are summarized in the form of a confusion matrix as shown in Table 11 As it can be seen, the observed performance is relatively good, particularly in its ability to accurately detect anomalies (TPR=1, accuracy=0.97). The model shows therefore perfect sensitivity with a high accuracy, together with a high precision and F1 score (0.92 and 0.96 respectively) using the synthetic dataset.

TABLE 11. CONFUSION MATRIX FOR BASIC OCSVM IMPLEMENTATION

OCSVM_Prediction	Anomalous	Normal
Label: Anomalous	6,579	0
Label: Normal	543	12,204

2-step classification model

For the 2-step model, a traffic classifier was added as a first processing step via the use of a random forest. As it can be seen in Table 12, it was highly successful in classifying the traffic flows, achieving almost perfect performance.

TABLE 12. PERFORMANCE OF TRAFFIC CLASSIFICATION

Task	Precision	Recall	F1-Score	Support
Command unit DL video	1	1	1	1,998
Command unit UL video	1	1	1	675
Data file transfer CAD/GIS	1	1	1	5,999
EMS Internet access	1	0.99	1	322
EMS data transfer	1	1	1	305
EMS patient tracking	0.99	0.99	0.99	333
Mobile video camera	1	1	1	1,635
Secure file transfer	1	1	1	1,020
VoIP	1	1	1	7,039

Finally, the overall performance of the system (aggregating all predictions among the different traffic classes) is shown in Table 13 in the form of a confusion matrix. As it can be seen, this second iteration of the model has proven to slightly improve its performance, mainly with regards to its precision, with TPR=1, accuracy=0.97, precision=0.93 and F1=0.96. Beyond the slight improvement of the performance observed with the given tested, the main advantage of the second approach is the greater visibility it enables, being able to associate anomalous with given services or with certain devices; and to better fine-tune the model in real-life scenarios. As an example, Figure 33 shows the TPR and precision rates for each of the individual models per traffic type, exhibiting how, for example, the models for the video streams are the least effective.

TABLE 13. CONFUSION MATRIX FOR 2-STEP MODEL

OCSVM_Prediction	Anomalous	Normal
Label: Anomalous	6,664	0
Label: Normal	488	12,162

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

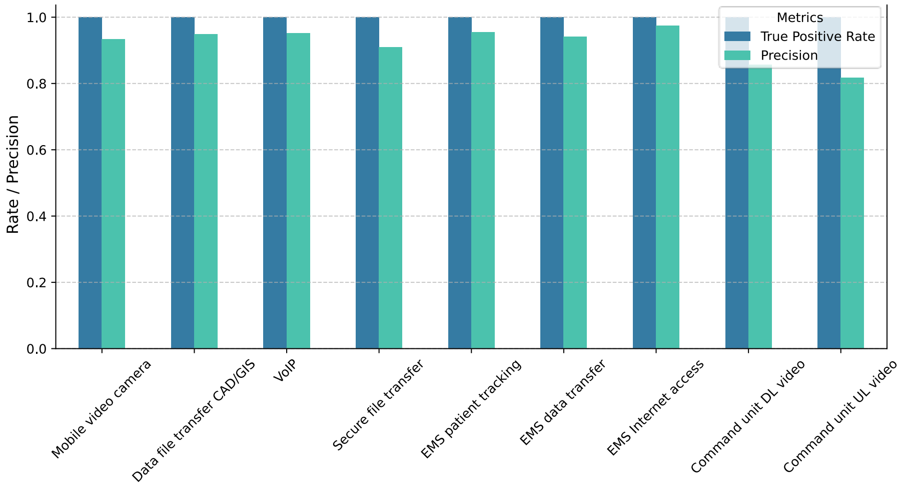


FIGURE 33. MODEL PERFORMANCE BY TRAFFIC TYPE

2.2.1.2.2 Energy consumption and resilience global impact

With regards to the first two KPIs defined in Table 10, it is challenging to objectively measure the improvement in energy consumption resulting from the design and implementation of the anomaly detection and isolation system for B5G networks, primarily because there are no available benchmark datasets to compare against and because the simulation of complex scenarios (such as the ones described in E3) resulting in unexpectedly high energy consumptions are hard to simulate. Without a baseline dataset of normal energy consumption patterns, it becomes difficult to quantify the precise energy savings or efficiency gains that the system achieves.

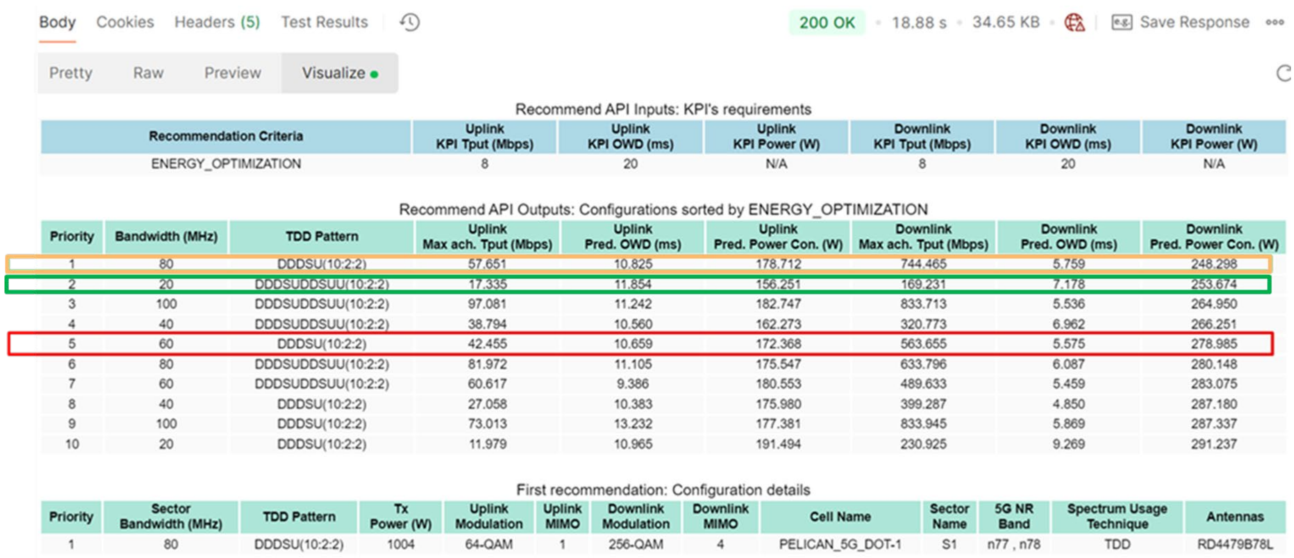
However, the results from the implemented machine learning models (both for the network and application-aware detections systems) demonstrate its effectiveness in detecting anomalies, specifically identifying UEs that contribute to unexpectedly high energy consumption. Through the isolation of these anomalous UEs into a separate slice, the impact on overall network energy consumption can be significantly reduced. Although direct measurement comparisons are limited, the model clearly shows its capability to address energy inefficiencies by dynamically managing these outliers, highlighting the potential for operational savings and improved network sustainability in real-world B5G environments.

It also must be highlighted that the implemented system not only achieves energy efficiencies via the isolation of anomalous UEs, but also via the optimization of the network configurations applied to the NPN, allowing to enforce the configuration that would produce the lowest energy consumption while still offering expected performance as seen in E UC1 PoC2. The use of an anomaly detection system and enforcer is critical in this type of scenarios in order to fulfill the energy savings promised by the NPN Recommender, as it helps to ensure that the network behavior stays close to the traffic models with which the NPN was able to produce its energy consumption and performance predictions.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya

As an example, for the specific traffic model defined in section E UC1 PoC2, the ML algorithms implemented as part of the 6G-DAWN project is capable of producing 4 additional configurations with lower power consumption than the initial configuration of Bandwidth - 60MHz and TDD Pattern - DDDSU (10:2:2) as seen in Figure 34. In this scenario, the optimal energy-efficient configuration with minimal resource usage that meets the required KPIs (Throughput, One Way Delay, and Energy Consumption) is Bandwidth - 20MHz and TDD Pattern - DDDSUDDSUU (10:2:2). More concretely, using either of the two top recommended configurations for energy efficiency, the system has the potential to achieve the following energy savings:

- 60Mhz Uplink-172W·h vs 20MHz Uplink-156W·h would achieve a 9% energy saving.
- 60Mhz Downlink-278W·h vs 20MHz Downlink-253W·h would achieve a 9% energy saving.



Recommend API Inputs: KPI's requirements

Recommendation Criteria	Uplink KPI Tput (Mbps)	Uplink KPI OWD (ms)	Uplink KPI Power (W)	Downlink KPI Tput (Mbps)	Downlink KPI OWD (ms)	Downlink KPI Power (W)
ENERGY_OPTIMIZATION	8	20	N/A	8	20	N/A

Recommend API Outputs: Configurations sorted by ENERGY_OPTIMIZATION

Priority	Bandwidth (MHz)	TDD Pattern	Uplink Max ach. Tput (Mbps)	Uplink Pred. OWD (ms)	Uplink Pred. Power Con. (W)	Downlink Max ach. Tput (Mbps)	Downlink Pred. OWD (ms)	Downlink Pred. Power Con. (W)
1	80	DDDSU(10:2:2)	57.651	10.825	178.712	744.465	5.759	248.298
2	20	DDDSUDDSUU(10:2:2)	17.335	11.854	156.251	169.231	7.178	253.674
3	100	DDDSUDDSUU(10:2:2)	97.081	11.242	182.747	833.713	5.536	264.950
4	40	DDDSUDDSUU(10:2:2)	38.794	10.560	162.273	320.773	6.962	266.251
5	60	DDDSU(10:2:2)	42.455	10.659	172.368	563.655	5.575	278.985
6	80	DDDSUDDSUU(10:2:2)	81.972	11.105	175.547	633.796	6.087	280.148
7	60	DDDSUDDSUU(10:2:2)	60.617	9.386	180.553	489.633	5.459	283.075
8	40	DDDSU(10:2:2)	27.058	10.383	175.980	399.287	4.850	287.180
9	100	DDDSU(10:2:2)	73.013	13.232	177.381	833.945	5.869	287.337
10	20	DDDSU(10:2:2)	11.979	10.965	191.494	230.925	9.269	291.237

First recommendation: Configuration details

Priority	Sector Bandwidth (MHz)	TDD Pattern	Tx Power (W)	Uplink Modulation	Uplink MIMO	Downlink Modulation	Downlink MIMO	Cell Name	Sector Name	5G NR Band	Spectrum Usage Technique	Antennas
1	80	DDDSU(10:2:2)	1004	64-QAM	1	256-QAM	4	PELICAN_5G_DOT-1	S1	n77 , n78	TDD	RD4479B78L

FIGURE 34. RECOMMENDATION RESPONSE

However, these savings can only be achieved, while still complying to the expected UL Throughput, DL Throughput, One Way Delay KPIs, if all possible disruptions and anomalous behavior is addressed by the network. In a nutshell, effectively detecting and isolating anomalous users and network components is key in achieving the potential energy savings attainable via the optimization of network configurations, in reducing additional energy overuse and in reducing the impact of anomalous components to other users, therefore improving the resiliency of the network.

2.2.1.3 Achievements and lessons learned

As already stated, the current PoC demonstrates a scenario where an anomalous device is quarantined to (i) reduce the negative impact it is able to inflict to other devices in terms of QoS, to (ii) minimize the negative impact of that the device is causing by diagnosing and correcting its behavior and to (iii) improve the overall resilience of the network.

In terms of the capability to detect energy-related anomalies via the creation of a Network Digital Twin based on 3 domain knowledge areas (theoretical, empirical and trained) – i.e. the Network anomaly detection system – the primary achievement of this research project has been establishing a robust correlation between the network configuration and traffic patterns of cellular networks with their expected energy consumption. These findings highlight how different configurations and usage scenarios impact energy efficiency, providing valuable insights for optimizing network operations. A key takeaway is the absence of a reliable instantaneous energy index, as energy consumption measurements at granular time scales exhibit high variance and are prone to significant errors. Instead, the research underscores the necessity of aggregating data over a minimum duration of one hour (with aggregated energy consumption values in the magnitude of 1 Wh) to mitigate variance and produce meaningful, reliable energy consumption metrics.

In terms of the capability of creating service specific anomaly detection mechanisms (via the Application-aware anomaly detection system), this PoC has designed a two-step unsupervised learning algorithm capable of pinpointing possible offending UEs in a B5G network when anomalies in energy consumption or other KPIs are detected by the NDT. Notably, the implemented model has been trained and tested to detect traffic-based anomalies in Public Protection and Disaster Relief (PPDR) scenarios, demonstrating its robustness and adaptability. This capability is achieved using untrained data, highlighting the model's effectiveness in identifying irregularities without prior knowledge of the traffic patterns, making it a valuable tool for dynamic and high-stakes network environments.

In terms of its contributions to the future architecture of 6G networks, this PoC has contributed with the definition of custom extension to the NEF interface, that allow network intelligence to expose detailed energy consumption metrics and anomaly patterns to external applications, enabling seamless integration of energy efficiency into service-level criteria. This aligns with the concepts outlined in 3GPP TR 22.882, which explores the inclusion of energy efficiency as a key service requirement in future networks. The demonstrated ability to detect and address energy anomalies not only advances energy-aware service design but also supports the optimization of network resources, paving the way for a sustainable, intelligent 6G ecosystem that prioritizes energy efficiency as a foundational service characteristic.

Furthermore, the implemented scenario, which shares some similarities with the use case 5.6 outlined in 3GPP TR 22.882, introduces multiple key differences and enhancements with respect to the latter one:

- While in the use case proposed by the 3GPP TSG SA [3GPP TSGSA[13]], it is the company operating the NDT the one responsible for finding that a certain component within the NPN

[13] 3GPP TSG SA https://www.3gpp.org/3gpp-groups

(in their case, the UPF) is consuming an abnormal amount of energy (using the information provided by the operator), in the PoC proposed in this project, this responsibility is delegated to the 6G DAWN framework. More concretely, the AE, using the information collected by the MS (consisting of multiple software probes and sensors) can detect the anomalous event and notify the interested parties by sending a notification event to the AF. This modification greatly reduces the effort required by NDT operators to effectively respond to these types of events, relieving external AFs from the burden of detecting anomalous energy consumption events only requiring them to handle these via the implementation of their own business-specific workflows.

- While TR 22.882 [3GPP TR282] [14]does not explicitly mention how the energy consumption measurement information, together with other network usage-related analytics used to diagnose the event, are exposed to the NPN operator (they identify this challenge as a potential new requirement), this project is proposing to employ an augmented version of the 3GPP NWDAF component to do this.
- While in the use case in TR 22.882 [3GPP TR282] [15] the only way of resolving the negative impact of the device towards the network is to apply an application-specific fix (which is out of the scope of the network), this project introduces an interface that allows the enforcement of preventive network-specific actions with the intention to reduce its negative impact on the network. More specifically, this is supported by the 6G DAWN DE component, which allows the controlled redirection of the UE to a different slice via the network Core.

These key differences introduce numerous advantages with respect to the initial use case 5.6 in 3GPP TR 22.882, simplifying the work required by NPN operators to handle these events, achieving better interoperability with 3rd party systems, and improving the resiliency of the network by automatically actuating upon the detection of anomalous behavior.

2.2.1.3.1 Other anomalous events

In addition to the scenario implemented as part of the project, where the isolation was determined in terms of expected traffic flows by the application-aware detection system, there are multiple other energy-related anomalous scenarios that would benefit from the proposed 6G DAWN architecture and from the extension of the current 3GPP specifications proposed in this project. Some of the possible behaviors or scenarios with the potential to have a detrimental effect on the energy efficiency of the network are:

[14] 3GPP TR 22.882 "Study on Energy Efficiency as service criteria": https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4096

[15] 3GPP TR 22.882 "Study on Energy Efficiency as service criteria": https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4096

Financiado por la Unión Europea NextGenerationEU · Plan de Recuperación, Transformación y Resiliencia · UNICO I+D · Gobierno de España · cttc Centre Tecnològic de Telecomunicacions de Catalunya

- **Intermittent connections:** the continuous establishment and termination of device connection can strain the network and provoke high energy consumption. One possible response from the network could be to release any existing PDU sessions from the device and block future connection requests with a SM back-off timer, similarly as how the network could react to a TOO_FREQUENT_SERVICE_ACCESS event (Table 6.7.5.3-3 TS 23.288 [3GPP TS238]).

- **Defective MCS negotiation:** the incorrect selection of the session Modulation and Coding Scheme (either maliciously or due to a faulty device), selecting modes which incur in high energy consumption, may also have a negative impact on the network (both in terms of energy efficiency and QoE for other users). The appropriate detection and handling of these anomalies would improve the resilience of 6G networks in these events.

- **Inefficient traffic patterns:** especially in NPN networks where traffic patterns and network configurations can be adjusted to achieve the best possible performance, unplanned traffic behavior may critically affect energy efficiency. One example would be devices unexpectedly sending immediately bursty, non-critical traffic, breaking the break sleep mode of the different radio components, instead of batching all traffic in periodic, long-lived transmissions.

- **Transmissions at cell borders:** communication with devices at cell borders are specially energy intensive due to a greater signal attenuation (on top of other signal impairments, requiring higher energy per transmitted bit), to interferences with neighboring cells and to higher overheads (especially when frequently changing between base stations within the neighboring area).

While some of these scenarios could be directly detected via the monitoring of unexpected events solely from a network traffic or mobility perspective (instead of how these affect the energy consumption of the network), some more complex scenarios involving a combination of multiple, smaller deviations which jointly contribute to a significant rise in energy consumption could only be detected via the proposed measures. Moreover, in cases where energy consumption has a direct impact on the availability of the network, such as those where power outages occur, may have a direct impact on the resilience of the network and its ability to continue operation in critical circumstances.

Financiado por la Unión Europea NextGenerationEU Plan de Recuperación, Transformación y Resiliencia UNICO I+D GOBIERNO DE ESPAÑA cttc Centre Tecnològic de Telecomunicacions de Catalunya

3 Conclusions

6G DAWN delivers a Decentralized AI closed loop (MS-AE-DE) that is implemented, validated with KPIs, by exploiting inter-building block interfaces across all RESILIENT PoCs, those defined in previous deliverable E3.

The implemented decentralized 6GDAWN AI (MS-AE-DE) achievements in RESILIENT PoCs are as follows:

- Decentralized intrusion detection and mitigation in kubernetes cluster with E2E 5G connectivity and video-streaming service with accuracy of almost 97%. The PoC achievements include: Network-anomaly-detection-based assurance of the healthy status of a Kubernetes cluster, Full-scale and visibility of network traffic in a Kubernetes cluster, Enriched and consistent flow telemetry over highly volatile and dynamic IP addresses, and multi-model parallel processing for anomaly detection to scale to the volume and velocity of flow telemetry data produced but the different sensors.
- Improving the trustworthiness of FL model employed in 6GDAWN by Blockchain implementation for integrating FL in O-RAN environments. The implemented strategy utilizes Polygon's Layer 2 solutions to develop a DApp designed for managing and validating machine learning model training and data exchanges across a multi-vendor landscape to support trusted collaborative learning.
- Improving the reliability of 6G networks by detecting and isolating users who have high energy consumption. The PoC demonstrates a scenario where an anomalous device is quarantined to (i) reduce the negative impact it inflicts on other devices in terms of QoS, to (ii) minimize the negative impact of that the device is causing by diagnosing and correcting its behavior, and to (iii) improve the overall resilience of the network.

Overall, E5 illustrates the successful implementation and validation of decentralized AI (MS-AE-DE) in all RESLIENT PoCs of 6GDAWN through their defined KPIs in E3.

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D
GOBIERNO DE ESPAÑA
cttc Centre Tecnològic de Telecomunicacions de Catalunya