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

SUCCESS-6G: DEVISE

WP4 Deliverable E11

Final testing and validation of service KPIs

Project Title:	SUCCESS-6G: DEVISE
Title of Deliverable:	Final testing and validation of service KPIs
Status-Version:	v1.0
Delivery Date:	30/04/2025
Contributors:	Francisco Paredes, Miguel Fornell (IDNEO), Javier Santaella (CELLNEX), Carmen Vicente (CELLNEX), Pavol Mulinka, Roshan Sedar, Charalampos Kalalas, Miquel Payaro (CTTC), Guillermo Candela Belmonte (Optare), Michail Dalgitsis, Eftychia Datsika, Maria Serrano, Angelos Antonopoulos (NBC)
Lead editor:	Francisco Paredes (IDNEO), Charalampos Kalalas (CTTC)
Reviewers:	Charalampos Kalalas, Miquel Payaro (CTTC)
Keywords:	Testing; validation; service architecture; KPI values

Document revision history

Version	Date	Description of change
v0.1	19/03/2025	ToC created
v0.2	30/03/2025	Content added in Sections 2 and 3
v0.3	15/04/2025	Content added in Sections 4 and 5
v0.4	28/04/2025	Minor editing modifications
v1.0	30/04/2025	Final version uploaded to the website

Disclaimer

Acknowledgment

The research conducted by SUCCESS-6G - TSI-063000-2021-39/40/41 receives funding from the Ministerio de Asuntos Económicos y Transformación Digital and the European Union-NextGenerationEU under the framework of the “Plan de Recuperación, Transformación y Resiliencia” and the “Mecanismo de Recuperación y Resiliencia”.

[1] http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en_US

Executive Summary

This deliverable presents a comprehensive report on the final testing activities and methodology employed for validating the Key Performance Indicators (KPIs) associated with the user story “Vehicular condition monitoring with security guarantees”, conducted within the framework of the SUCCESS-6G-DEVISE project. The primary objective was to assess and confirm the functional readiness of the final testing architecture and ensure that the various service enablers are properly integrated and performing in accordance with the defined KPIs. The document describes the testing methodology followed at the two SUCCESS-6G testbeds (CELLNEX Mobility Lab and SUPERCOM platform), including details of the testbed configurations, tools, and metrics employed across different scenarios. In particular, emphasis is given on the establishment of the connectivity of the vehicular on-board unit with the edge computing infrastructure, the development of the data analytics components, and the implementation of service orchestration during vehicle handovers. Multiple tests were executed to validate the end-to-end system behavior, capturing performance metrics under varying mobility patterns, network loads, and environmental conditions. The final list of the achieved KPI values is also reported. These results serve to demonstrate the operational maturity of the SUCCESS-6G-DEVISE solutions and their suitability for deployment in next-generation vehicular networks powered by 6G technologies.

Table of Contents

List of Figures

List of Tables

1 Introduction

The integration of intelligent condition monitoring in vehicular systems is expected to play a pivotal role in the development of autonomous and connected mobility, ensuring reliability, sustainability, and efficiency in future transportation ecosystems. The staggering volume of vehicular measurement streams, driven by the widespread deployment of onboard sensors, in conjunction with expanded computational resources, offers enhanced monitoring capabilities and unlocks unprecedented application scenarios. As such, real-time condition monitoring of onboard vehicular equipment, fault provisioning, and predictive maintenance represent key services within the evolving vehicle-to-everything (V2X) paradigm. Building on massive data availability, such services leverage the proliferation of advanced artificial intelligence (AI) models and knowledge-extraction techniques to reveal hidden patterns, detect unknown correlations, and generate actionable intelligence with minimal human intervention. In turn, the timely detection of irregular patterns in monitoring information and the identification of anomalies/failures enhance operational efficiency, improve vehicular safety, and reduce unplanned downtime and maintenance costs.

The "Final Testing and Validation of Service KPIs" deliverable outlines the final phase of testing and validation for the Key Performance Indicators (KPIs) of the vehicular condition monitoring service in the SUCCESS-6G-DEVISE project. It represents a critical milestone in ensuring that the service not only meets its functional requirements but also achieves the predefined KPIs essential for assessing its reliability, efficiency, and overall effectiveness in a real-world deployment context.

This document thoroughly documents the test activities carried out, offering a clear and structured account of the procedures, conditions, and environments under which the service was evaluated. It highlights the testing infrastructure used, as well as the tools employed to facilitate on-board unit connectivity, data collection, analytics, and KPI computation. A significant portion of the report (Section 3) is dedicated to explaining the methodology adopted for KPI measurement. This includes the process of test scenario design and the technical approaches taken to ensure accurate and reproducible results. The methodology was carefully crafted to align with real-world vehicular scenarios and to reflect realistic use cases involving vehicular condition monitoring.

By presenting the results obtained through these comprehensive tests, the deliverable serves to validate the readiness of the service for further development, scaling, or deployment. It also provides assurance to stakeholders that the service performance is in line with the expected benchmarks and technical objectives defined at earlier stages of the SUCCESS-6G-DEVISE project.

2 Final testing architecture and services

2.1 Network Architecture

The CELLNEX Mobility Lab, located at the Circuit Parcmotor Castellolí near Barcelona (Spain), is a pioneering and innovative test space for the development of Intelligent Transportation Systems (ITS) technological solutions associated with 5G, sustainable mobility, and autonomous vehicles. The circuit has been equipped with C-V2X, 5G, and edge computing technologies and a private wireless network with coverage throughout the venue. The CELLNEX Mobility Lab is based on green infrastructures that operate under the premise of efficient energy management and environmental sustainability, since most of them are self-sustainable sites that use eolic and/or solar energy. Besides, future road deployments will follow the same premises, as most of the roads are located in isolated areas with few possibilities to connect to the grid power.

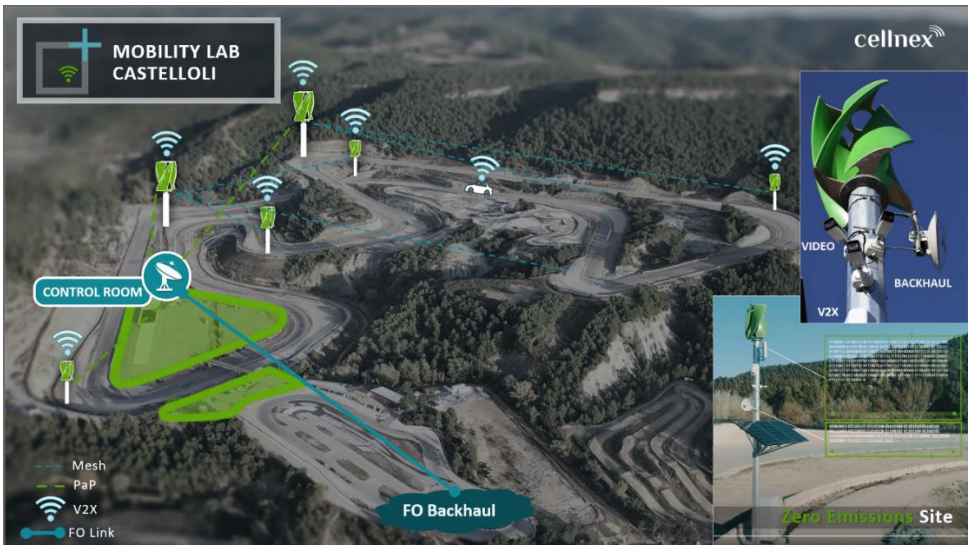


Figure 1: CELLNEX Mobility Lab (Castelloli)

Regarding communication infrastructures, the Cellnex Mobility Lab in Castelloli has a 5G private network which has been updated to provide Mobile Edge Computing (MEC) technology, including two additional distributed User Plane Functions (UPFs) besides the 5G Core and the central UPF system. This 5G network, based on the Druid solution, is complemented with a 5G RAN composed of two 5G New Radios from the Sunwave solution.

As reported in Deliverable E5 [4], the computational infrastructure available in the Cellnex Mobility Lab is based on LENOVO servers SR650 and SE350 to provide the required clusters of virtual resources for the “Local Cloud” and the remote “edge nodes”. Thus, there are three clusters of virtual computing resources. Currently, all of them are deployed in the Control Room location to ensure the reliability and availability of the computational edge resources for the experimental activities in the SUCCESS-6G project.

- **Local Cloud**: composed of two virtualized LENOVO SR650 servers. It provides a set of virtual machines, one VM dedicated to 5G-CORE SW services and the others for Central Apps.
- **Edge Node 1** and **Edge Node 2:** each one composed of one virtualized LENOVO server SE350. Whereas a VM of each server hosts the associated distributed UPF, the other VMs are dedicated to the critical services that must be deployed and run at each edge node.

These edge servers have been moved from the remote edge nodes called “node 1” and “node 8”, to the Control Room location, to overcome some issues with the power supply available in the nodes. Regarding 5G RAN infrastructure, the architecture for this project is composed of two types of nodes, as shown in Figure 2, depending on the power supply features:

- **Node 1**: It is a Grid Site, connected to the grid power supply.
- **Node 8**: It is a Green Site that gets its power supply from wind and solar energy.

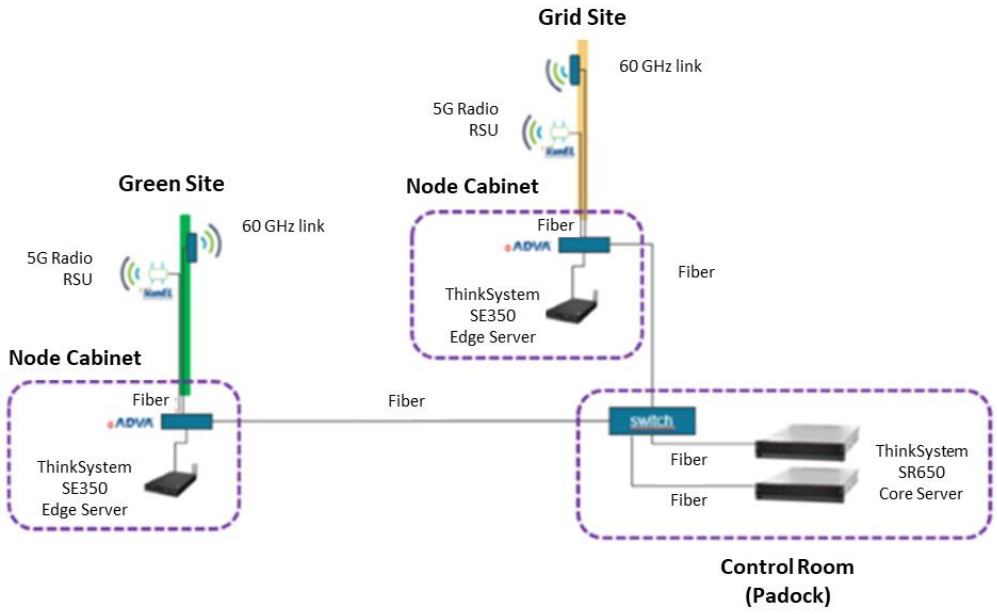


Figure 2: Cellnex Mobility Lab – Original ICT Architecture

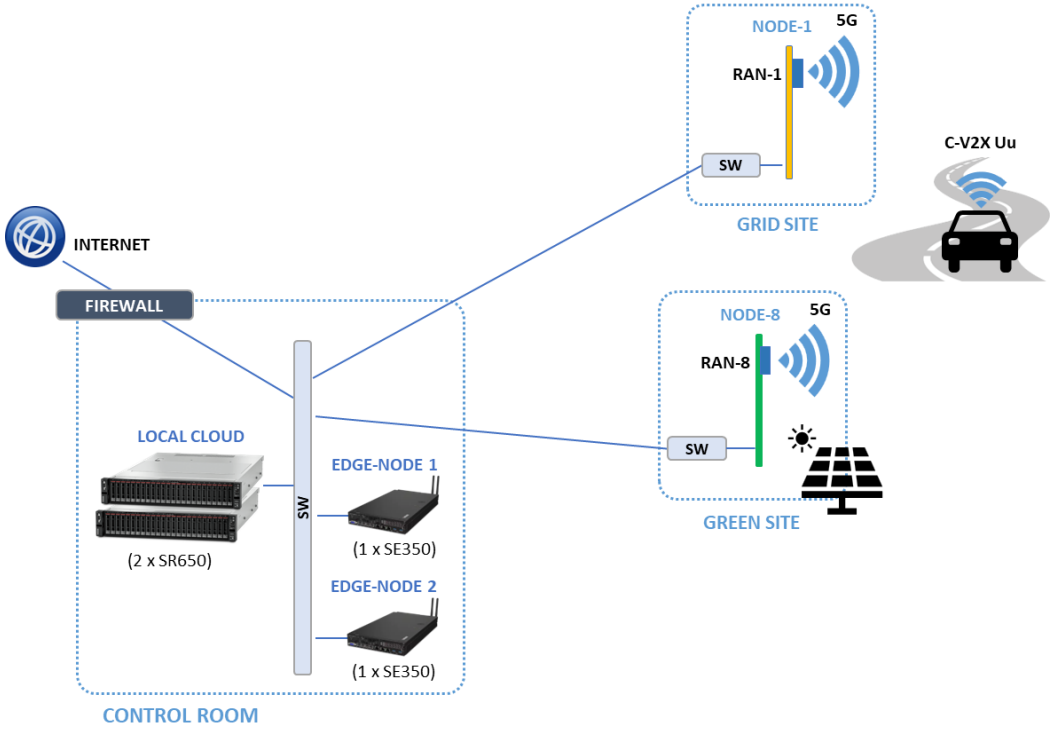


Figure 3: Cellnex Mobility Lab – Final ICT Architecture

The ICT infrastructure, displayed in Figure 3, could be accessed either locally from the Cellnex Control Room in the Mobility Lab or remotely by using a VPN connection. Network infrastructure offered to the SUCCESS-6G project supports a complete non-roaming 5G Private Network in the Castelloli Mobility Lab.

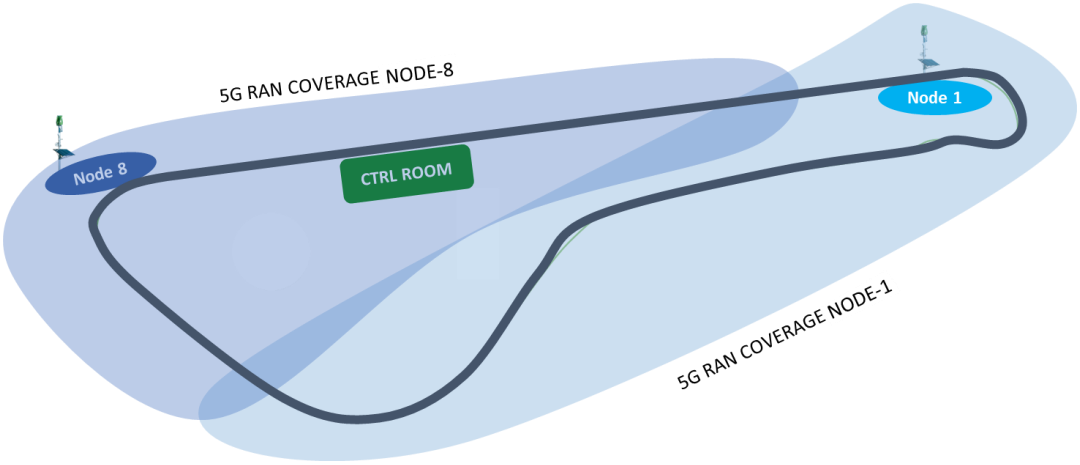


Figure 4: Location of gNBs and radio sectors in Castellolí test site

As explained before, it is composed of 5G-SA Core, MEC units and 5G New Radios for C-V2X and other novel ITS services in the mobility scope. The main elements of the SUCCESS-6G 5G Private Network are:

- 1 5G Core SA, service-based-architecture
- 1 Central UPF
- 2 Distributed UPF
- 2 MEC
- 2 5G-NR (BBU + RRU) 100 MHz in band n77. (3800 – 3900 MHz)

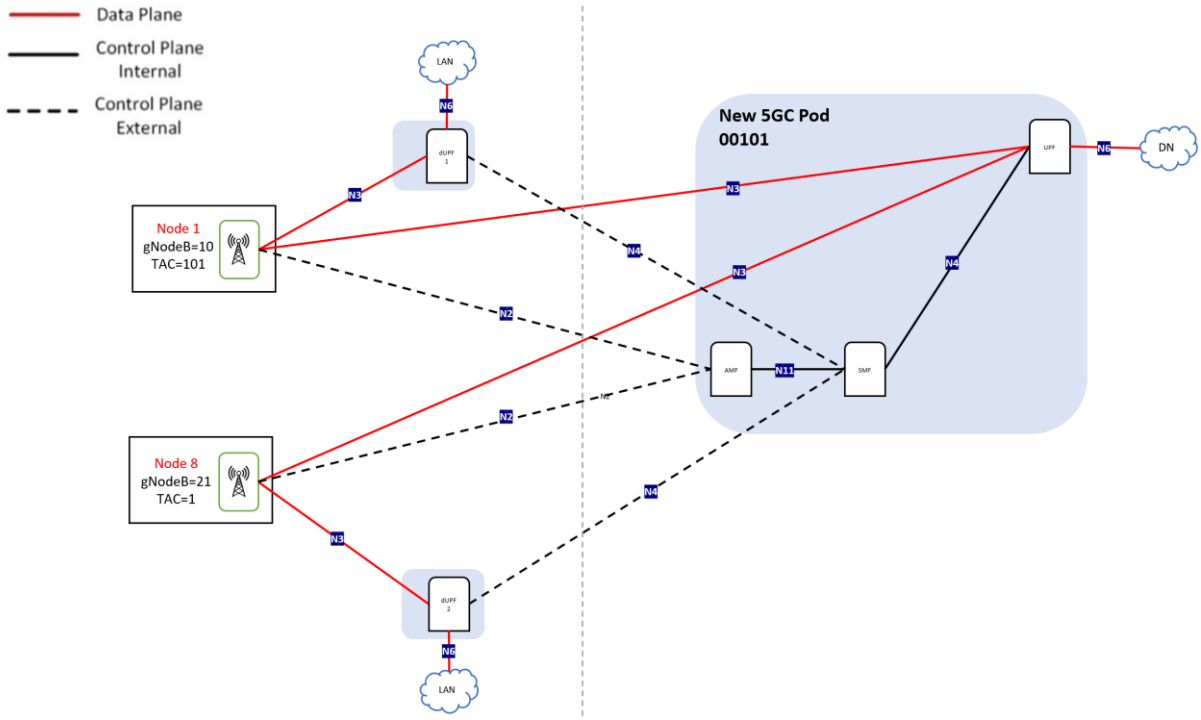


Figure 5: Internal architecture of the 5G SA networks deployed in the small-scale test site of Castellolí

The 5G Core and UPFs dedicated to the SUCCESS-6G project have been supplied by Druid, an Irish company specialized in 4G/5G private network solutions. The deployed solution consists of a 5G Stand-Alone (SA) configuration with centralized and distributed UPFs. There are two distributed UPFs, one associated with Node 1 and the other with Node 8.

There are three SW instances dedicated to the private 5G network for the SUCCESS-6G project: one for the 5G Core and the other two for the distributed UPFs. While the 5G Core instance is deployed and running in the local cloud, the distributed UPF instances are running on the edge nodes, one dUPF instance at each edge node.

Druid provides a technology platform called Raemis, which is a set of cellular software assets optimized for business use cases. This platform offers mobile communication solutions that are tailored to meet the needs of use cases that require technologies related to wireless cellular networks.

Figure 6: Raemis dashboard

The outdoor coverage of the 5G network is supported on SUNWAVE technology, using the n77 band. It is a distributed solution composed of BBU and RRU units.

- **BBU** nCELL-T5000, is the 5G-NR base station unit. It provides central control and management of the entire base station system, providing network access functions, direct access, and data interaction with 5G Core and baseband processing functions. Facilitates nGAP, XnAP interface, 5G NR access network protocol stack functions, RRC, PDCP, SDAP, RLC, MAC, and PHY protocol layer functions.
- **RRU** RU4370 converts fiber optic signals into cellular air technologies, providing outdoor coverage of the SUCCESS-6G 5G network.

Figure 7: 5G-NR RRU module

While Table 1 details the composition of each RAN site and main settings identifiers, Table 2 details the characteristics of the gNBs at the Castellolí small-scale test site:

Site name	Cell type	MIMO type	Equipment Provider	Energy supply	gNB ID	TAC	PCI	PLMN ID
Node 8	small cell	4x4	Sunwave	Self-sustainable	21	1	18	00101
Node 1	small cell	4x4	Sunwave	Grid connected	10	101	17	00101

Table 1: Features of the gNodeBs at the Castellolí small-scale test site

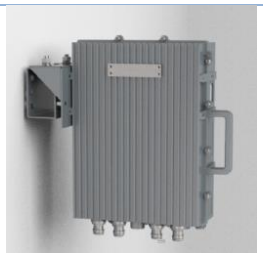
Site name	Product type	Product name	Product details	Illustration
Node 1 and Node 8 (Sunwave gNodeBs)	**Sunwave Antenna**	**JZD – 65DPG1515-3842T0**	• 4-port antenna • 15 dBi Gain • Bandwidth: 3800-4200 MHz	
	Sunwave RRU	**sCELL-3470RRU**	• Product Name: sCELL-3470RRU • 4T4R • Max total carrier BW is 100MHz for NR • 5W (37 dBm) Output Power • Bandwidth: 100 MHz in band n77. (3800 – 3900 MHz) • Weight 12Kg • 48 VDC	
	Sunwave Baseband Unit	**-**	• 2x RJ-45 100/1000 BASE-T Ethernet Port • 4 x 10Gbps ports SFP+ Ethernet Port • 4 x Radio Interface ports	

Table 2: Composition of gNBs at the Castellolí small-scale test site

In order to deploy the 5G network services and application services, all the LENOVO servers dedicated to the SUCCESS-6G project (2 SR650 servers and 2 SE350 edge-servers) have installed vSphere (vmware), which is a software that allows for virtualizing the computational resources and creating different virtual machines.

Figure 8: Vmware vSphere platform

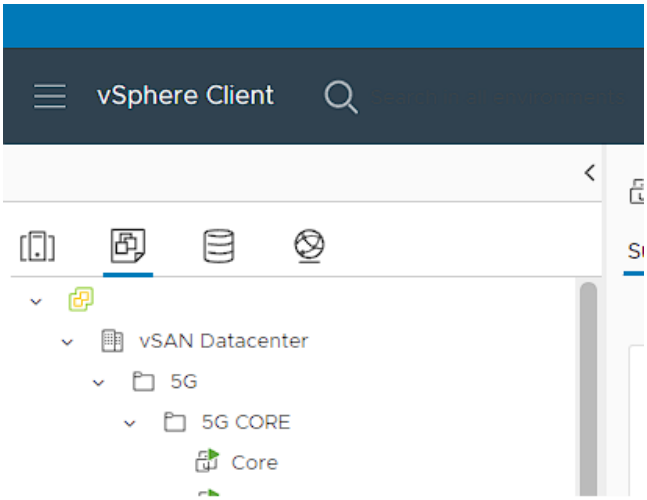


Figure 9: vSphere Client, where VMs are allocated

Even though all of them are located inside the Control Room, the logical configuration keeps physical resources separated for the three virtual clusters of computational resources.

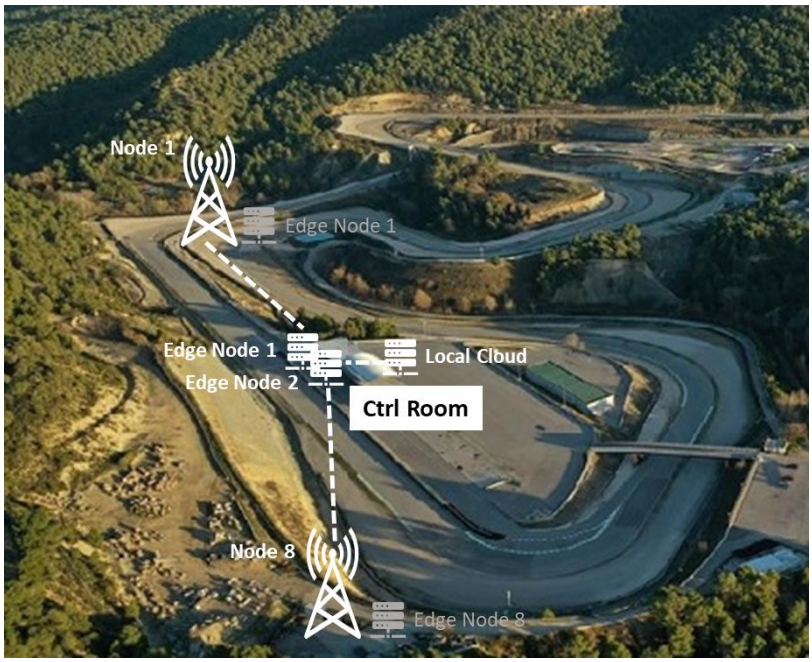


Figure 10: SUCCESS-6G virtual computational clusters

Figure 11: SR650 and SE350 Rack in the Control Room of the Cellnex Mobility Lab in Castellolí

The two distributed UPF are allocated in two SE350, identified as SE350#1 and SE350#2 in Figure 12. Each distributed UPF will remain in one of the two 5G Sunwave Radios.

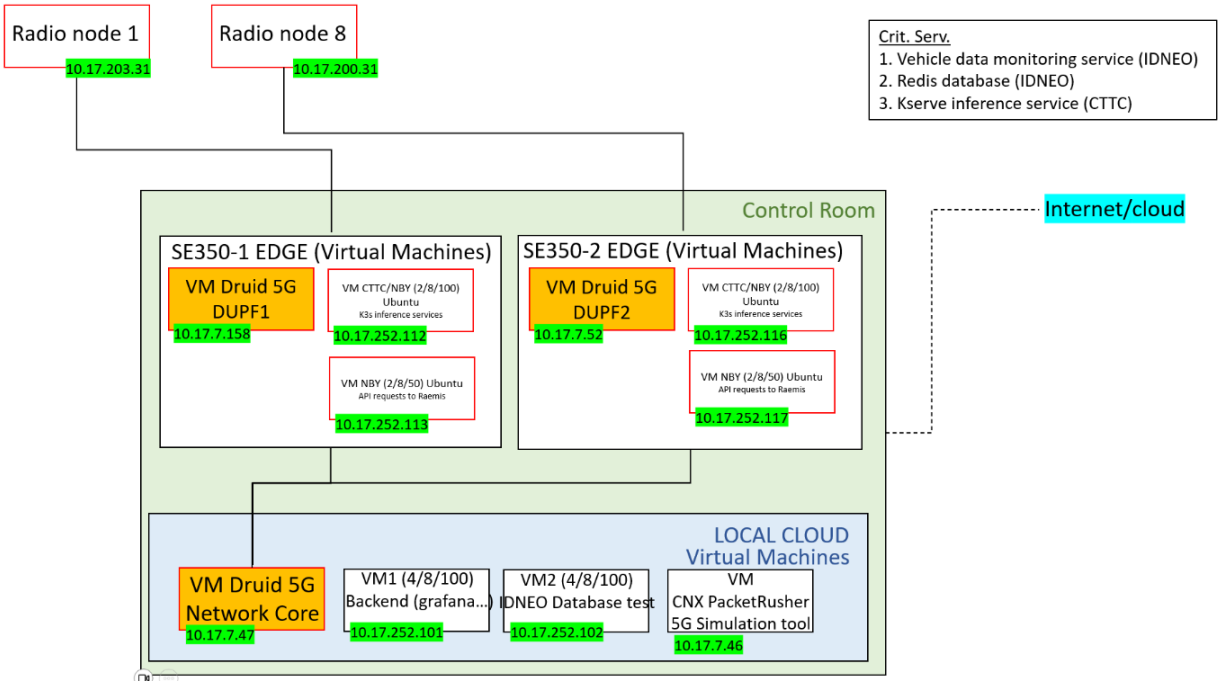


Figure 12: SUCCESS6G Computational Clusters and SW Services Architecture

To support the SUCCESS-6G project requirements, the three clusters of virtual computational resources host the following SW network services and applications, as in Table 3:

SERVICES AND NETWORK FUNCTIONS (LOCAL CLOUD: 2 x SR650)	CRITICAL SERVICES AND DUPF (EDGE NODES - SE350s)	
"Central Services": • VM Druid 5G SA Network Core • VM Backend (Grafana…) • VM IDNEO Database test • VM Network tools (Packet rusher, iperf3…)	**EDGE SE350 #1 - Node 1:** • VM Druid 5G Distributed UPF 1 • VM K3s inference services • VM API requests to Raemis (for migration services)	**EDGE SE350 #2 - Node 8:** • VM Druid 5G Distributed UPF 2 • VM K3s inference services • VM API requests to Raemis (for migration services)

Table 3: Services hosted by the three clusters

2.2 Data Acquisition

The data acquisition architecture deployed for the SUCCESS-6G use case 1 can be seen in Figure 13. The deployment of the applications involved migrating the V2X data parser and receiver from the vehicle and deploying a REDIS database on the server side. On the OBU side, the application responsible for use case 1 has been migrated from a European variant OBU to an American variant model, as the band used in the Castellolí tests is N77.

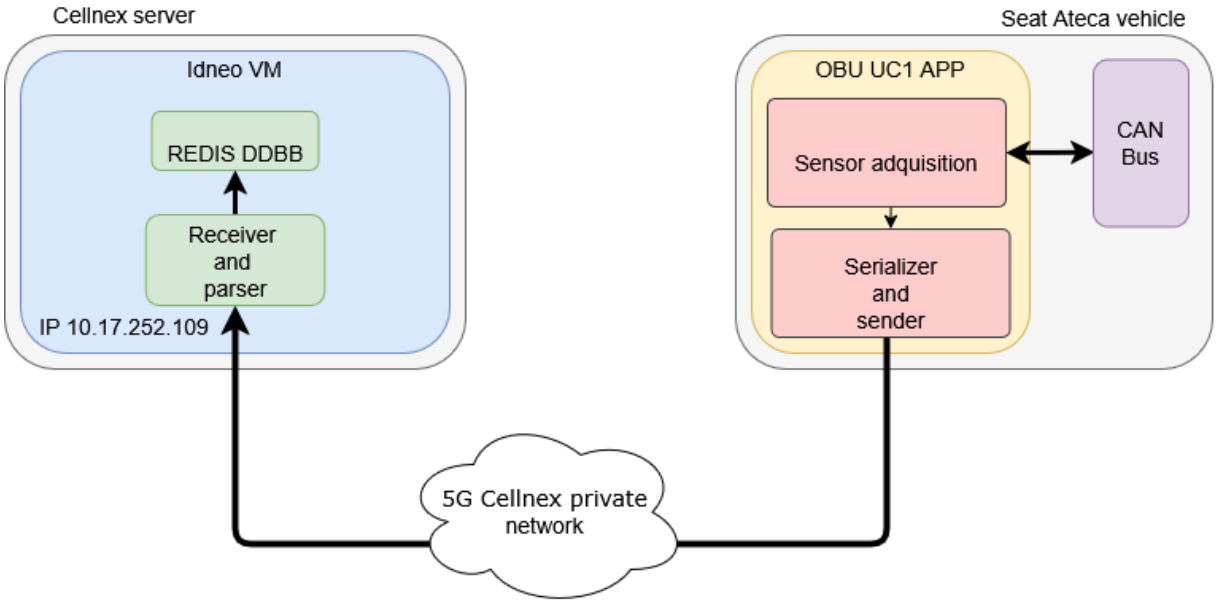


Figure 13: Data acquisition architecture

The deployment is carried out after a successful test of the connectivity at the network layer level, between the OBU and the Host that contains the applications. The test is a simple TCP/IP communication. In a ping test between the OBU and the Application Host, an average response of less than 17 milliseconds was obtained, as shown in Figure 14.

```
64 bytes from 10.17.252.102: seq=58 ttl=63 time=15.494 ms
64 bytes from 10.17.252.102: seq=59 ttl=63 time=14.426 ms
64 bytes from 10.17.252.102: seq=60 ttl=63 time=17.580 ms
^C
--- 10.17.252.102 ping statistics ---
61 packets transmitted, 61 packets received, 0% packet loss
round-trip min/avg/max = 10.422/16.385/36.854 ms
~ #
```

Figure 14: Successful connectivity test

The applications on the host that receive the data are containerized in Docker. Figure 15 depicts the reception of data in the deployed applications. On the left side, it is possible to observe the acquisition of binary data in the vehicle, while on the right side, the data is already being interpreted in its characteristic units.

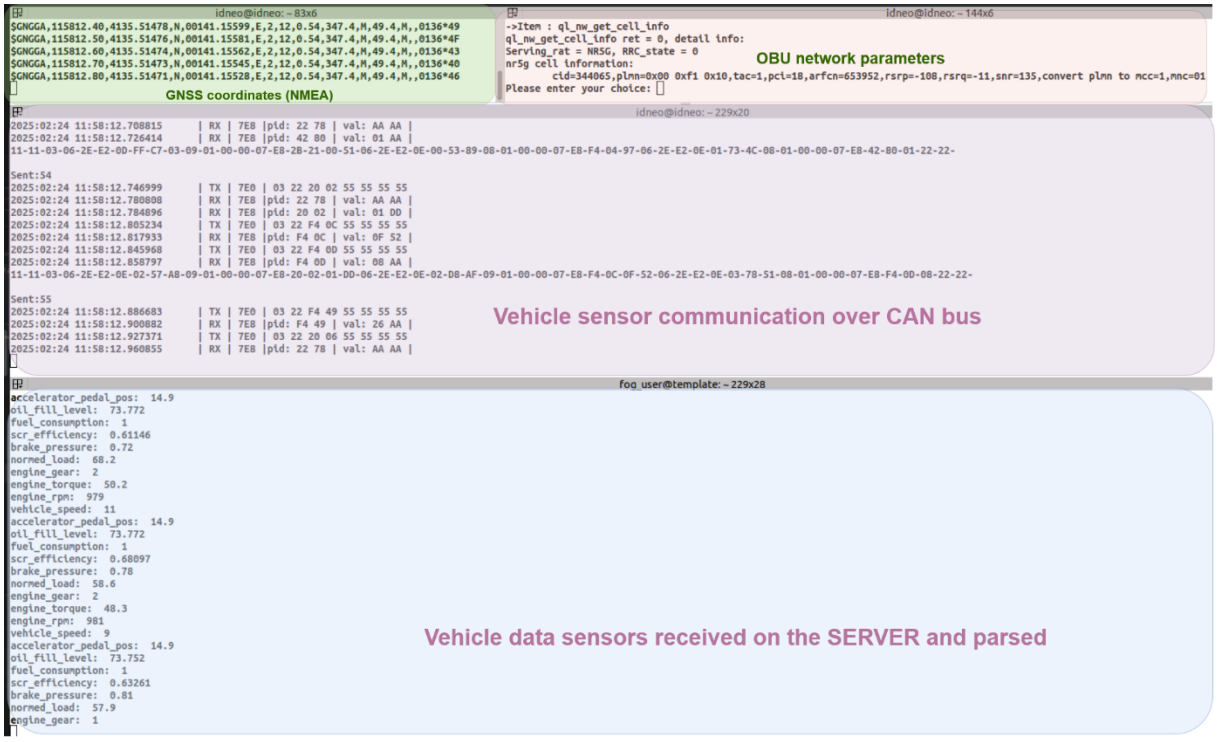


Figure 15: Reception of vehicular data

2.3 Condition Monitoring Service

The condition monitoring service during the test conducted at the Castellolí circuit is designed to evaluate and track the system's performance under real-world mobility conditions. During the test, the vehicle alternates between two 5G nodes (Edge_1 and Edge_2) while undergoing a **handover** [11]. Throughout the process, critical data on response times and system stability were gathered, offering an in-depth analysis of the system's performance in the test environment.

The primary objective of the monitoring service was to evaluate the system's performance based on:

- **Response Time**: The time elapsed between the detection of a handover and the moment the **kServe** service reaches the "UP" state (ready to operate).
- **Stability and Variability**: Fluctuations in response times during handovers, aiming to identify any anomalies or unstable behavior in the system.

The KPIs related to performance and response times monitored during the test are presented in the Section 4.3. These include key metrics such as success rate, average inference and response times, and queue size. Figure 16 represents the main components responsible for monitoring both inferences and performance metrics.

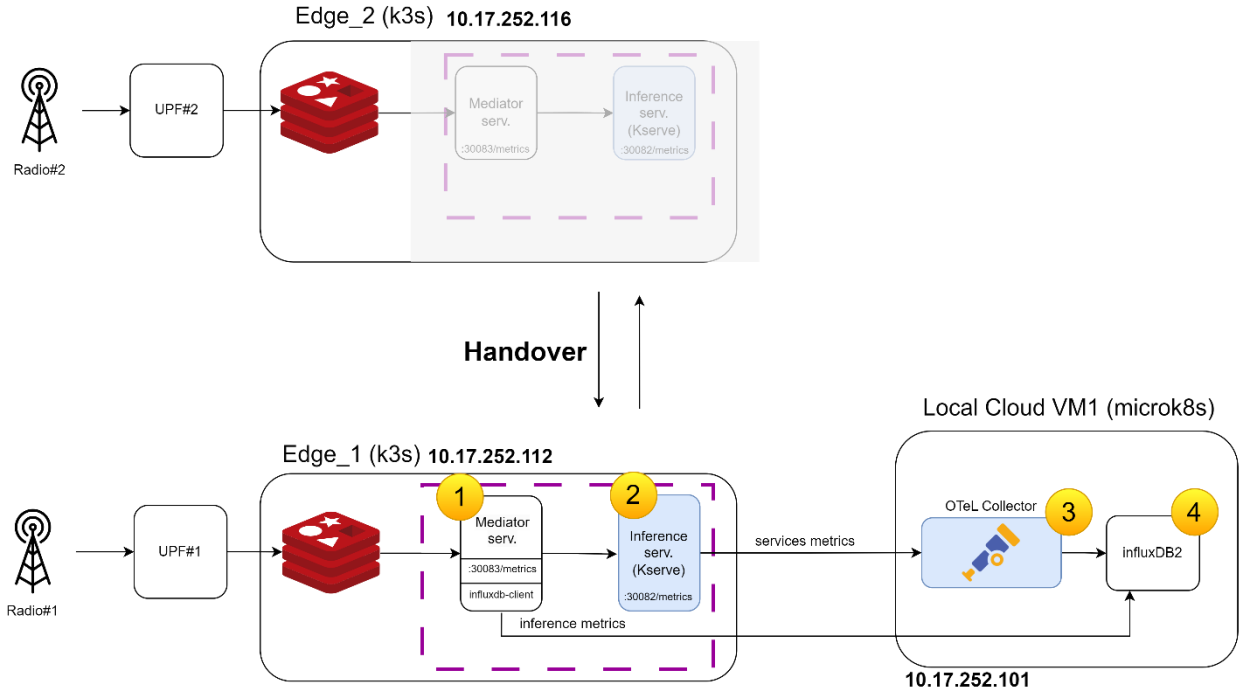


Figure 16: Conditioning monitoring service components

1. **Mediator Service**: This service is responsible for retrieving the vehicle metrics stores previously in the Redis (cache database), processing them, and sending them to the kServe inference service. Additionally, the Mediator joins the inference metrics with vehicle metrics ones, and those are sent to the observability backend InfluxDB2 for storage and further analysis.
2. **kServe (Inference Service)** performs the model inference and, in turn, exposes performance metrics via a Prometheus client. This allows real-time monitoring of metrics related to the inference service's performance, such as response times and other KPIs.
3. **Open Collector** scrapes the inference service's performance metrics and sends them to the observability backend InfluxDB2.
4. **InfluxDB2** acts as the centralized backend where both inference and performance metrics are stored. The processed inference metrics and kServe performance metrics are sent to InfluxDB2 through the Open Collector, enabling detailed and centralized analysis of all the data over time.

Figure 17 illustrates the **handover metrics** collected in InfluxDB2 (snapshot). This image showcases the real-time performance data during the handover process between the two 5G nodes, Edge_1 (purple signal) and Edge_2 (blue signal), providing insight into the system's behavior and response times during the test:

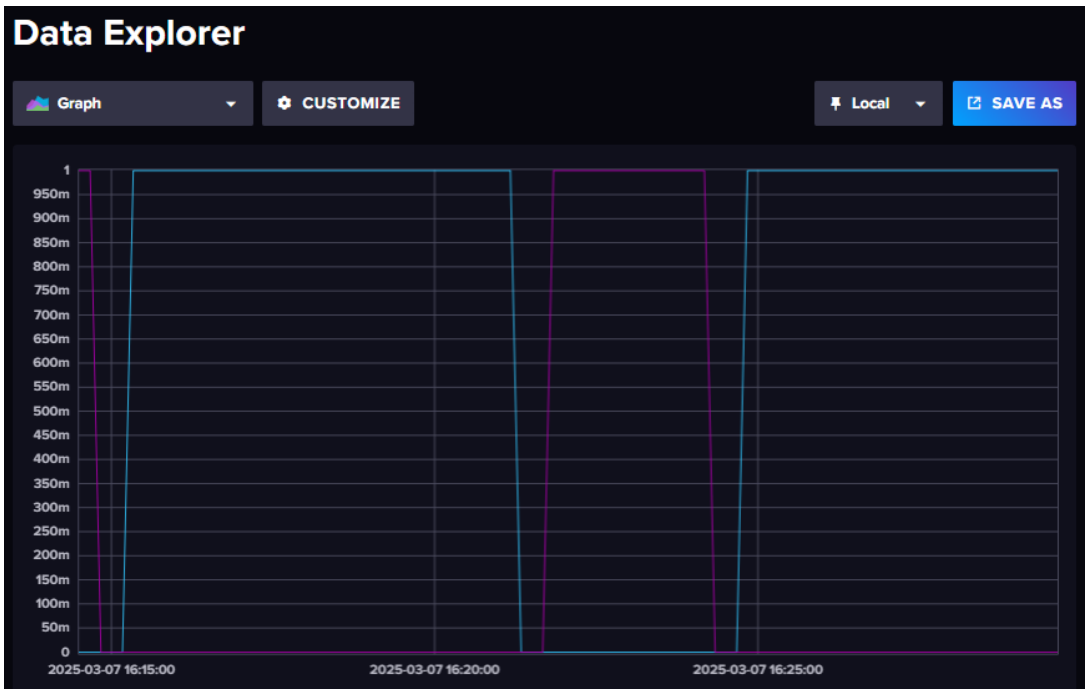


Figure 17: Handover metrics status operation

The following histogram in Figure 18 shows the distribution of response times recorded during the test. It visualizes how response times varied as the vehicle moved between the 5G nodes.

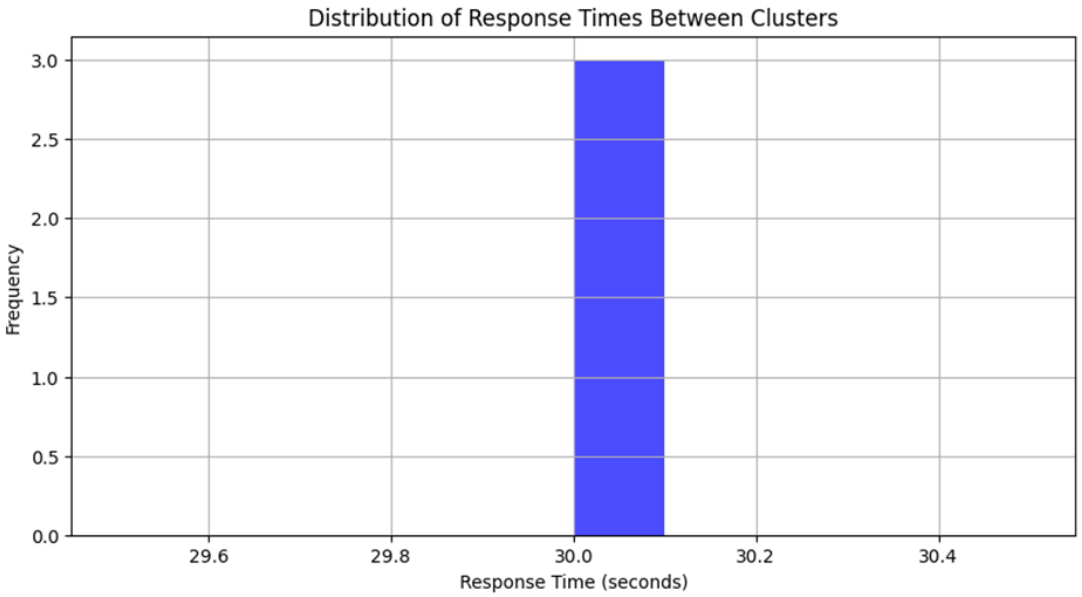


Figure 18: Response time distribution histogram

2.4 Service Orchestration

In the realm of telecommunications and networking, a service orchestrator plays a pivotal role in managing and coordinating the various components of service delivery. It oversees the orchestration of resources and processes to ensure the efficient provisioning, deployment, operation, and optimization of services across the network. Specifically, service orchestration encompasses the lifecycle management of services, including onboarding, deploying (instantiating), and terminating. These actions ensure the efficient provisioning and management of services across the Edge-Cloud Telco Continuum. Additionally, orchestration involves other key actions such as horizontal and vertical scaling, and migration (also known as service placement). The key aspects of service orchestration,

including lifecycle actions, scaling actions, migration, and their categorization into day actions, are outlined below:

- Lifecycle Management, which refers to i) onboarding, i.e., the process of adding a new service to the orchestration system, ii) deploying (instantiating), i.e., the allocation of resources and configuration of the service to run in the network, and iii) terminating, i.e., removing the service from the network and releasing associated resources.
- Scaling, which involves i) horizontal scaling, i.e., the increase of the number of service instances to handle higher loads or improve redundancy, and ii) vertical scaling that adjusts the resources (e.g., CPU, memory) allocated to a service instance to meet performance requirements.
- Migration (Service Placement), i.e., the relocation of a service from one site (physical or virtual node) to another to optimize resource usage, improve performance, or address failures. This may involve either deleting and re-deploying the service or deploying it at the destination site before removing it from the source site to ensure high availability.
- The aforementioned actions can also be categorized based on the timing of their operation in the network or infrastructure, referred to as Day 0, Day 1, and Day 2 orchestration activities:
    - Day 0: Onboarding and initial configuration of services.
    - Day 1: Deploying services and ensuring their proper instantiation and operation.
    - Day 2 (or Day n): Updating service configurations, either manually or automatically, and managing service operations dynamically.

Such an orchestration solution encompasses the essential capabilities that enable communication service providers to efficiently manage network resources, optimize service delivery, and ensure seamless operation. However, as networks evolve into programmable, software-driven, service-based architectures, they become more complex and require remarkable operational agility. The NearbyOne orchestrator can be used for onboarding and management of applications deployed in different edge clouds/decentralized architectures. In the scope of this project, NearbyOne manages service migrations by monitoring vehicle mobility, ensuring seamless and efficient transitions of the monitoring service between edge locations based on events coming from the 5G core, as illustrated in high-level in Figure 19.

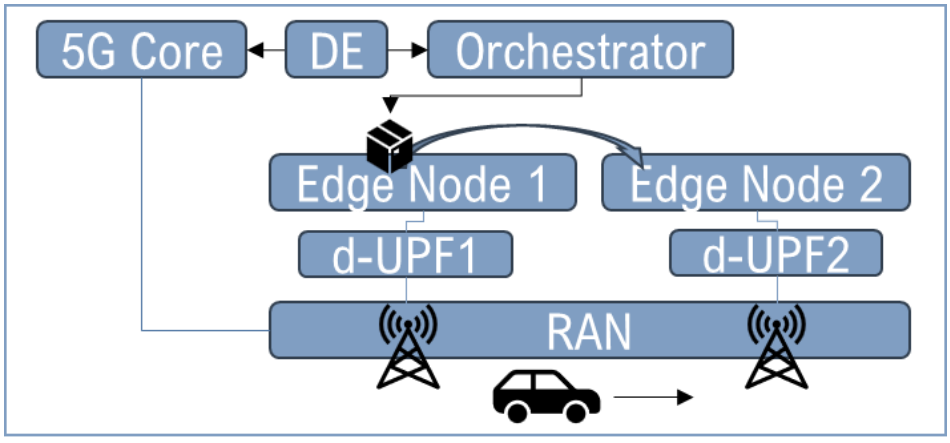


Figure 19: Service orchestration based on 5G core network events in mobility scenarios

3 Methodology for KPI collection

3.1 Trials at CELLNEX Mobility Lab, Circuit Parcmotor Castelloli

3.1.1 First round 19/12/2024

For the initial phase of testing, the 5G network registration procedures were executed using the On-Board Unit (OBU), and the quality of network parameters was assessed. Subsequently, the stability and reliability of the established network were evaluated to ensure that the applications to be deployed would operate without issues. As illustrated in Figure 20, the OBU modem was configured to operate on band n77.

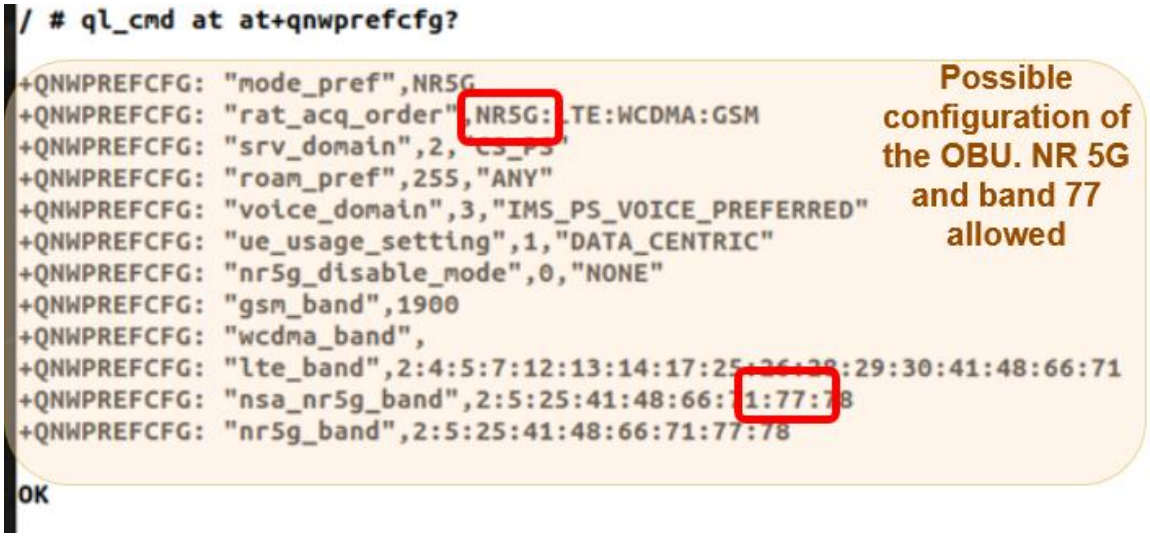


Figure 20: Configuration of OBU

The network was successfully detected with the Public Land Mobile Network (PLMN) identifier 99999 after the network scan, as shown in Figure 21.

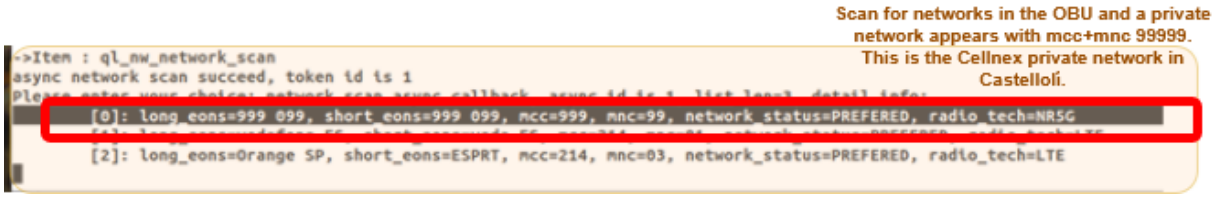


Figure 21: Network scan

The OBU successfully registered with the 5G network, after which the core network assigned it a Class A IP address, as shown in Figure 22.

Figure 22: Successful OBU network registration

Signal quality was analyzed at various locations within the test environment, along with bandwidth performance assessments using Iperf and round-trip time (RTT) measurements at each test point. Additionally, the initial versions of the data-receiving application were deployed, with a particular focus on evaluating network performance.

These connectivity tests were carried out in the vicinity of both base stations to fully validate the testbed. In addition, it was also possible to detect how the handover was working, even though the policies were not well adjusted for this purpose, as shown in Figure 23.

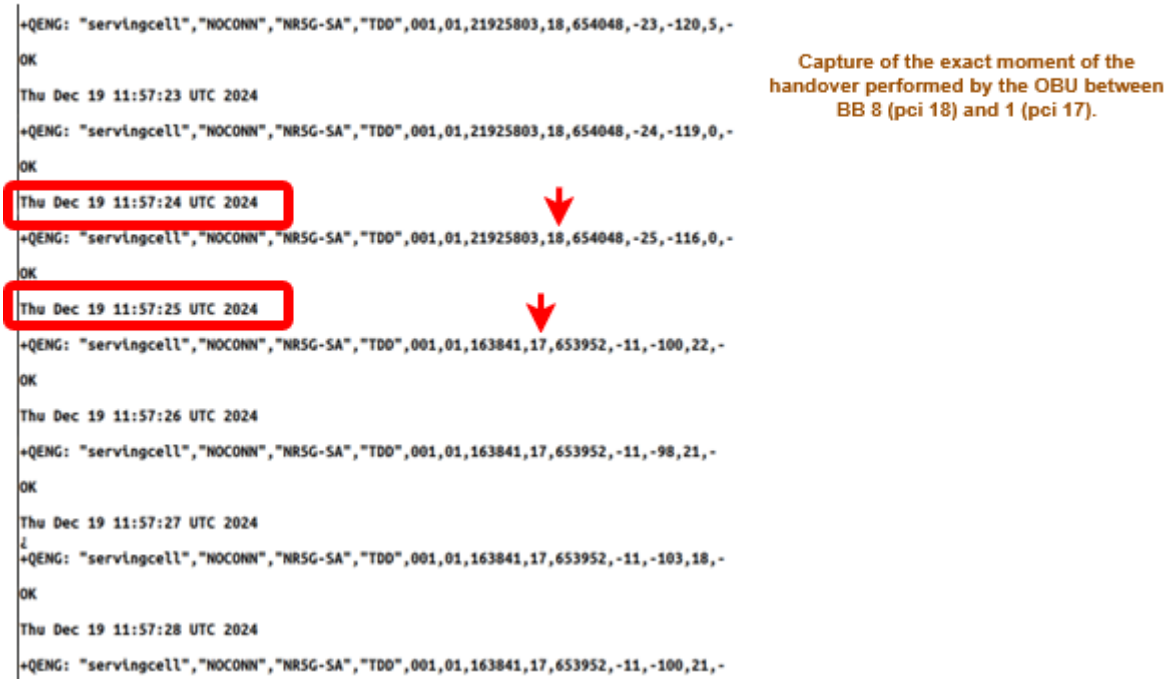


Figure 23: Handover phase

Before the SUCCESS-6G first round of end-to-end tests was performed in the Mobility Lab of Castelloli, more than one year before, Cellnex started collaborating with IDNEO in several tests focused on the 5G RAN deployed in the Castelloli Mobility Lab, to validate 5G connectivity and ensure the proper attachment of the OBU device to the 5G network deployed in Castelloli. Several series of tests were performed, increasing every time more network functionalities, up to achieving testing the whole end-to-end scenario: communication through the SUCCESS-6G 5G network between the vehicle and data servers deployed in the local and edge cloud.

This preliminary set of tests was executed according to the development of the SUCCESS-6G research activities and the outcomes provided by them. Thus, the first series of tests only validated that the OBU can be registered in the 5G network and get access to internet services. Then, after the deployment of the VM for IDNEO SW services in the Local Cloud, the tests were focused on the connectivity between the car and the REDIS database deployed in the IDNEO virtual machine. Besides the tests performed together with IDNEO, Cellnex has also tested several network settings to maximize data bandwidth between User Equipment and App Services.

In addition, related to the automatic SW recovery after a server reboot, to enhance the robustness and reliability of the 5G network, specialized scripts have been developed to ensure that, following a power outage, 5G services SW instances will not be deployed until it is confirmed that the Multus layer is fully active.

The tests began at approximately 10 AM. Both 5G nodes (node1-Edge 1 and node8-Edge 2) were activated. It has been configured 100M, 64QAM, 3D1U in the Radio.

- Conducted speed, bandwidth, latency, and QoS tests, with an average of 60Mbps and peaks of 100- 270 Mbps. It should be noted that we were conducting parallel tests, which might have affected the optimal use of the network.

We just disabled integrity from the BBU side for a quick test, as shown inFigure 24.

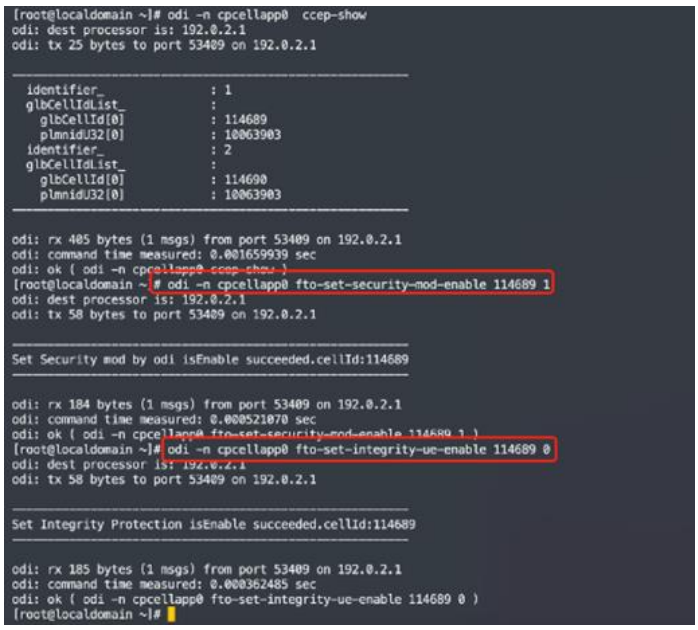


Figure 24: BBU configuration

After that, we ran the FTP again, and the following shows the test result.

- Node 1: FTP DL 93.74 Mbit/s

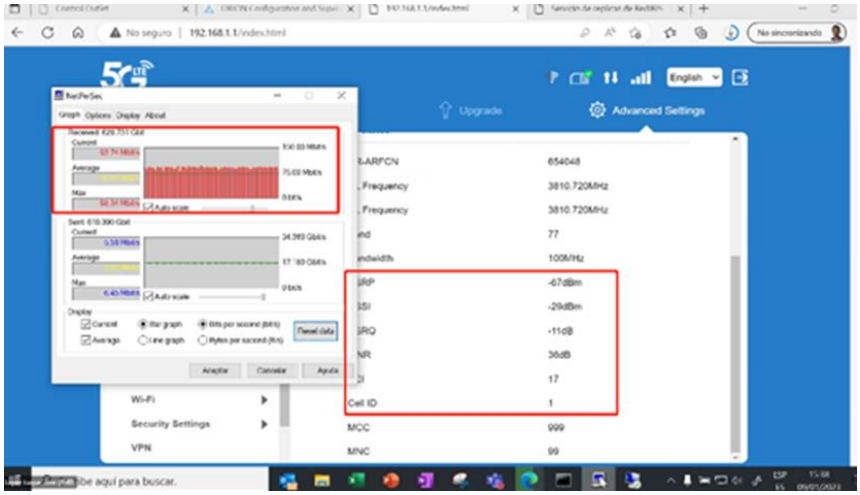


Figure 25: Node 1 throughput

- Node 8: FTP DL 51.93 Mbit/s

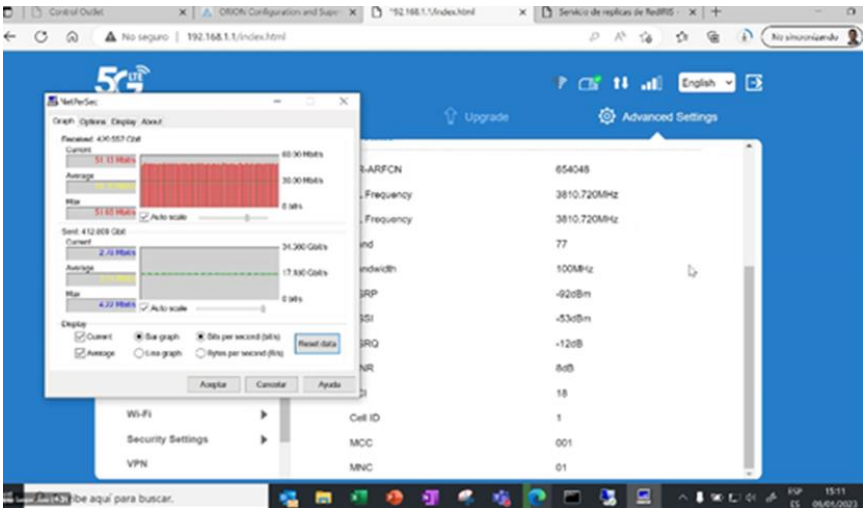


Figure 26: Node 2 throughput

The last objective of these tests was to evaluate the real-time automatic handover between radio nodes. During these tests, we observed that the handover between radio nodes did not occur automatically based on thresholds but was only possible manually. We identified an incorrect configuration, which was subsequently resolved after these tests.

Raemis dashboard shows the attached devices using the 5G SA network during the tests:

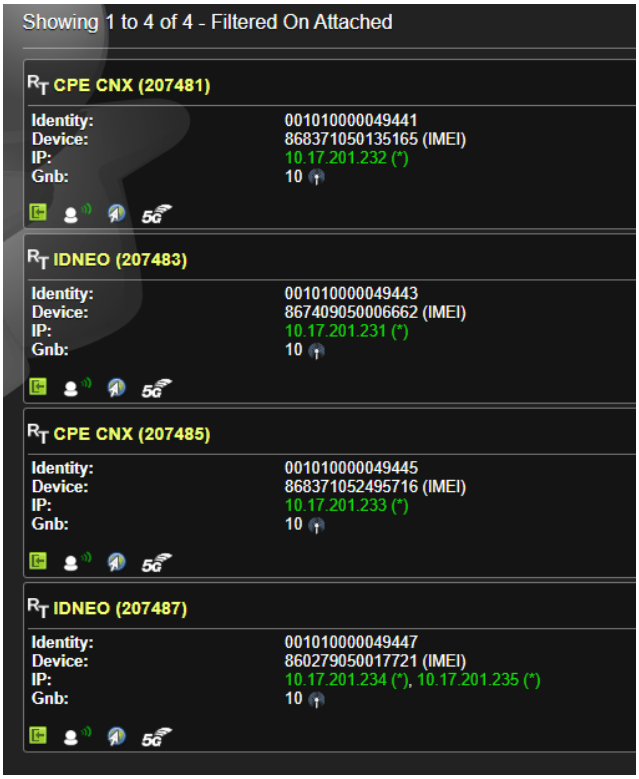


Figure 27: Attached devices in the Raemis dashboard

Some network KPIs extracted during the tests:

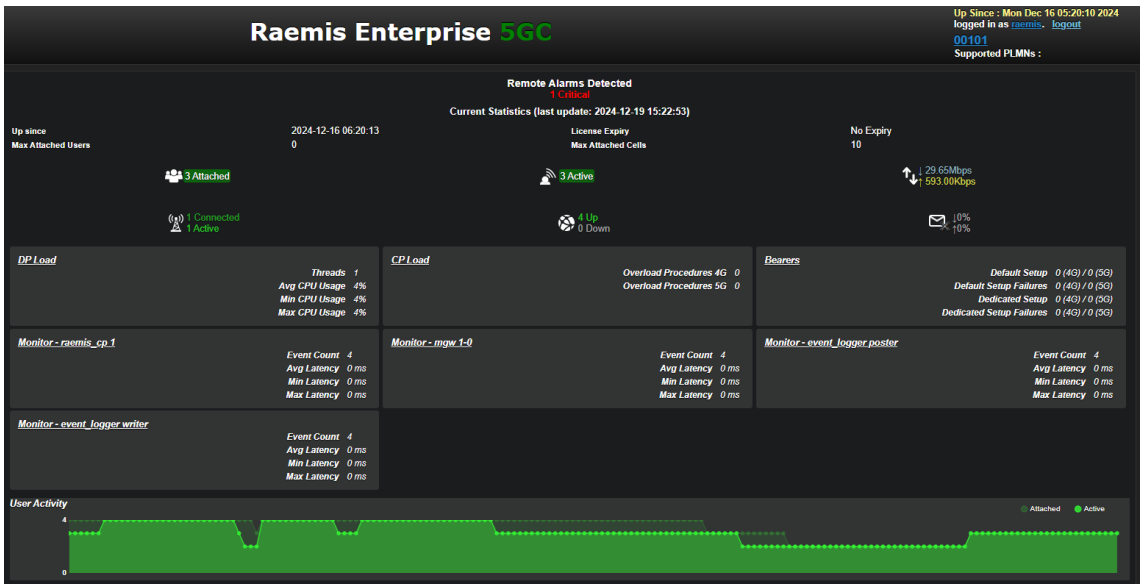


Figure 28: Extraction of network KPIs I

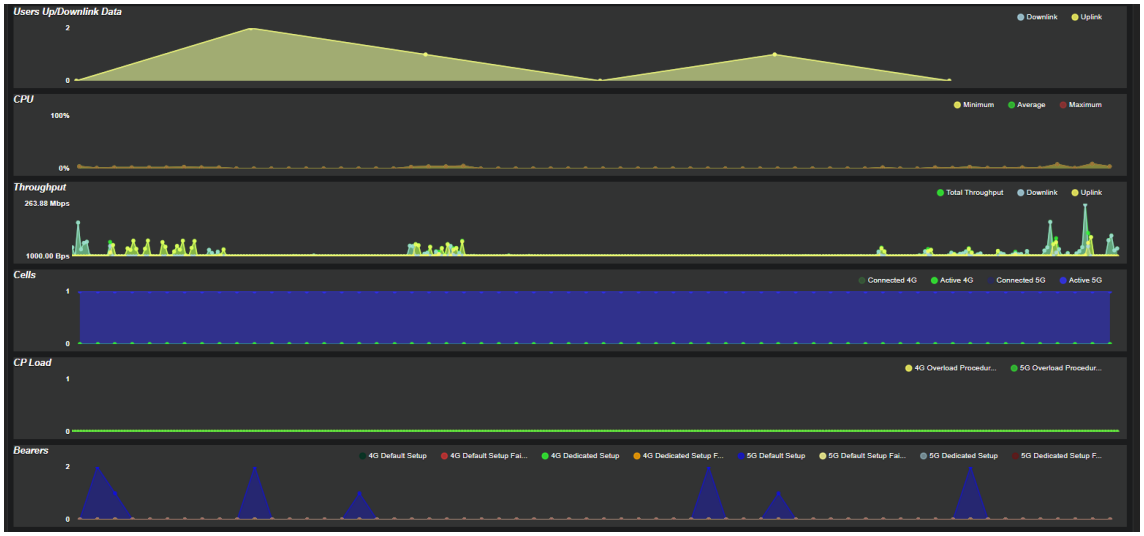


Figure 29: Extraction of network KPIs II

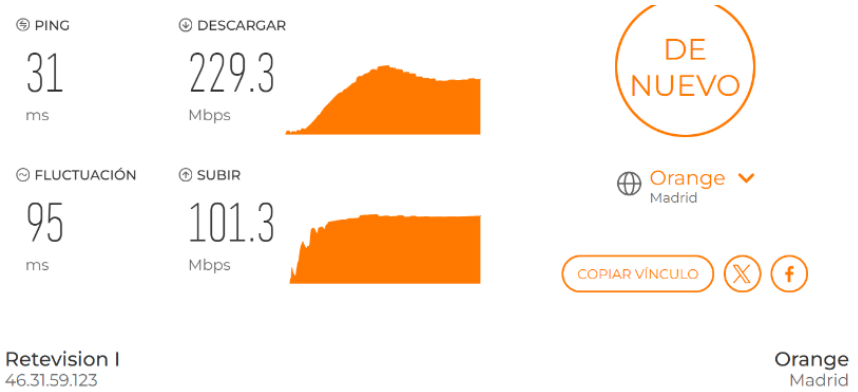


Figure 30: Network bandwidth test

In order to study the service orchestration based on 5G core network events in mobility scenarios, NBC has focused on the service migration functionality supported by the NearbyOne orchestrator. More specifically, during the series of trials performed in Mobility Lab of CELLNEX in the Circuit Parcmotor Castelloli, NBC has measured the service migration time that characterizes the handover process induced by the vehicle's mobility, considering two experimental scenarios: with and without the

inclusion of the Mediator, as defined in Section 2.3.

At first, we set up the Decision Engine (DE), as defined in [5], to handle all Subscription Permanent Identifiers (SUPIs) the same way. However, it was soon observed that some SUPIs needed special treatment, such as the SUPI of the vehicle that is moving between the base stations. We had to change our approach to deal with a specific SUPI, i.e., the one associated with the vehicle. This change was important because it made our system more accurate and efficient in monitoring the vehicle data sessions and handovers.

3.1.2 Second round 24/02/2025

On the second round of testing activities, the KPIs related to Use Case 1 were evaluated, and signal quality assessments were conducted within the circuit area. Six measurement points were established to analyze parameters such as RSRP, RSRQ, and SNR, which characterize the signal quality received by the 5G modem. Additionally, bandwidth and latency tests were performed to assess network performance.

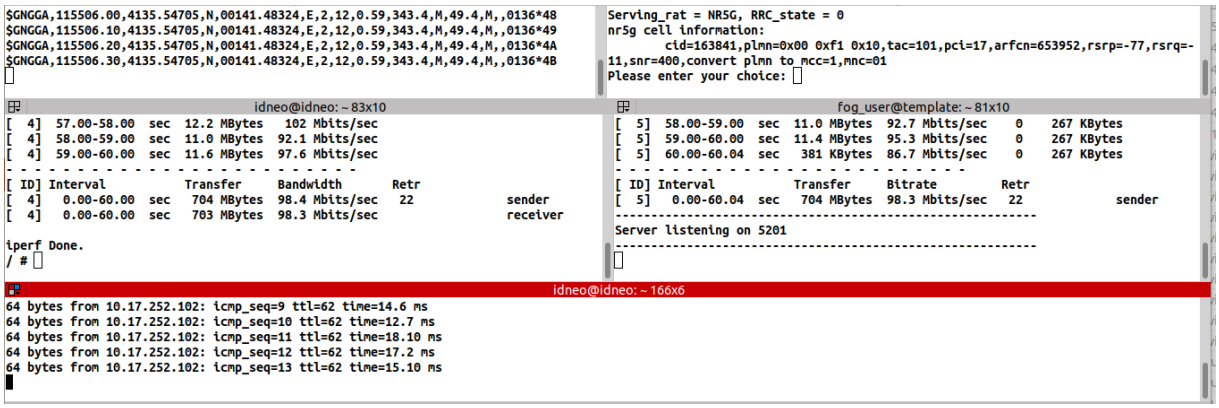


Figure 31: OBU connectivity during the second round of tests

To conduct the tests necessary to collect the KPIs for use case 1, an initial test plan has been designed with measurements to be taken at 6 points relative to node 1, as shown in Figure 32. As these points are in the line of sight of both nodes 1 and 8, they can be used as measurement points in the future. At each of these points, latency, download and upload throughput, and signal quality measurements have been taken.

Figure 32: Castellolí testbed and reference points for measures

In addition to the KPIs, Table 4 reports the measured values for throughput, RTT, RSRP, RSRQ, and SNR, where the behaviour of these parameters referenced to the six points can be observed.

POINT	TYPE	THROUGHTPUT (Mbps)	RTT (ms)	RSRP (dBm)	RSRQ (dB)	SNR (dB)
P1	UL	32	13.3 – 20.1	-77	-11	40.0
P1	DL	98.4	12.7 – 18.1	-77	-11	40.0
P2	UL	31.8	13.5 – 18.4	-90	-11	28.5
P2	DL	94.8	12.3 – 26.4	-91	-11	26.5
P3	UL	43	18.4 – 25.3	-106	-12	3.5
P3	DL	92.7	18.4 – 25.3	-106	-12	6.5
P4	UL	51.9	14.6 – 25.2	-103	-11	11.5
P4	DL	92.4	13.8 – 22.9	-102	-11	9.5
P5	UL	18.1	23.8 – 31.8	-103	-11	13.5
P5	DL	52.8	28.1 – 34.2	-106	-12	10.5
P6	UL	32	13.3 – 20.1	-77	-11	40.0
P6	DL	98.4	12.7 – 18.1	-77	-11	40.0

Table 4: Throughput, RTT, RSRP, RSRQ, and SNR measured at different points

The necessary services were deployed on the edge server. These services were containerized to enable a more flexible and modular deployment approach. On the OBU side, network traces were collected, and event and data counters were integrated into the vehicle's data acquisition application. These additions enabled the analysis of packet loss and the detection of potential performance issues. Additionally, the system was tested under the conditions of maximum vehicle speed to evaluate performance during high-mobility data transfers.

Due to the need to test the auto-handover between both nodes, Cellnex configured the RAN handover functionality. In particular, the following steps were followed:

Handover Function Starts Determine: Handover function startup decision refers to the process of deciding whether to start the handover function according to the startup conditions of each handover function, which mainly includes the following factors:

- Whether the switch of the handover function has been turned on.
- Whether the signal quality of the service cell meets the conditions.

Measurement Control Distribution：Measurement control distribution refers to the process in which the gNodeB sends measurement configuration information to the vehicle.

Any of the following scenarios will trigger the gNodeB to send measurement configuration information:

- When the vehicle enters the connection state, the gNodeB will send it to the vehicle through RRCReconfiguration Measurement configuration information.
- If the measurement configuration information is updated after the vehicle is in the connected state or the handover is completed, gNodeB will also send updated measurement configuration information to the vehicle through RRCReconfiguration.

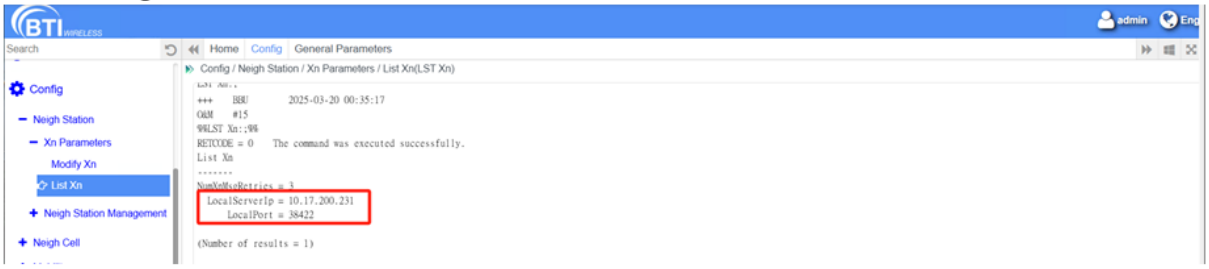
Measurement report reporting: After receiving the measurement configuration information sent by the gNodeB, the vehicle executes the measurement as instructed, filters the measured value according to the "filter coefficient", and then judges the event. When the event entry conditions are met, the vehicle will report the measurement report to the gNodeB.

Target cell or target frequency point decision: This process mainly includes the following three aspects

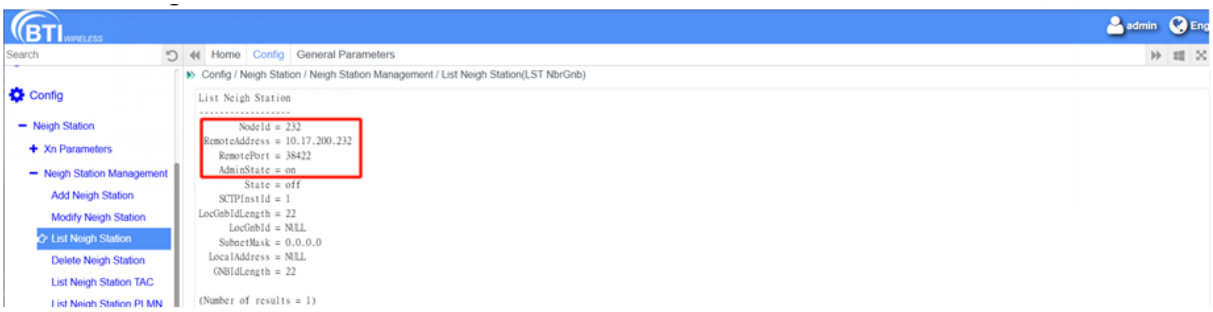
- Processing of measurement report.
- Determination of handover strategy.
- Generation of target cells or target frequency points.

Handover execution: After the decision on the target cell or target frequency point list is completed, the gNodeB will perform handover according to the selected handover strategy (including handover and redirection).

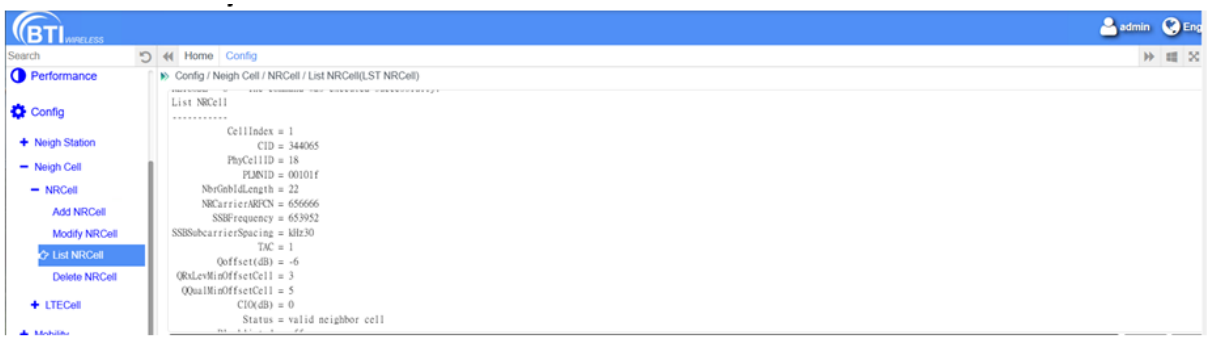
1. **Configure XN local address**



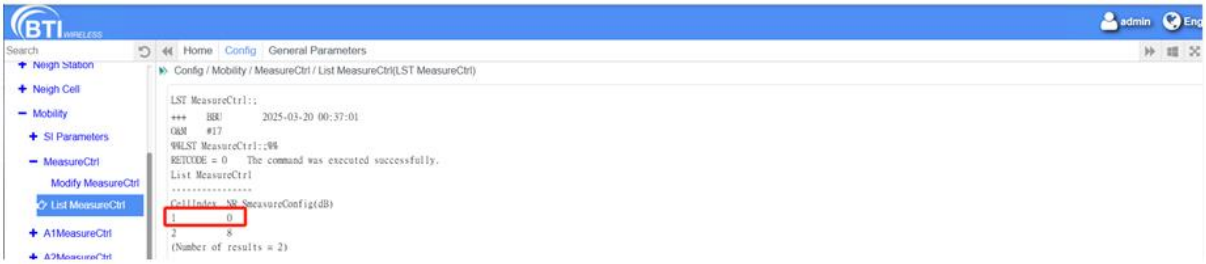
2. **Add Neighbor Station**



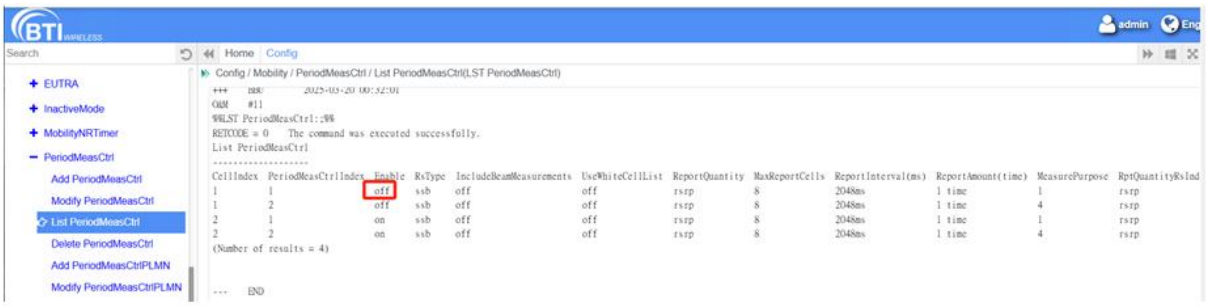
3. **Add a cell to the adjacent station**



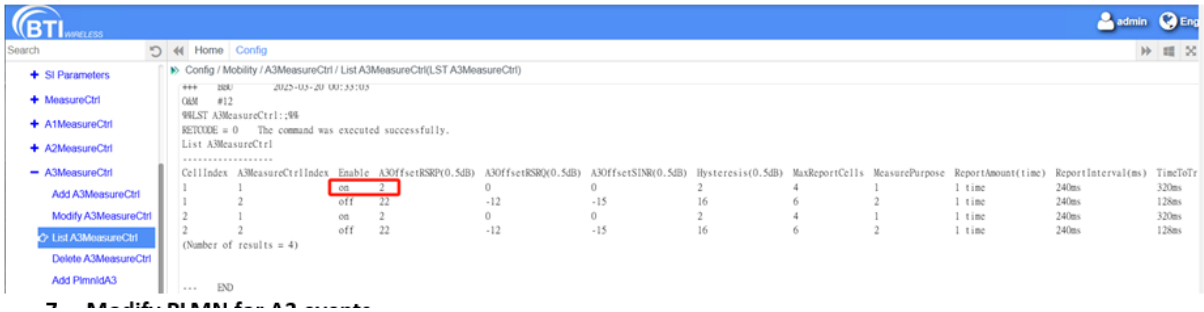
4. **Modify NR measurement start threshold (P91)**



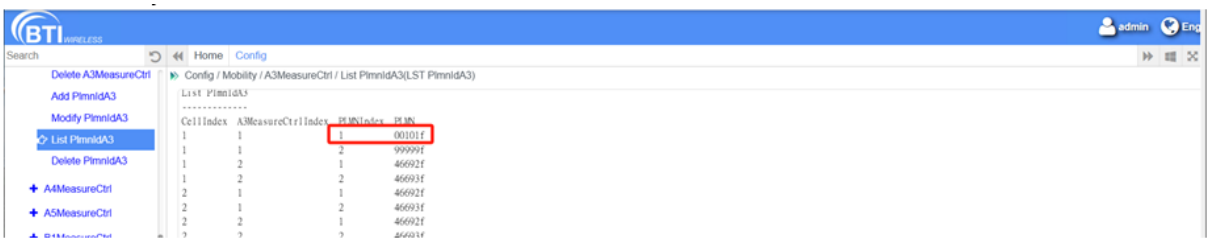
5. **Modify periodic measurement config**



6. Enable A3 Measurement



7. Modify PLMN for A3 events



After this functionality configuration, the handover between both nodes was applied and tested, working according to the expectations, as shown in Figure 33:

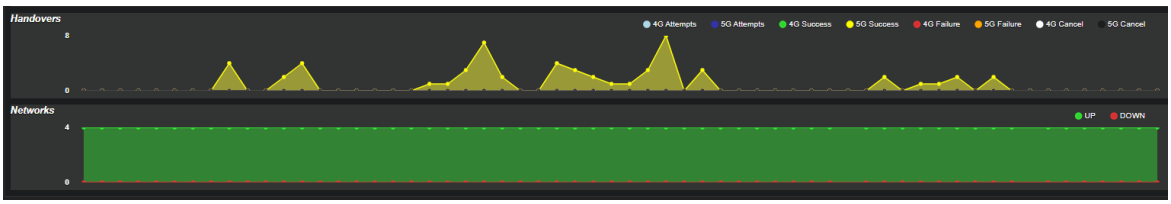


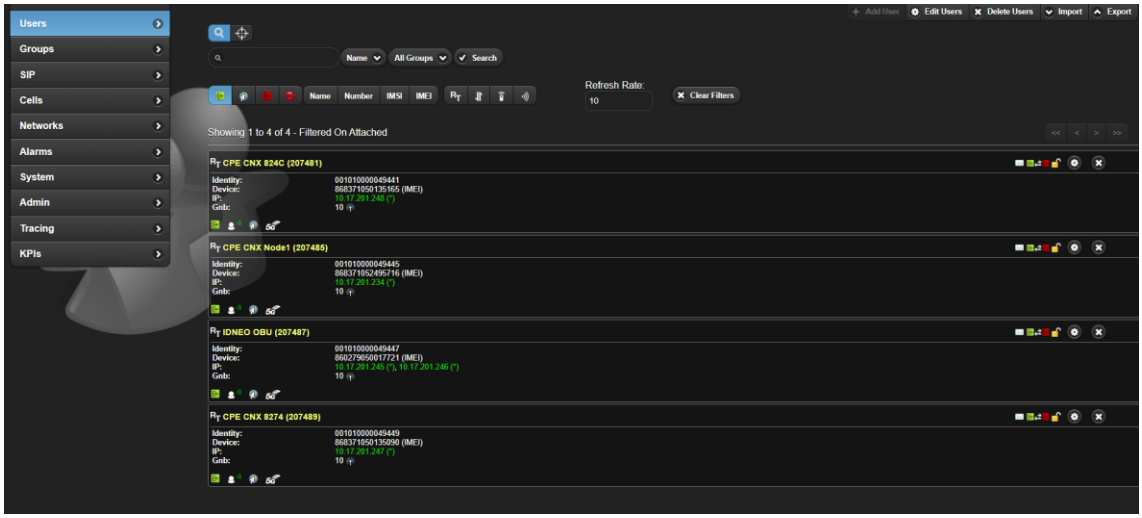
Figure 33: Successful handover

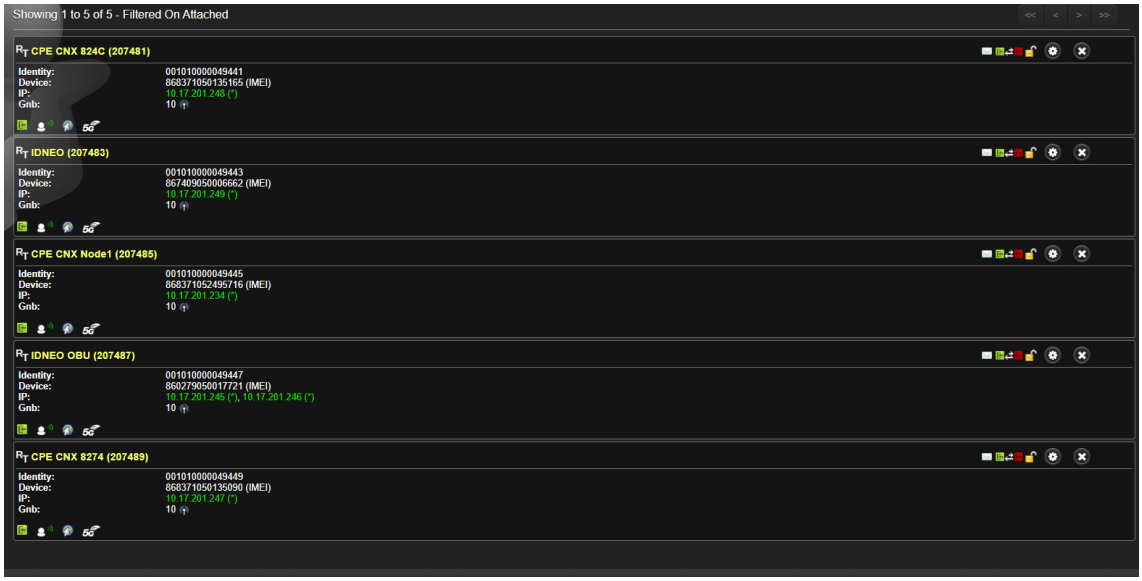
In the second round of testing, we encountered a significant issue related to our 5G core subscription model. Initially, the DE of the NearbyOne orchestrator was configured to subscribe exclusively to *http_api_create* requests. While this setup captured the creation of new data sessions effectively, it failed to account for the dynamic nature of our network, particularly the handover events. These handovers are crucial for maintaining seamless service continuity, and their omission led to gaps in our system's responsiveness. Recognizing this oversight, we expanded our subscription model to include *http_api_update* requests. This change was crucial in capturing all handover events, thereby providing a more comprehensive and accurate reflection of the network's state. By incorporating these updates, we significantly improved the robustness and reliability of our service, ensuring that all critical events were monitored and managed appropriately.

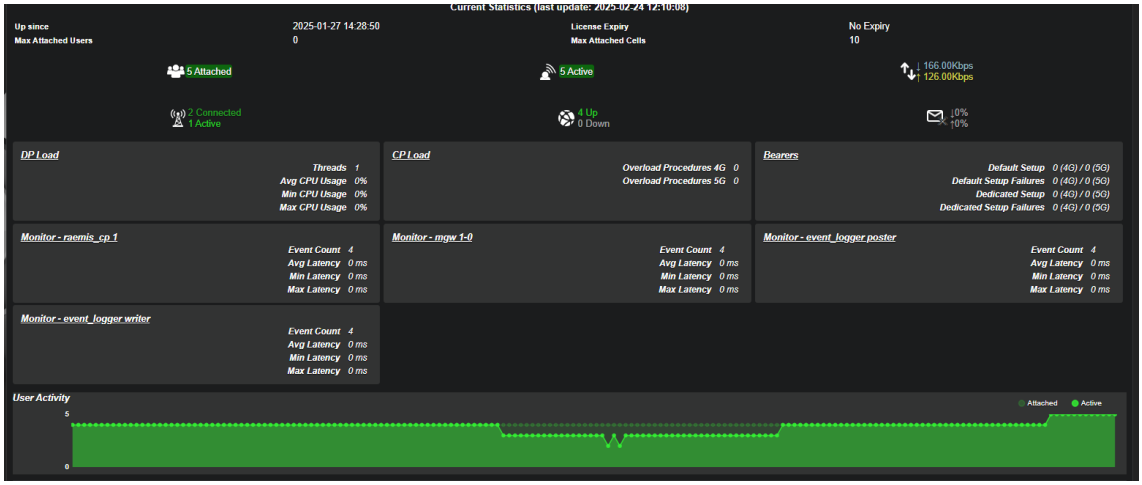
3.1.3 Third round 07/03/2025

After completing all necessary network configurations to optimally execute the use case tests, the experiments were successfully carried out in the third round of tests. Both nodes 1 and 8 were utilized, with automatic handover between nodes, and network monitoring and analysis conducted during the tests.

Figure 34 illustrates all the information captured in the Raemis dashboard related to the vehicle approaching node 1:







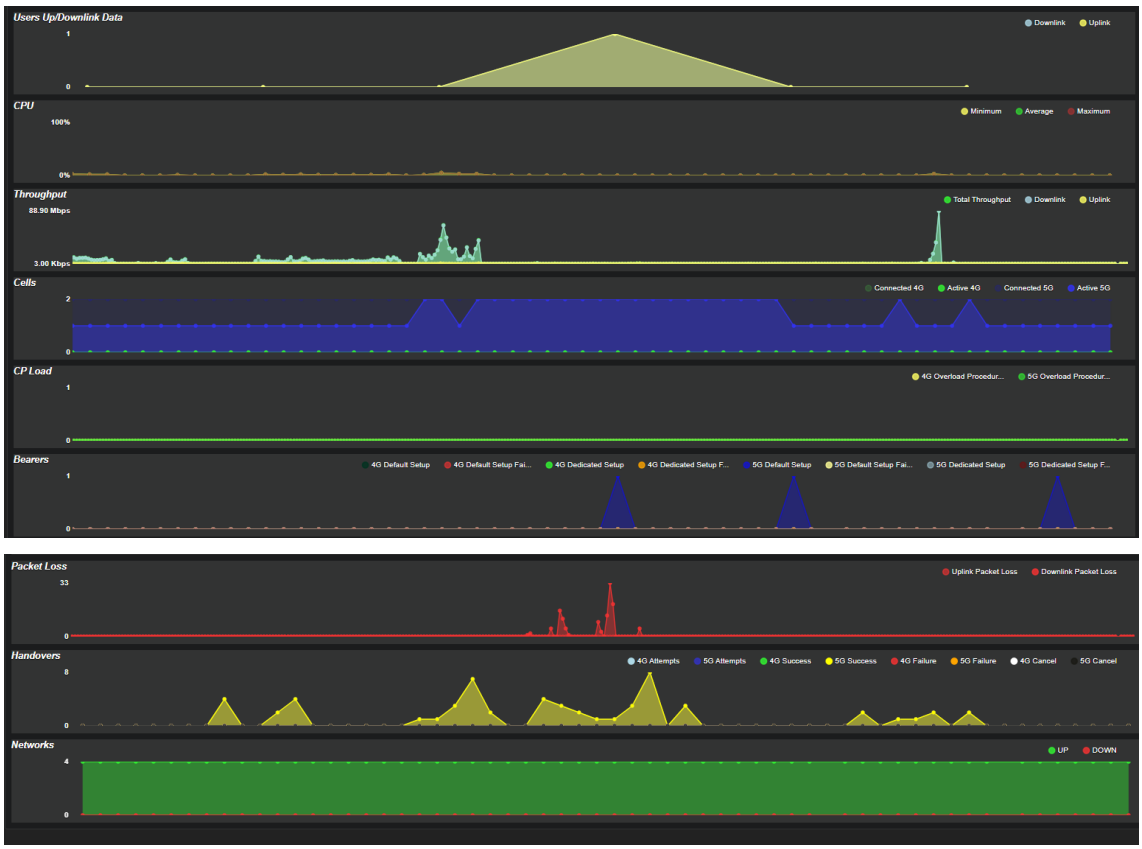
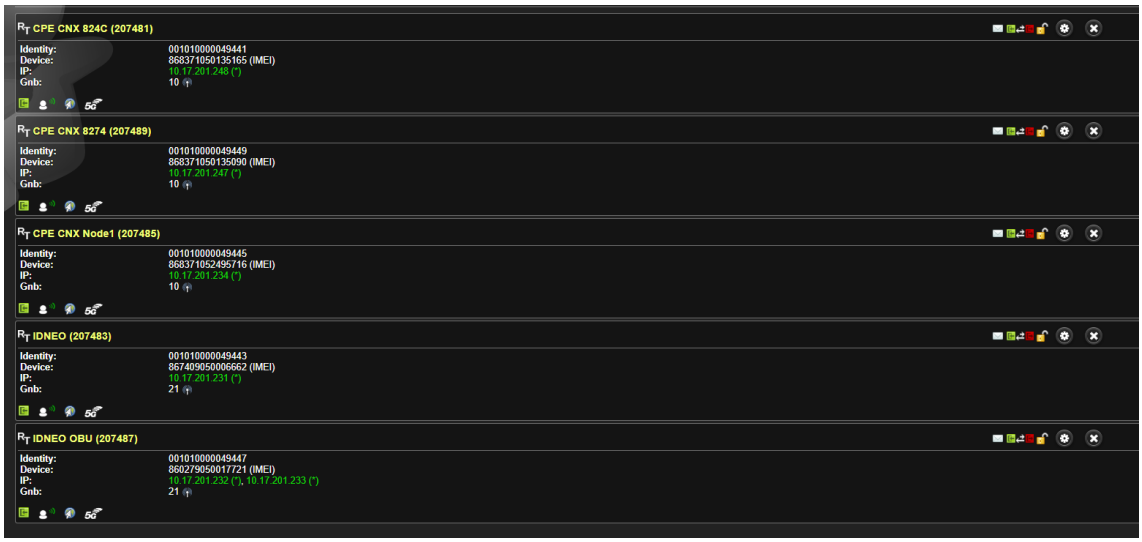


Figure 34: Vehicle approaching node 1

Accordingly, Figure 35 illustrates all the information captured in the Raemis dashboard related to the vehicle approaching node 8:



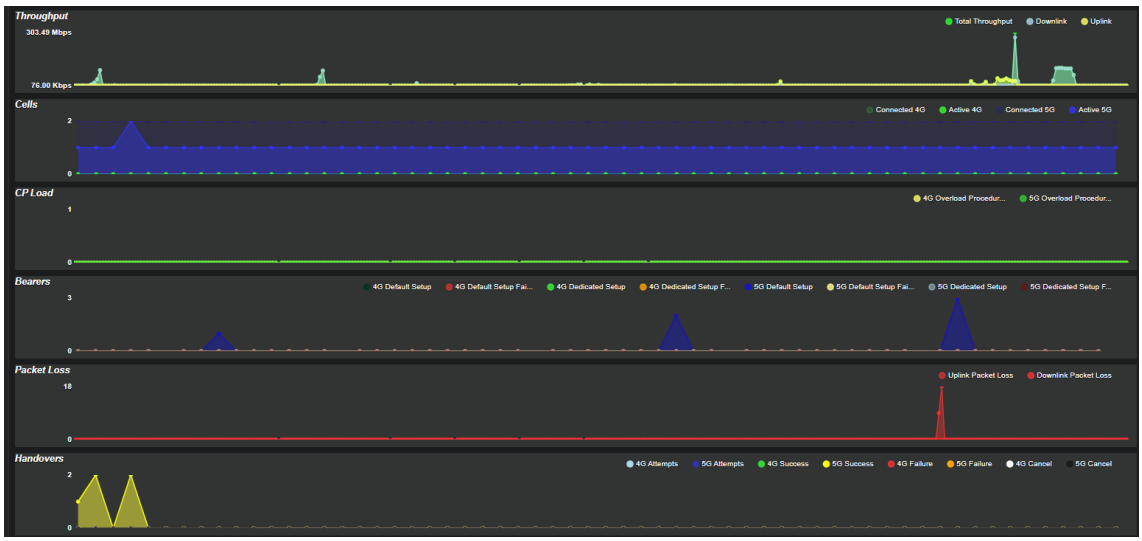


Figure 35: Vehicle approaching node 8

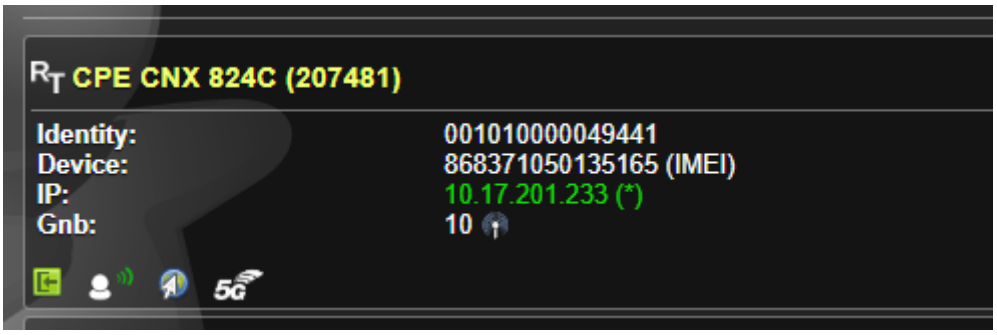


Figure 36: Iperf node 1

```
------------------------------------------------------------
Server listening on 5201
------------------------------------------------------------
Accepted connection from 10.17.201.237, port 50576
[  5] local 10.17.7.46 port 5201 connected to 10.17.201.237 port 50577
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec   896 KBytes  7.34 Mbits/sec
[  5]   1.00-2.00   sec  2.09 MBytes  17.5 Mbits/sec
[  5]   2.00-3.00   sec  2.36 MBytes  19.8 Mbits/sec
[  5]   3.00-4.00   sec  6.56 MBytes  55.1 Mbits/sec
[  5]   4.00-5.00   sec  1.75 MBytes  14.7 Mbits/sec
[  5]   5.00-6.00   sec  2.62 MBytes  22.0 Mbits/sec
[  5]   6.00-7.00   sec  2.06 MBytes  17.3 Mbits/sec
[  5]   7.00-8.00   sec  1.82 MBytes  15.2 Mbits/sec
[  5]   8.00-9.00   sec  3.23 MBytes  27.1 Mbits/sec
[  5]   9.00-10.00  sec  3.02 MBytes  25.3 Mbits/sec
[  5]  10.00-10.25  sec   240 KBytes  7.99 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.25  sec  26.6 MBytes  21.8 Mbits/sec                  receiver
```

Figure 37: Server (node 1)

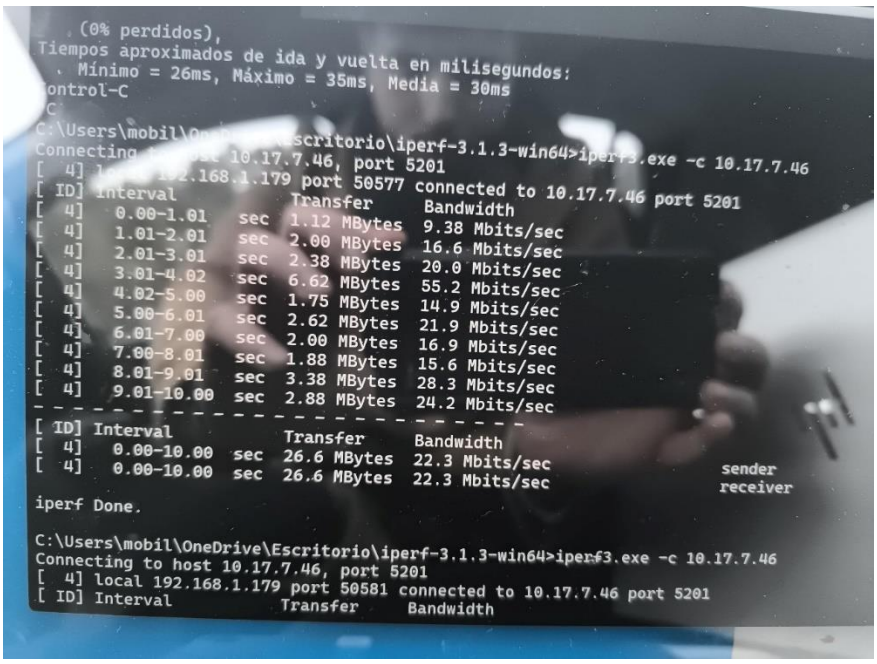


Figure 38: Client (node 1)

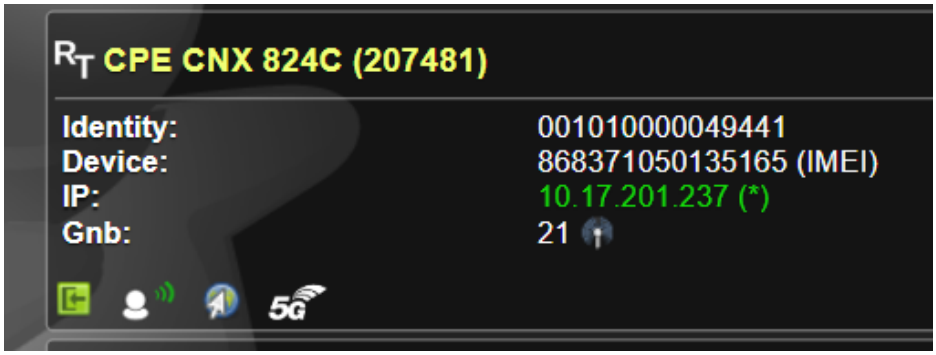


Figure 39: Iperf node 8

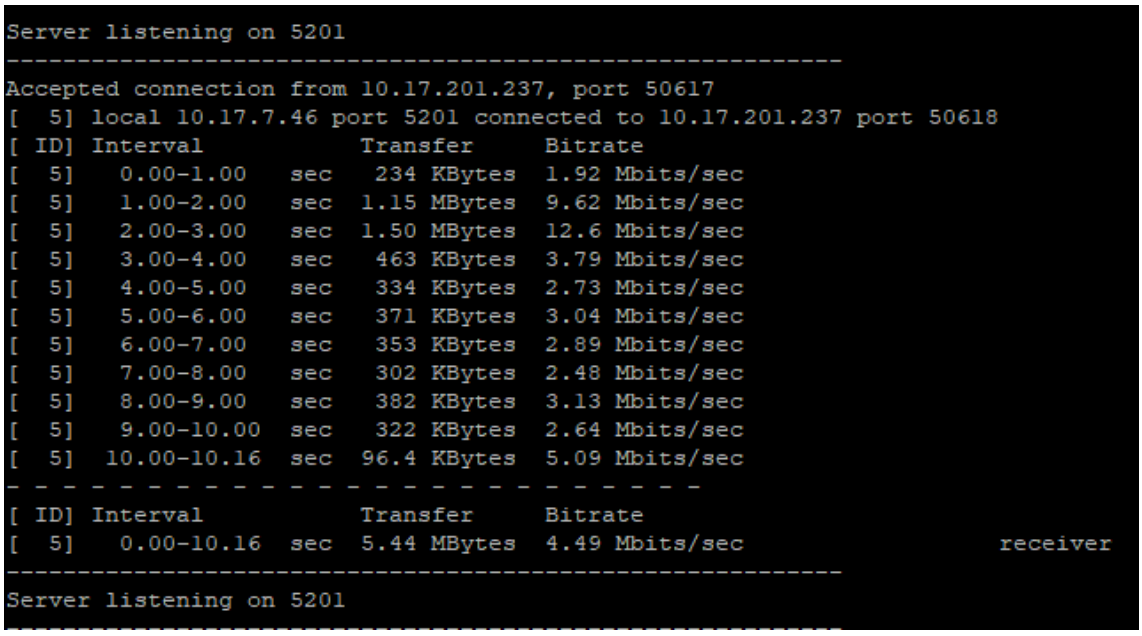


Figure 40: Server (node 8)

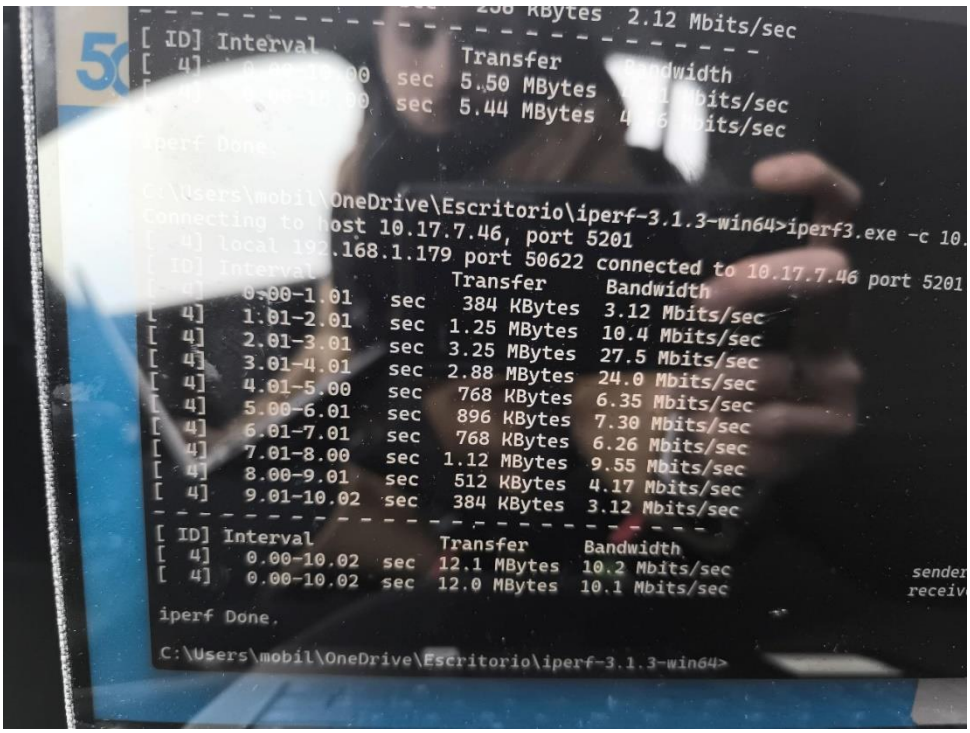


Figure 41: Client (node 8)

It is worth noting that the third round of testing activities brought to light another layer of complexity in our system. Although all handovers resulted in successful high-level migrations (NearbyOne level orchestration), we observed persistent low-level service failures (resource orchestrator level). These failures were traced back to Kubernetes (K8s) resource orchestrator reporting that the NodePort of the migrated service was unavailable. In addition, another reported issue was related to the migrated KServe service. This issue highlighted the critical role of the mediator application in our architecture. Without the mediator, the KServe service was treated merely as a custom K8s resource, which led to a significant oversight: the generated pod creation time was not captured by the high-level orchestrator. The KServe Helm chart was marked as ready based on its own criteria, without considering the actual readiness of the underlying Pod K8s resources. This discrepancy caused the high-level orchestrator to operate on inaccurate information, leading to service migration time miscalculations. As a result, the Pod creation time was measured separately on the K8s level.

3.2 Proof-of-concept at SUPERCOM

3.2.1 Anomaly detection and model prediction performance

3.2.1.1 Machine learning pipeline for anomaly detection in connected vehicles

A fully automated, scalable, and production-ready machine learning pipeline for anomaly detection in connected vehicles was designed to operate in real-time environments and under real driving conditions [11]. The open-source implementation of the service is publicly available at https://github.com/5uperpalo/success6g-edge, and a detailed description of the building blocks of our modular monitoring solution is also provided in [1]. This solution integrates the entire lifecycle of an anomaly detection system, including data preprocessing, feature selection, model training, hyperparameter optimization, explainability, and deployment orchestration.

Our approach is based on real-world telemetry data from a SEAT Ateca R4 2.0 TDI, covering diverse driving scenarios (urban, highway, and idle conditions). The core of the system is a light gradient boosting machine (LightGBM) classification model for anomaly detection, leveraging its advanced features in processing large volumes of sequential sensor readings and detecting subtle patterns indicative of potential equipment failures. By training on historical sensor data, LightGBM can effectively classify different types of faults and determine their root causes. Its superior capability to detect anomalies with limited training data has been demonstrated in our previous work [2], where it was evaluated against deep learning models, including TabTransformer [7] and TabNet [8], which served as benchmark classifiers. LightGBM-based classifier was optimized using Optuna's Bayesian optimization and extensively analyzed with SHAP [3] to ensure transparency and explainability, a critical requirement for safety and diagnostic applications in connected vehicles. Furthermore, the pipeline is deployed in production using KServe, enabling scalable, low-latency inference services fully integrated with InfluxDB for telemetry monitoring and MLflow for model tracking, as also detailed in [5]. This architecture supports real-time anomaly detection, making the system suitable for predictive maintenance, security monitoring, and operational diagnostics in modern connected fleets.

As shown in Table 5, the training dataset consists of real telemetric data recorded under various vehicle operational states:

Dataset Name	Scenario
DS1_stopped_with_ignition_on_22Feb24_115812.csv	Stopped with ignition on
DS1_stopped_with_ignition_on_25Jan24_124019.csv	Stopped with ignition on
DS1_stopped_with_ignition_on_25Jan24_151531.csv	Stopped with ignition on
DS1_stopped_with_ignition_on_25Mar24_153740.csv	Stopped with ignition on
DS2_national_road_90km_h_max_25Jan24_153019.csv	National road driving (90 km/h)
DS2_national_road_90km_h_max_25Mar24_133516.csv	National road driving (90 km/h)
DS3_highway_120km_h_max_22Feb24_121145.csv	Highway driving (120 km/h)
DS3_highway_120km_h_max_25Mar24_154857.csv	Highway driving (120 km/h)

Table 5: ML training data sources

As illustrated in Figure 42, the pipeline is composed of modular stages:

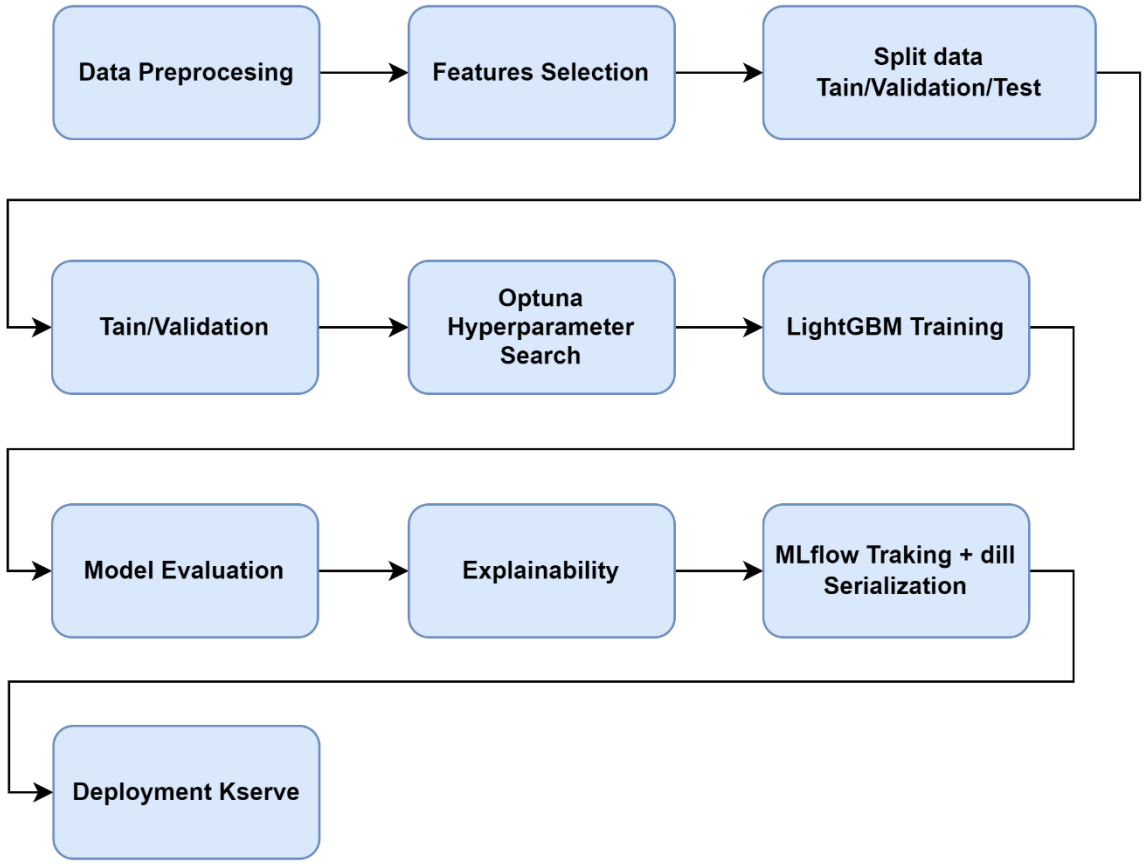


Figure 42: Anomaly detection ML pipeline

After loading, datasets were concatenated, cleaned, and preprocessed to remove constant columns, unify formats, and standardize variable names. A synthetic anomaly class (0/1) was assigned to simulate detection scenarios. The following preprocessing steps were also conducted:

- Dropping high-NaN/constant columns.
- Removing highly correlated variables.
- Imputation and handling of rare categories.
- Standard scaling for continuous features.

It is noted that the inference model requires as input the following specific features of the vehicular measurements to perform correct classification:

- Vehicle_speed
- Time_since_engine_start
- Normed_load_value
- Accelerator_pedal_position
- Engine_torque
- Oil_fill_level
- Engine_oil_temperature
- Fuel_level
- Fuel_consumption
- Brake_pressure
- Engaged_gear_raw_signal_Bits_0_7
- Efficiency_of_the_SCR_catalytic_converter

To efficiently optimize hyperparameters in our LightGBM-based classifier, we leverage Bayesian optimization techniques using the Optuna library [9]. Unlike traditional methods such as grid search or random search, which explore the hyperparameter space in a brute-force or purely stochastic manner, Bayesian optimization builds a probabilistic model to predict promising or informative hyperparameter values, improving search efficiency over traditional grid/random search. As such, the search process is guided effectively based on prior evaluations, significantly reducing the computational cost. Additionally, we incorporate a pruning mechanism to facilitate the early stopping of unpromising trials based on intermediate validation performance. This further enhances efficiency by allocating computational resources to more promising configurations.

Hyperparameter	Value
lambda_l1	0.0002553053419191757
lambda_l2	0.000828301110340063
num_leaves	31
feature_fraction	0.5
bagging_fraction	0.8861062217231551
bagging_freq	7
min_child_samples	20
n_estimators	1000

Table 6: Best hyperparameters found

Metric	Value
Accuracy	1.0
F1-score	1.0
Precision	1.0
Recall	1.0
Confusion Matrix	[[35, 0], [0, 10]]

Table 7: Evaluation metrics

To ensure model transparency and validate the learned patterns, a feature importance analysis was conducted using the trained LightGBM model. Figure 43 depicts the key features influencing anomaly detection. It can be observed that the model focuses strongly on variables linked to engine load, time, torque, and critical performance sensors, while other variables (e.g., fuel level, brake pressure, accelerator pedal position) had no impact on the final decision-making.

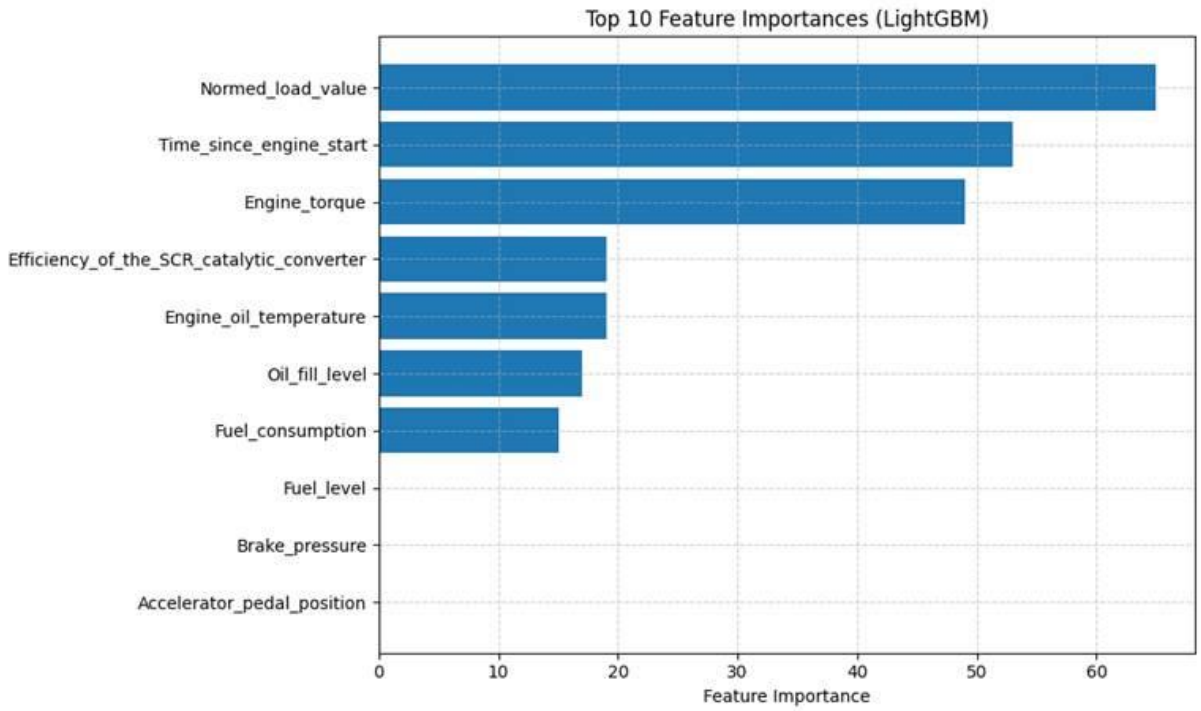


Figure 43: Features importance (LightGBM)

To add an additional layer of explainability, SHAP (SHapley Additive exPlanations) values were computed to analyze individual and global feature contributions of the model outputs. SHAP is a widely used explainability technique that leverages cooperative game theory to quantify the impact of each feature on the model's classification decisions [10]. The SHAP summary plot in Figure 44 offers insights into how features influence predictions toward normal or anomalous classes.

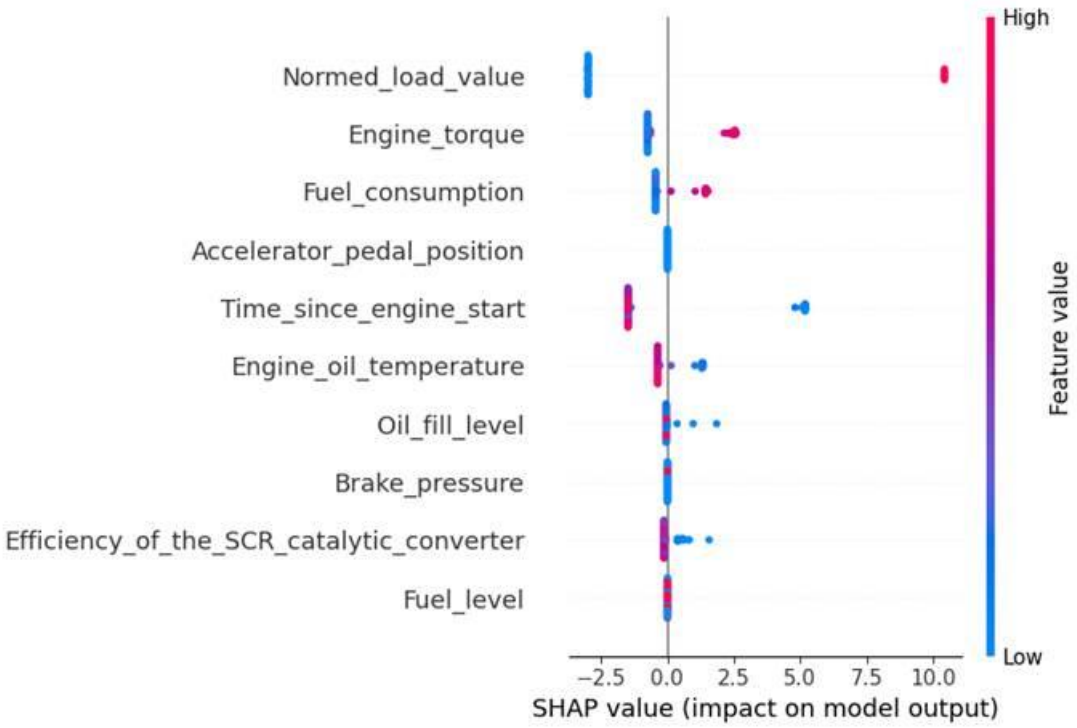


Figure 44: SHAP value impact on LightGBM model outputs

It can be noted that Normed_load_value and Engine_torque are the most influential in pushing predictions toward anomalies.

3.2.1.2 Evaluate model accuracy and KPIs for anomaly detection

This section presents a comprehensive evaluation of the anomaly detection model deployed and served in a real driving scenario, specifically during the Castellolí circuit test. The model was deployed using KServe, enabling scalable and low-latency real-time inference directly from vehicle telemetry streams. Data was collected and stored in InfluxDB, allowing a precise and systematic analysis of production inference behavior. The objective of this evaluation is to assess model performance based on key anomaly detection KPIs (Precision, Recall, F1-score) and to validate its effectiveness and responsiveness under real-world vehicle operation conditions.

The analyzed dataset contains 9,487 telemetry records, including predicted anomaly probabilities, and retains the same features as the training set, ensuring consistency for evaluation. The summary statistics of the main features are shown in Table 8.

Feature	Mean	Std	Min	25%	50%	75%	Max
Accelerator_pedal_position	16.93	9.62	0.00	14.90	14.90	14.90	86.30
Brake_pressure	1.61	29.29	0.00	0.00	0.00	0.00	655.33
Efficiency_of_the_SCR_catalytic_converter	0.76	0.25	0.00	0.64	0.88	0.92	0.98
Engine_oil_temperature	9.65	28.59	0.00	0.00	0.00	0.00	113.20
Engine_torque	38.75	45.51	0.00	25.80	28.60	32.40	392.90
Fuel_consumption	0.69	3.45	0.00	0.00	0.00	0.00	29.00
Fuel_level	2.47	7.58	0.00	0.00	0.00	0.00	28.00
Normed_load_value	29.68	15.80	0.00	23.80	25.70	28.70	97.70
Oil_fill_level	70.23	10.35	0.00	67.23	71.51	73.93	92.09
Time_since_engine_start	470.17	1634.49	0.00	0.00	0.00	0.00	10190.00

Table 8: Anomaly injection values

To systematically evaluate the model's sensitivity and robustness, we generated synthetic anomalies based on the clean feature dataset. This process ensures controlled and reproducible testing. To ensure anomalies are significant and detectable, the following amplified abnormal values were used across different features, as shown in Table 9.

Feature	Anomalous Value
Accelerator_pedal_position	200
Brake_pressure	2000
Efficiency_of_the_SCR_catalytic_converter	2.0
Engine_oil_temperature	250
Engine_torque	700
Fuel_consumption	70

Fuel_level	100
Normed_load_value	200
Oil_fill_level	200
Time_since_engine_start	-100

Table 9: Anomaly injection values

The following anomaly patterns are considered:

- Sparse Anomalies (Figure 45): Random isolated anomalies were injected into 1%, 5%, and 10% of the dataset to simulate point anomalies.

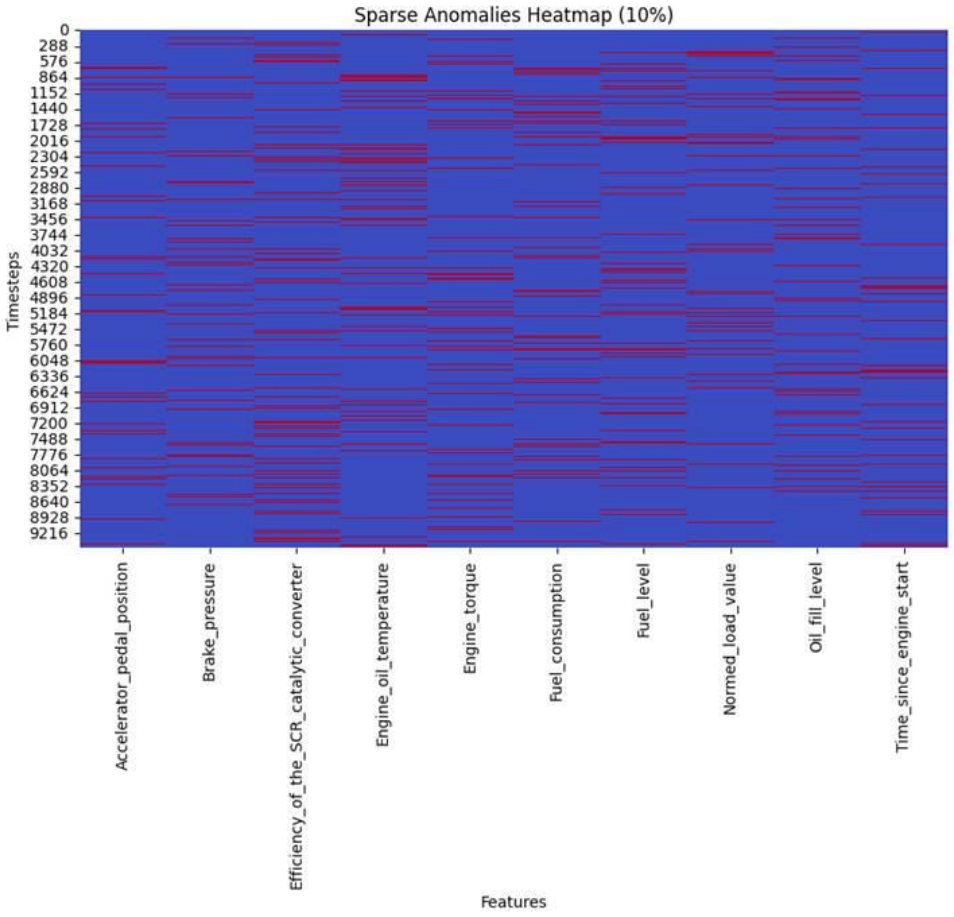


Figure 45: Sparse anomalies heatmap where synthetic faults are introduced in 10% of the dataset.

- Collective Anomalies (Figure 46): Sequential (temporal) anomalies were injected using variable numbers of time blocks (10, 100, and 200 blocks, each covering continuous time periods). In each block, half of the features were selected randomly for anomaly injection, creating partial system failures.

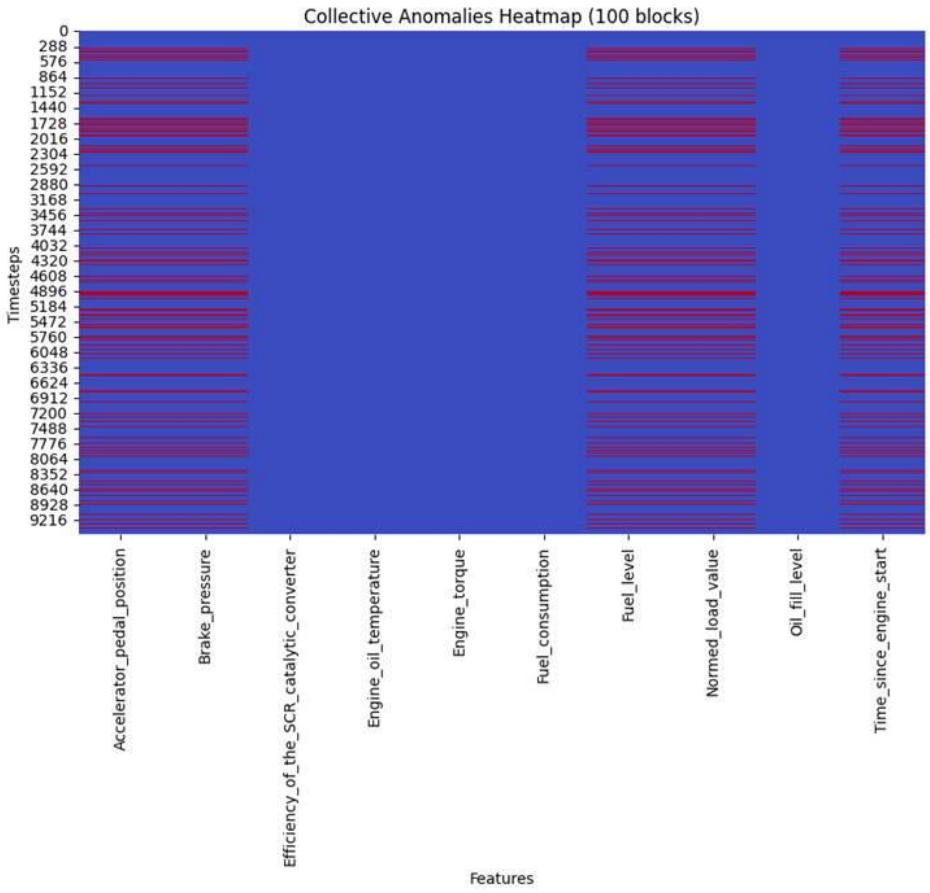


Figure 46: Collective anomalies heatmap for 100 consecutive timesteps.

Based on a manually labelled ground truth and expert-defined thresholds, the following performance metrics were computed, as shown in Figure 47 and Figure 48.

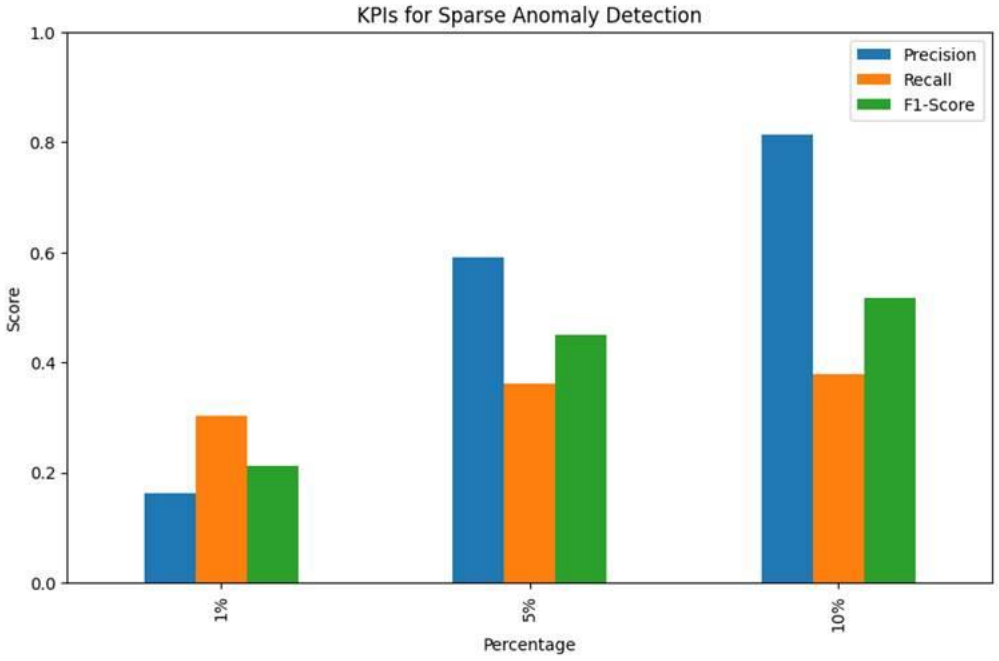


Figure 47: KPIs for sparse anomaly detection

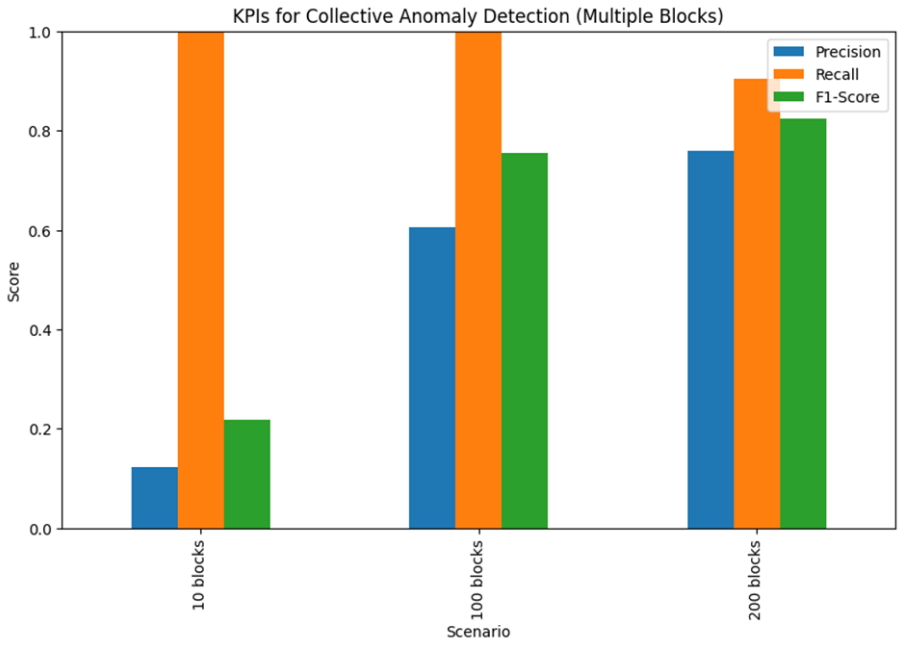


Figure 48: KPIs for collective anomaly detection

For the case of sparse anomalies detection in Figure 47, precision metric is shown to improve significantly as anomaly density increases (from 16% at 1% anomalies to 81% at 10%), meaning the model becomes more accurate when anomalies are frequent. Recall remains low across all percentages (30% at 1%, 36% at 5%, 38% at 10%), indicating the model struggles to detect many actual sparse anomalies. F1-Score grows with anomaly density but stays moderate, reflecting limited balance between precision and recall. As such, the model detects frequent anomalies well but fails to identify rare, isolated anomalies, suggesting the need for improved sensitivity to sparse faults.

For the case of collective anomalies detection in Figure 48, recall is perfect (100%) for 10 and 100 blocks and remains very high (90%) for 200 blocks, confirming that the model reliably detects collective anomalies affecting continuous time windows. Precision improves as the number of anomaly blocks increases (from 12% at 10 blocks to 76% at 200 blocks), meaning the model becomes more accurate in distinguishing real anomalies when they are more frequent. F1-Score shows significant growth (from 21% at 10 blocks to 82% at 200 blocks), reflecting a better balance between precision and recall as the collective anomalies become more evident. As such, the model is highly effective at detecting collective anomalies, especially when anomalies are widespread, demonstrating strong performance in identifying sustained system faults.

3.2.2 Evaluation of the condition monitoring service in terms of security

For the evaluation of secure and trustworthy vehicle condition monitoring, the SUPERCOM platform has been utilized to collect KPIs relevant to the proposed edge-based security framework, specifically in user story 1.2. As explained in Deliverable E10 [5], a general-purpose laptop emulates a vehicle's OBU, which generates periodic basic safety messages (BSMs) containing the vehicle's condition monitoring data. These multiple streams of BSMs, generated by the OBUs, are aggregated at the edge node. The edge node simulates a roadside unit within the vehicle infrastructure and serves to inject vehicle data streams into a pre-trained deep reinforcement learning (DRL)-based misbehavior detection model. The trained DRL model, developed in [6], is made available through the Kserve inference service, enabling real-time detection of security vulnerabilities in the vehicle's data.

For this evaluation, the open-source VeReMi dataset, which encompasses a wide range of misbehaviors associated with vehicular security vulnerabilities, has been utilized. Two common variants of attacks (i.e., data replay and DoS) have been selected to assess and report the relevant KPIs. Specifically:

- **Data replay attack**: A vehicle re-transmits or replays valid BSMs previously received from other vehicles. In this case, the vehicle uses its own ID while replaying the data and tries to exploit the conditions that existed at the time of the original BSM transmission. The attack could also be carried out in Sybil mode by changing the attacker ID.
- **Denial-of-service (DoS) attack**: A vehicle transmits BSMs at a frequency higher than the limit set by the standard. Such a high volume of data transmission would result in extensive periods of network congestion and unavailability to serve other legitimate vehicles. DoS attacks may also be launched by setting all BSM fields to random values (i.e., DoS random). Such behaviours can be concealed in a subtle way using compromised vehicles' identities (Sybil mode).

The security-related KPIs measured include accuracy, precision, recall, and F1-score. Additionally, upon detecting a security vulnerability, the mean time to detect (MTTD) is reported.

4 Key Performance Indicators

In the following, we report the achieved values for the measured KPIs during our experiments conducted at the two SUCCESS-6G testbeds.

4.1 Use Case 1: Vehicular condition monitoring and fault provisioning

4.1.1 User story 1.2: Vehicular condition monitoring with security guarantees

KPI	Relevant SUCCESS-6G enabler (based on E5)	SUCCESS-6G testbed (based on E10) Castelloli vs SUPERCOM	Achieved value
Number and frequency of failures	-End-to-end condition monitoring, failure identification, and visualization for V2X systems -Data analytics for informed decision-making regarding the vehicles condition status -C-V2X OBU	Castelloli	0
		SUPERCOM	Sparse anomalies (1%, 5%, 10% samples of the dataset) Collective anomalies (10, 100, 200 timesteps)
Downtime	-End-to-end condition monitoring, failure identification, and visualization for V2X systems (service downtime) -C-V2X OBU (device downtime)	Castelloli	Service downtime: Equivalent to service migration time (Section 4.3.1)
			Device downtime: 30-40s
Reliability	C-V2X OBU	Castelloli	100% during test
Accuracy	Trustworthy knowledge transfer at the edge in V2X systems Secure V2X edge intelligence with physics-informed learning	SUPERCOM	Data Replay attack: 0.9831 DoS attack: 0.9996
Precision	Trustworthy knowledge transfer at the edge in V2X systems Secure V2X edge intelligence with physics-informed learning	SUPERCOM	Data Replay attack: 0.9469 DoS attack: 0.9993

Recall	Trustworthy knowledge transfer at the edge in V2X systems Secure V2X edge intelligence with physics-informed learning	SUPERCOM	Data Replay attack: 1.0 DoS attack: 1.0
F1-score	Trustworthy knowledge transfer at the edge in V2X systems Secure V2X edge intelligence with physics-informed learning	SUPERCOM	Data Replay attack: 0.9725 DoS attack: 0.9996
Mean Time to Detect (MTTD)	Trustworthy knowledge transfer at the edge in V2X systems	SUPERCOM	Data Replay attack: 4.85 ms DoS attack: 4.72 ms

Table 10: Service KPIs for user story "vehicular condition monitoring with security guarantees"

4.2 5G network

4.2.1 Core

KPI	**SUCCESS-6G testbed (based on E10)** **Castelloli vs SUPERCOM**	**Achieved value**
Number of seconds this system has been running	Castelloli	Up Since: Mon Dec 16 05:20:10 2024
Max_attached users permitted	Castelloli	20
Max attached radios permitted	Castelloli	2
Number of attached Radios	Castelloli	2
Number of active radios (more than 1 user attached)	Castelloli	2
Number of attached Users	Castelloli	4
Number of Active Users (not idle mode)	Castelloli	4
Average CPU usage for PS	Castelloli	4%
Current UL bits per second on S1-U/N3	Castelloli	72,08 Mbps
Current DL bits per second on S1-U/N3	Castelloli	263,88 Mbps

Current UL bits per second on Sgi/N6 (Internet)	Castelloli	119,54 Mbps
Current DL bits per second on Sgi/N6 (Internet)	Castelloli	276,83Mbps
Network Latency	Castelloli	6ms. - 17 ms.
FTP Throughput	Castelloli	93,74 Mbps (Node 1) 51,93 Mbps (Node 8)

Table 11: Network KPIs that can be extracted from the core.

4.2.2 On-Board Unit

KPI	**SUCCESS-6G testbed (based on E10)** **Castelloli vs SUPERCOM**	**Achieved value**
Uptime	Castelloli	100% during testing
Latency	Castelloli	6ms. - 20 ms.
Bandwidth	Castelloli	5- 100 MHz Estimation: 25-30 MHz
DL (downlink) throughput **- Very good radio conditions** **- Good radio conditions** **- Medium radio conditions**	Castelloli	98.4 Mbps 94.8 Mbps 52.8 Mbps
UL (uplink) throughput **- Very good radio conditions** **- Good radio conditions** **- Medium radio conditions**	Castelloli	52.8 Mbps 43 Mbps 18.1 Mbps
Reliability	Castelloli	100%
Communication range	Castelloli	100%
RSRQ	Castelloli	-11, -12
RSRP	Castelloli	-77, -106
SNR	Castelloli	3.5 - 40

Table 12: Network KPIs that can be extracted from the final user (e2e)

4.3 Additional key performance indicators

Going beyond the KPIs defined in Deliverable E5 [4], a more comprehensive set of KPIs was measured

using the data collected during the testing activities in the Castelloli circuit.

KPI	SUCCESS-6G testbed (based on E10) Castelloli vs SUPERCOM	Achieved value
Inference success rate (%)	Castelloli	100%
Inference failure rate (%)	Castelloli	0%
Average inference time (seconds)	Castelloli	0.0151
Average HTTP request time (seconds)	Castelloli	0.0174
Request queue size	Castelloli	950
Handover Minimum response time (seconds)	Castelloli	30
Handover Average response time (seconds)	Castelloli	30
Handover Maximum response time (seconds)	Castelloli	30

Table 13: Additional KPIs

KPI	SUCCESS-6G testbed (based on E10) Castelloli vs SUPERCOM	Precision	Recall	F1-Score
Sparse (isolated anomalies) 1%	SUPERCOM using Castelloli measurements	0.162463	0.302402	0.211370
Sparse (isolated anomalies) 5%	SUPERCOM using Castelloli measurements	0.591519	0.362215	0.449302
Sparse (isolated anomalies) 10%	SUPERCOM using Castelloli measurements	0.813507	0.378711	0.516825

Collective anomalies 10 blocks	SUPERCOM using Castelloli measurements	0.122592	1.000000	0.218409
Collective anomalies 100 blocks	SUPERCOM using Castelloli measurements	0.605332	1.000000	0.754151
Collective anomalies 200 blocks	SUPERCOM using Castelloli measurements	0.758453	0.903506	0.824650

Table 14: Additional KPIs for anomaly detection

4.3.1 Service migration time measurements

This section presents an analysis of service migration time measurements from the Castellolí experiment conducted on March 7, 2025. The results provide insights into various aspects of service orchestration and handover (HO) performance, focusing on both overall success rates and the impact of the Mediator component on migration times. Through comparative evaluations, including success counts, average migration times, cumulative distribution functions (CDFs), and stacked bar plots, we examine key performance trends and potential bottlenecks in the service migration process. The findings highlight the trade-offs between increased orchestration control and additional overhead introduced by the Mediator, offering a comprehensive view of the system's efficiency and areas for optimization.

The plot in Figure 49 presents an overview, evaluating different aspects of service orchestration and HO performance. The x-axis represents four key categories: overall handover success from the DE perspective, successful high-level service orchestration, low-level orchestration issues, and successful low-level service orchestration. The y-axis displays the corresponding counts for each category, providing a comparative view of their occurrences during the experiment. The overall HO success from the DE perspective and successful high-level service orchestration both recorded a count of 57, indicating that high-level orchestration processes were generally reliable. However, low-level orchestration issues were observed 15 times, highlighting challenges in finer-grained service execution. Despite these issues, successful low-level service orchestration was recorded 42 times, showing that even though difficulties existed, the majority of low-level orchestration attempts were still successful. Within the successful low-level service orchestration category, additional results differentiate cases with and without the Mediator component. The experiment was conducted 25 times with the Mediator and 17 times without it, though this difference was due to time constraints rather than performance-related factors. The results overall provide key insights into the strengths and potential bottlenecks in the service orchestration framework.

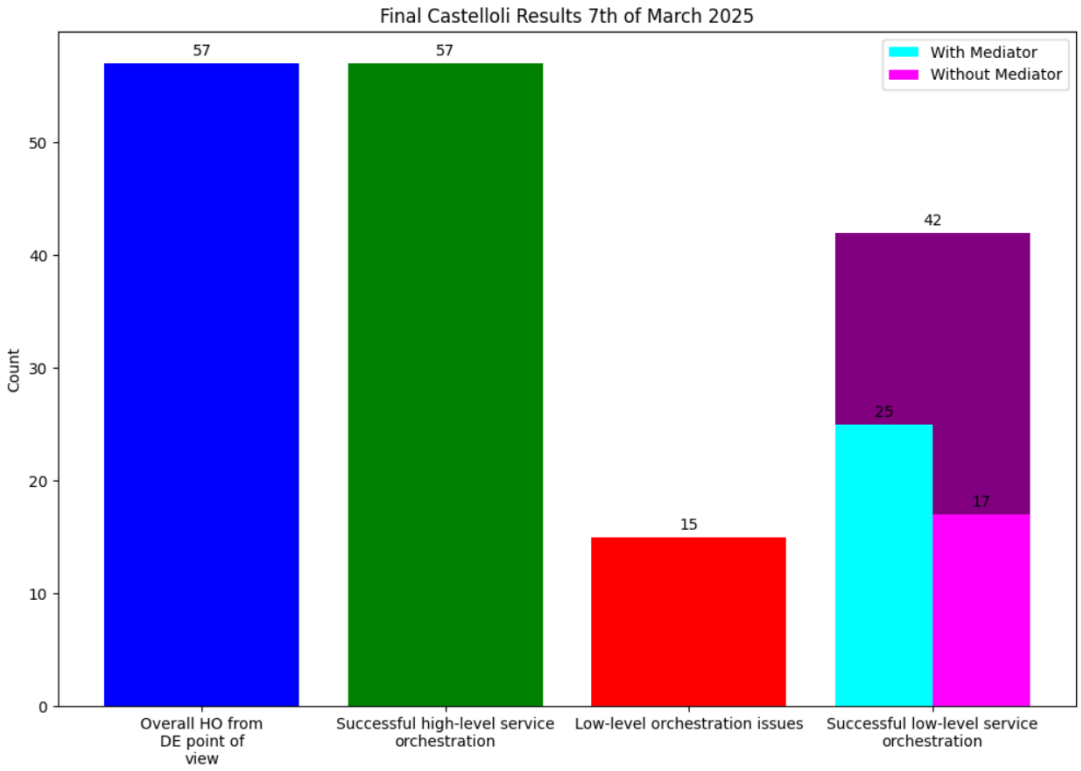


Figure 49: An overview of the final tests in the car circuit regarding service orchestration

In Figure 50, the plot presents the comparison of average migration times between scenarios with and without the Mediator component. The y-axis represents the migration time in seconds, while the x-axis differentiates the two experimental conditions. The error bars represent the standard deviation, providing insight into the variability of migration times in each case. The results show that the migration time with the Mediator has a higher mean value of 66.75 seconds, compared to 24.57 seconds without the Mediator. This significant difference highlights the additional overhead introduced when using the Mediator component. However, the higher standard deviation (10.92 seconds with Mediator vs. 3.38 seconds without Mediator) suggests that the migration process with the Mediator experiences more fluctuations in execution time. These results indicate that while the Mediator introduces additional latency, its role in orchestrating the migration process may involve more complex tasks, potentially contributing to improved service continuity or enhanced control over the process. Further analysis would be required to assess whether the added time leads to tangible benefits, such as reduced service disruptions or improved resource management, that justify the additional overhead.

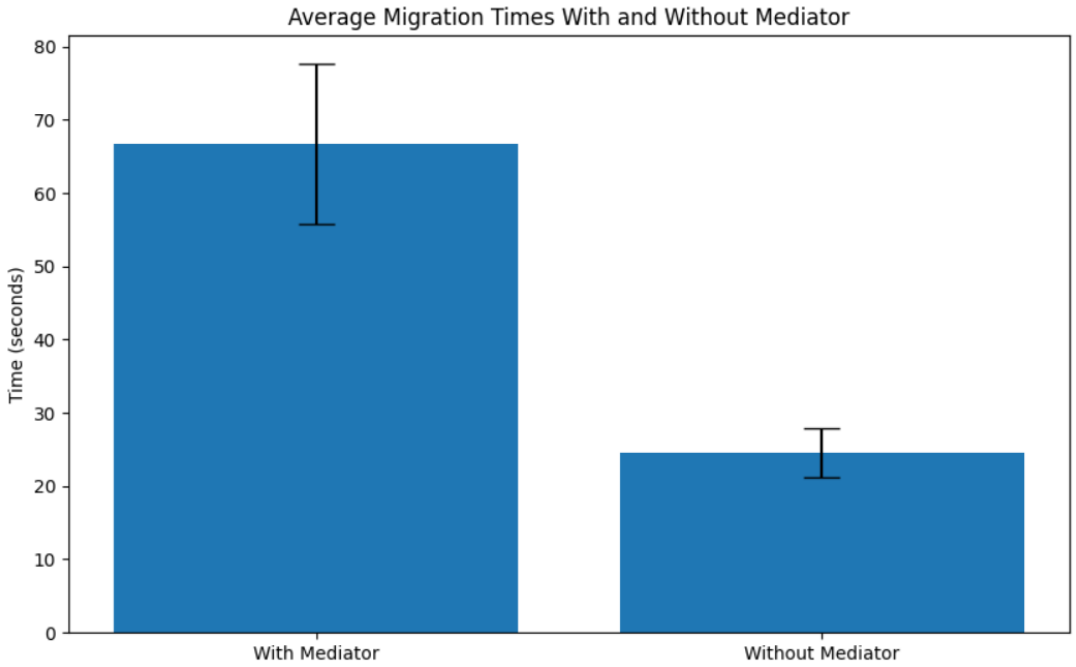


Figure 50: Comparison of service migration time with and without the *Mediator* service

In the following plots, we present finer-grained information, which illustrates the CDF with and without the Mediator service as well as the distribution of the service migration time duration.

The CDF plots in Figure 51 and Figure 52 visually represent the proportion of service migrations that complete within a given time. The x-axis represents migration times (in seconds), while the y-axis shows the cumulative probability, ranging from 0 to 1. The curve starts at zero and gradually increases as more migrations are completed. A steeper slope indicates a higher concentration of migrations occurring within that time range, while flatter regions suggest outliers or slower migrations.

Specifically, in the case of the Mediator, from the plot in Figure 51, we observe that the majority of service migrations are completed within a specific time range (60 to 70 seconds), with the curve rising quickly in that region. However, some migrations take significantly longer (around 90 seconds), as shown by the slower growth in the upper end of the distribution. This suggests that while most migrations follow a predictable pattern, a few cases experience higher delays. Understanding this distribution is valuable for performance optimization, as it allows us to set realistic expectations for migration times and identify areas for improvement.

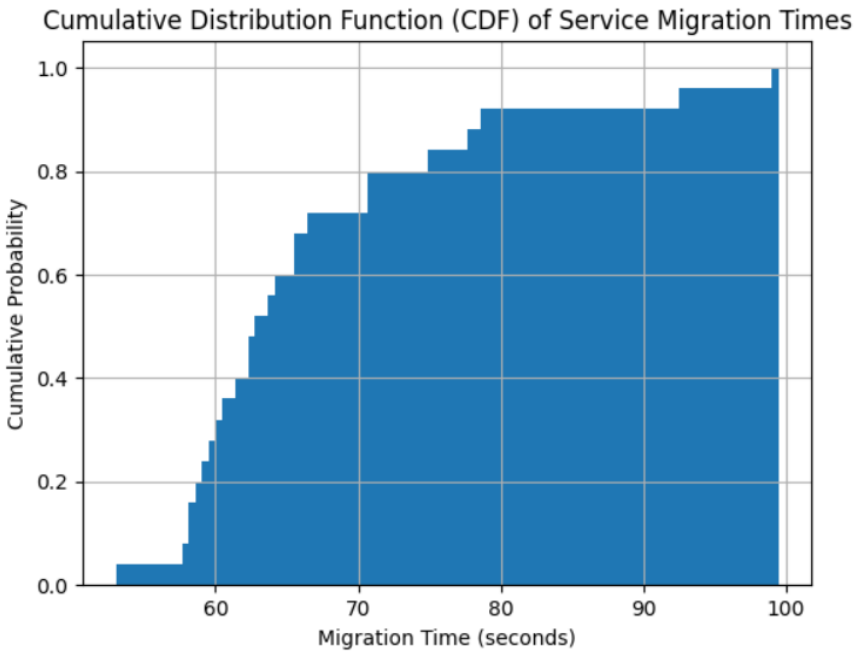


Figure 51: CDF of service migration times with the Mediator

On the other hand, without the Mediator, from the plot in Figure 52, we observe that around 50% (y = 0.5) of migrations are completed in approximately 4 to 5 seconds, meaning the median migration time is within this range. The curve rises steeply between 3 to 6 seconds, indicating that most migrations occur within this time frame. Beyond 6 seconds, the curve flattens, signifying that fewer migrations take longer. At y = 1.0 (or 100%), the plot reaches its highest point, meaning all migrations are complete, with the longest one taking slightly over 7 seconds. This distribution suggests that while most migrations happen efficiently within a predictable time frame, a small percentage take longer. Identifying the causes of these longer migrations can help optimize performance and reduce overall delays.

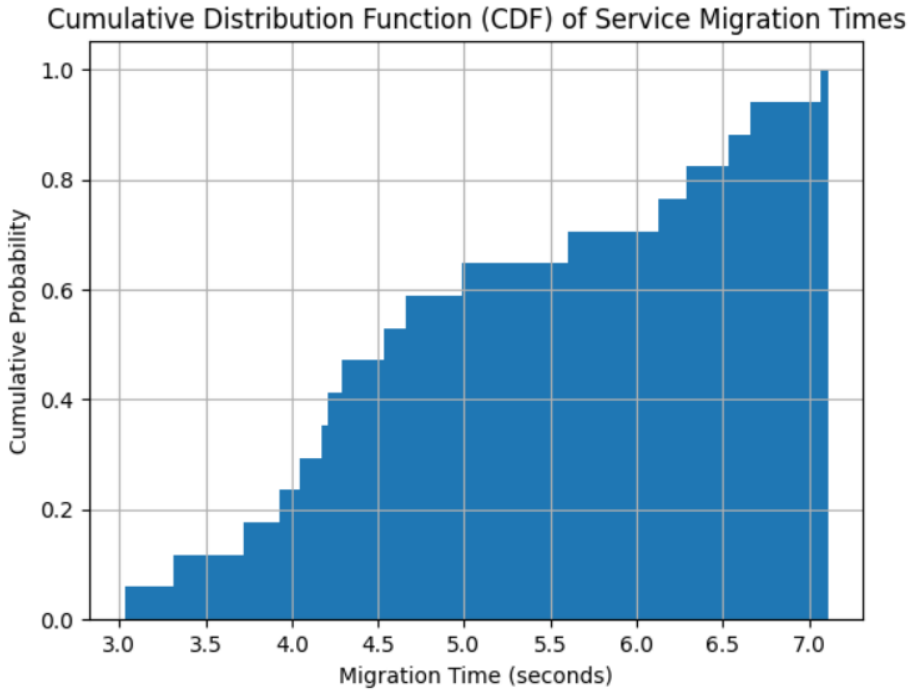


Figure 52: CDF of service migration times without the Mediator

Finally, the bar plot in Figure 53 illustrates the distribution of overall migration times without the Mediator, considering both the migration time of KServe custom K8s resources and the pod running time. In the stacked bar chart, the blue bars represent the migration times of Kserve without the mediator, while the orange bars represent the pod running time (the pod is generating on the background when the Kserve custom resource is running). The combined height of each bar represents the overall migration time, which includes both the migration time and the pod running time. The pod running times range between 15 to 25 seconds for each entry, adding variability to the overall migration time. The chart clearly shows that the pod creation time (orange segment) constitutes a significant portion of the overall time for each migration. It is evident that the pod running time generally adds a considerable amount to the overall migration time, with each stacked bar's height indicating how much extra time the pod creation adds to the process. This combination provides a clearer view of the total time it takes to complete a migration, including both the service migration and the time required for pod creation and initialization. This kind of stacked bar visualization is useful for understanding the separate contributions of migration time and pod creation time to the overall process, highlighting the relative importance of each in the migration duration.

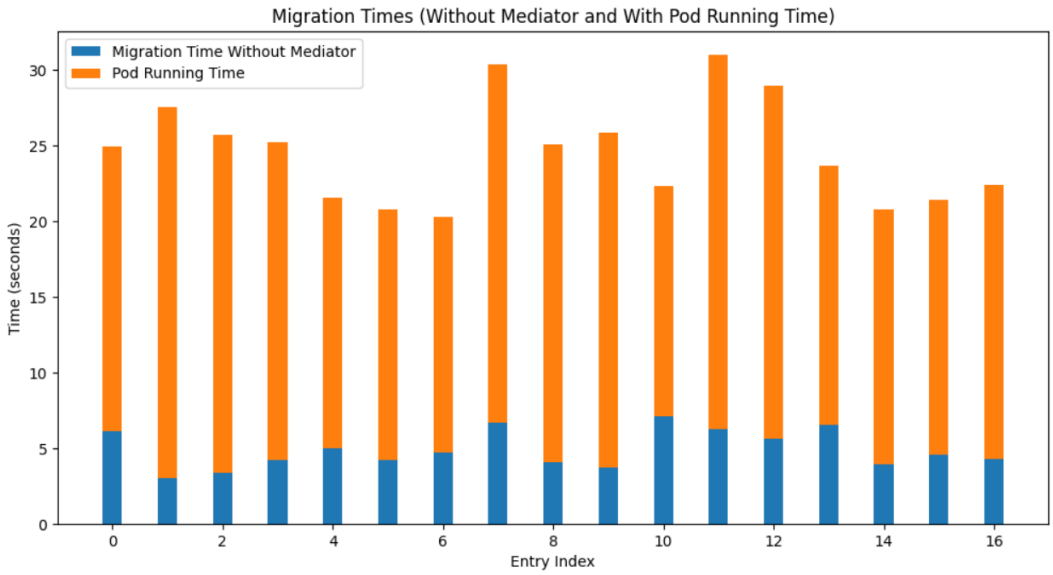


Figure 53: Migration times without the Mediator, but also including the pod creating times

4.4 Unmeasured key performance indicators

Besides the measured indicators, a small subset of the KPIs defined in E5 were not eventually tracked. We provide the details as follows:

- Number of paging failures: The metric was not measured as it was deemed irrelevant to the specific context of the use case.
- Packet loss ratio: Although not directly measured within the radio environment of the testing infrastructure, this metric can be indirectly associated with the number of (synthetic) failures introduced.
- Occlusion length: This metric was not recorded, as missing values in the dataset were not treated as anomalies. Instead, specific anomalous values were intentionally injected (as shown in Table 9) to emulate fault conditions.
- Unauthorized access: This metric was not tracked, given that secure end-to-end connectivity was assumed throughout the testing environment.

- Remaining useful lifetime: The metric was not measured because the introduction of synthetic anomalies to simulate faults made it impractical to accurately assess the actual lifespan of real equipment.
- Maintenance response time: The metric was not evaluated since there was no closed-loop functionality implemented upon the detection of the (synthetically introduced) anomalies.
- Mean time to resolve: For similar reasons as above, the metric was not assessed since there was no closed-loop mechanism implemented upon the detection of security incidents.
- Energy consumption: Although not directly measured, this metric can be inferred based on the training time of the classification models deployed at the edge and used in the evaluation.

5 Summary of experimental validation and insights

A series of real-world tests were carried out at the Cellnex Mobility Lab at Circuit Parcmotor Castelloli, with the aim of evaluating the performance and operational robustness of the vehicular condition monitoring service developed under the SUCCESS-6G-DEVISE project. These tests were crucial in assessing not only the end-to-end functionality of the service but also the seamless integration and interoperability of its enabling technological components.

The experimental validation focused on replicating realistic vehicular scenarios, including handovers across the edge monitoring infrastructure, varying vehicular speeds and network conditions, to rigorously challenge the system's capabilities in a controlled but dynamic environment. Key elements of the condition monitoring service—such as edge-based data processing, service orchestration, communication interfaces, and advanced analytics—were assessed in terms of their responsiveness, reliability, and overall impact on service quality.

Throughout the testing campaign, extensive data were collected on performance metrics aligned with the defined KPIs, with emphasis on handover information to ensure service resilience during vehicular movement. These metrics provided a quantitative basis for evaluating the system’s real-time behavior and its ability to maintain service continuity under demanding conditions. The insights gained from this validation exercise not only confirmed the feasibility of the solution in a near-deployment environment but also helped identify opportunities for fine-tuning specific components, such as optimizing the data ingestion pipeline, improving predictive maintenance algorithms, or enhancing V2X link reliability.

Overall, the testing activities served as a vital step toward demonstrating the readiness of the condition monitoring service for further integration into broader intelligent transportation systems, paving the way for scalable deployments in 6G-enabled vehicular networks.

6 Annex A

6.1 Photos from the Castelloli trials on 19/12/2024

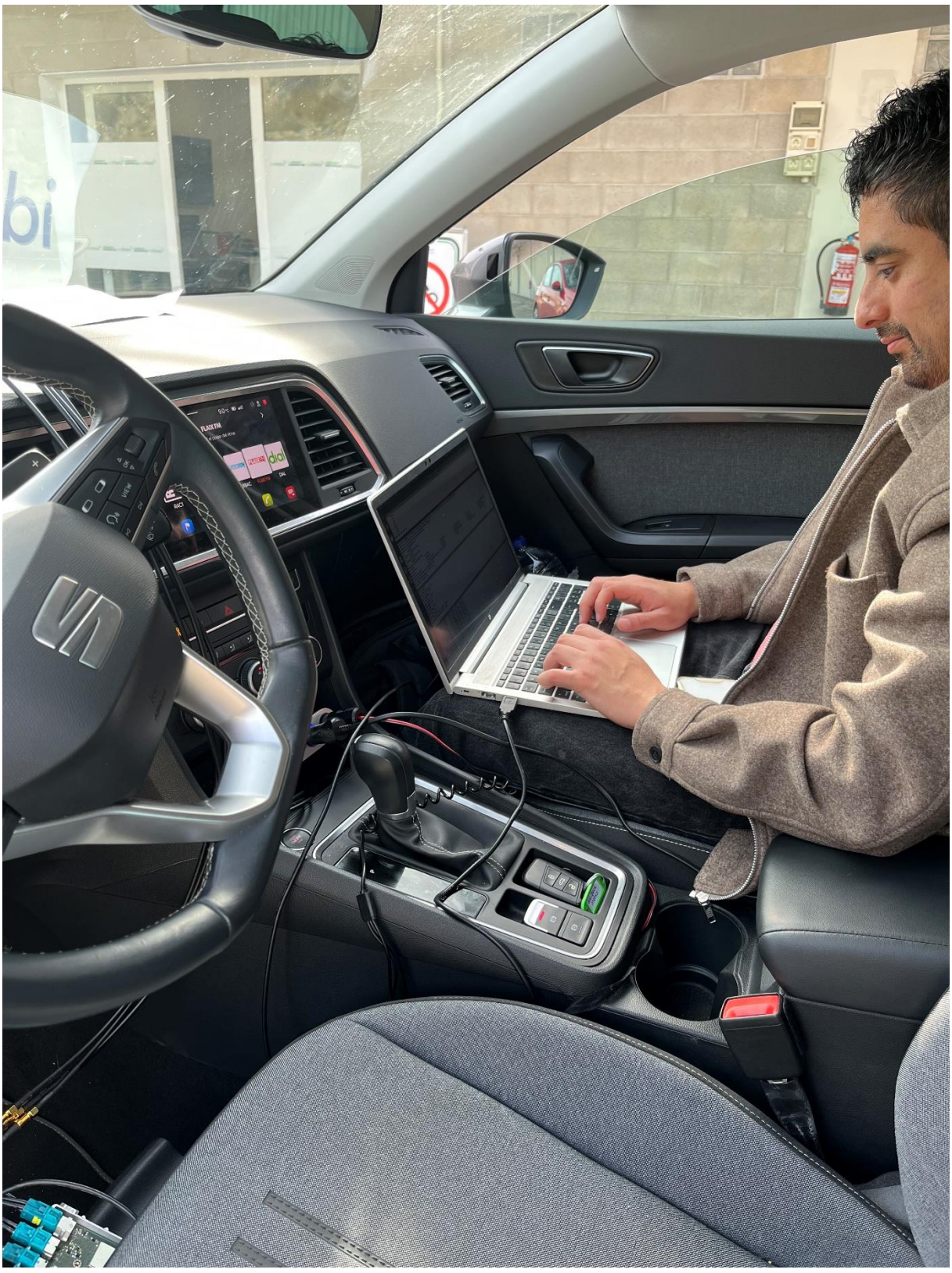

3109 MCZ
Ateca


idnes


3109 MCZ

6.2 Photos from the Castelloli trials on 24/02/2025


3109 MCZ



6.3 Videos from the Castelloli trials on 07/03/2025

Handover from node 1 to node 8 (click on the image to play)

Handover from node 8 to node 1 (click on the image to play)

References

[1] P. Mulinka et al., "Information Processing and Data Visualization in Networked Industrial Systems," 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 2021, pp. 1-6, doi: 10.1109/PIMRC50174.2021.9569603.

[2] P. Mulinka, S. Sahoo, C. Kalalas and P. H. J. Nardelli, "Optimizing a Digital Twin for Fault Diagnosis in Grid Connected Inverters - A Bayesian Approach," 2022 IEEE Energy Conversion Congress and Exposition (ECCE), Detroit, MI, USA, 2022, pp. 1-6, doi: 10.1109/ECCE50734.2022.9947986.

[3] A. Beattie et al., "A Robust and Explainable Data-Driven Anomaly Detection Approach For Power Electronics," 2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Singapore, Singapore, 2022, pp. 296-301, doi: 10.1109/SmartGridComm52983.2022.9961002.

[4] Deliverable E5: https://success-6g-project.cttc.es/images/deliverables/E5-DEVISE-2.pdf

[5] Deliverable E10: https://success-6g-project.cttc.es/images/deliverables/E10-DEVISE.pdf

[6] R. Asensio-Garriga et al., "ZSM-Based E2E Security Slice Management for DDoS Attack Protection in MEC-Enabled V2X Environments," in IEEE Open Journal of Vehicular Technology, vol. 5, pp. 485-495, 2024, doi: 10.1109/OJVT.2024.3375448.

[7] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, “Tabtransformer: Tabular data modeling using contextual embeddings,” arXiv preprint arXiv:2012.06678, 2020.

[8] S. O. Arık and T. Pfister, “TabNet: Attentive Interpretable Tabular Learning,” arXiv preprint arXiv:1908.07442, 2020.

[9] T. Akiba et al., “Optuna: A Next-generation Hyperparameter Optimization Framework,” in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019

[10] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[11] C. Kalalas, P. Mulinka, G. Candela Belmonte, M. Fornell, M. Dalgitsis, F. Paredes Vera, J. Santaella Sánchez, C. Vicente Villares, R. Sedar, E. Datsika, A. Antonopoulos, A. Fernández Ojea, M. Payaro, ``AI-Driven Vehicle Condition Monitoring with Cell-Aware Edge Service Migration," arXiv preprint, arXiv:2506.02785, June 2025, https://arxiv.org/abs/2506.02785