

SAFE-6G

A Smart and Adaptive Framework for Enhancing Trust in 6G Networks

Deliverable D4.7: User-centric Reliability Function

Date: 30/09/2026

Version: V1.0

DISCLAIMER

This document contains information, which is proprietary to the SAFE-6G Consortium (“A Smart and Adaptive Framework for Enhancing Trust in 6G Networks”) Consortium that is subject to the rights and obligations and to the terms and conditions applicable to the Grant Agreement number: 101139031. The action of the SAFE-6G Consortium is funded by the European Commission.

Neither this document nor the information contained herein shall be used, copied, duplicated, reproduced, modified, or communicated by any means to any third party, in whole or in parts, except with prior written consent of the SAFE-6G Consortium. In such case, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced. In the event of infringement, the consortium reserves the right to take any legal action it deems appropriate.

This document reflects only the authors’ view and does not necessarily reflect the view of the European Commission. Neither the SAFE-6G Consortium as a whole, nor a certain party of the SAFE-6G Consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is accurate or free from risk, and accepts no liability for loss or damage suffered by any person using this information.

The 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.

Grant Agreement	101139031
Document number	D4.7
Document title	User-centric Reliability Function
Lead Beneficiary	UNIWA
Editor(s)	Panagiotis Karkazis (UNIWA)
Author(s)	Panagiotis Karkazis (UNIWA) Dimitrios Uzunidis (UNIWA) Christos Betzelos (UNIWA) Stamatia Drampalou (UNIWA) Anastasios Vetsos (UNIWA) Emmanouil Atsilimis (UNIWA) Christina Charitou (UNIWA) Nelly Leligou (UNIWA) Nikolaos Miridakis (UNIWA)
Dissemination level	Public
Contractual date of delivery	30/09/2026
Status	Final
File name	SAFE-6G_D4.7_V1.0.pdf

Revision History

Version	
V0.1	Initial ToC
V0.2	First Contribution of the UNIWA team
V0.3	Online contributions (second round), with iterative editor feedback.
V0.4	Online contributions (third round), with iterative editor feedback.
V0.5	First review comments performed by IQBT and UPV
V0.7	Final draft after the comments from the second review performed by the Technical Steering Committee. Final review performed by the Editor.
V1.0	Final version following the Quality check.

GLOSSARY

Abbreviations/Acronym	Description
5G	5th Generation of Mobile Networks
6G	6th Generation of Mobile Networks
AI	Artificial Intelligence
API	Application Programming Interface
CISM	Container Infrastructure Service Management
cLoTw	calibrated Level of Trustworthiness
CoCo	Cognitive Coordinator
CPU	Central Processing Unit
CQI	Channel Quality Indicator
DNN	Deep Neural Network
DT	Decision Tree Classifier
eMBB	Enhanced Mobile Broadband
FL	Federated Learning
gNB	Next Generation Node B
GPU	Graphics Processing Unit
HPC	High-Performance Computing
IPFIX	IP Flow Information Export
k-NN	k-Nearest Neighbors
KPI	Key Performance Indicator
LLM	Large Language Model
LoTw	Level of Trustworthiness
ML	Machine Learning
MLOps	Machine Learning Operations
nApp	Network Application
NFVI	Network Functions Virtualization Infrastructure
nLoTw	Non-calibrated Level of Trustworthiness
OOD	Out-of-distribution
PRC	Precision-Recall
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RF	Random Forest
RL	Reinforcement Learning
ROC	Receiver-Operating-Characteristic
RTSP	Real-Time Streaming Protocol
SINR	Signal-to-Interference-plus-Noise Ratio
SVC	Support Vector Classifier
TF	Trust Function
UC	Use Case
UE	User Equipment
URLLC	Ultra-Reliable and Low-Latency Communications
vApp	Vertical Application
VIM	Virtualised Infrastructure Manager
VNF	Virtualised Network Function
VR	Virtual Reality

WP
XR

Work Package
Extended Reality

EXECUTIVE SUMMARY

This deliverable presents the design, implementation and evaluation of the Reliability Trust Function (TF) within the SAFE-6G cognitive trustworthiness framework. The Reliability TF ensures predictable, stable and user-aligned service behaviour across the edge-cloud continuum, with particular emphasis on QoE-critical applications such as real-time video streaming and immersive media.

The work carried out in this deliverable focuses on identifying reliability threats, analysing performance-degradation causes, and developing AI-driven mechanisms for proactive reliability assurance. The Reliability TF leverages a multi-layer monitoring framework that collects telemetry from physical servers, virtualisation layers and application components. This telemetry enables the TF to detect anomalies, anticipate reliability degradation and enforce corrective actions aligned with user-centric trust requirements.

A range of Artificial Intelligence (AI)/ Machine Learning (ML) models has been developed and evaluated, including lightweight classifiers, deep neural networks and transformer-based approaches. These models support QoE prediction, multi-layer anomaly detection and out-of-distribution (OOD) attack identification. Reinforcement Learning (RL) and Federated Learning (FL) were initially explored as potential techniques for distributed reliability optimisation; however, analysis of SAFE-6G requirements and experimental findings indicated that these methods are not optimal for low-latency inference and multi-layer telemetry fusion. As a result, the Reliability TF adopts alternative AI approaches that better satisfy the project's objectives.

The scientific work conducted in Task 4.6 has led to **six peer-reviewed publications** [1], [2], [3], [4], [5], [6] and the creation of **two openly accessible multi-layer datasets** [7], [8], which support proactive scaling, reliability inference and transformer-assisted attack detection. These datasets complement external resources used for validation, such as Simu5G QoE traces [9] and the 5G-NIDD OOD dataset [10].

Overall, the results demonstrate that the Reliability TF provides an effective, user-centric mechanism for assessing and enforcing reliability in distributed 6G environments. The TF integrates seamlessly with the SAFE-6G architecture and contributes essential capabilities to the cognitive trustworthiness framework. Its integration and end-to-end validation will be carried out in subsequent WP5 activities.

KEYWORDS

User-centric approach, Reliability Trust Function, Quality of Experience (QoE), Multi-layer Monitoring, Artificial Intelligence, Transformer Models, Level of Trustworthiness (LoTw), Edge-Cloud Continuum, Video Streaming, Anomaly Detection

TABLE OF CONTENTS

1	<i>Introduction</i>	1
1.1	The rationale behind the structure	1
2	<i>Related Work</i>	3
2.1	Openly accessible datasets	5
2.2	Standardization efforts	6
3	<i>Role of the Trust Function in SAFE-6G</i>	8
4	<i>Reference Architecture</i>	10
4.1	Requirements	12
4.2	Architectural considerations of the user-centric reliability function	13
5	<i>Trust Function Layers</i>	14
5.1	Main functionalities and ml models according to the LoTw	14
5.1.1	Low LoTw (1–40 %)	14
5.1.2	Medium LoTw (41–80 %)	15
5.1.3	High LoTw (81–100 %)	15
5.2	Sequence diagram	16
5.3	Local AI Agent	18
5.3.1	Technical Responsibilities	18
5.4	Vertical Applications (vApps)	19
5.4.1	LoTw flavors	19
5.4.2	Technical Responsibilities	22
5.5	Network Application (nApp)	22
5.5.1	Technical Responsibilities	22
6	<i>Trust Evaluation & Enforcement Logic</i>	25
6.1	Impact of user characteristics	25
6.2	Validation exploiting an openly accessible dataset	26
6.2.1	Simu5G-based QoE dataset	26
6.2.2	5G-NIDD dataset for Out-of-Distribution (OOD) attack detection	29
6.3	Validation exploiting SAFE-6G dataset	35
6.3.1	Multi-layer monitoring and proactive scaling dataset	36
6.3.2	Transformer-Assisted Multi-Layer Attack Detection Dataset	41
7	<i>Future Research Pathways</i>	44
8	<i>Conclusions</i>	45
9	<i>REFERENCES</i>	46

List of FIGURES

Figure 1: High-level overview of the Reliability TF.....	10
Figure 2: Low Flavour	14
Figure 3: Medium Flavour.....	15
Figure 4: High Flavour	16
Figure 5: Sequence Diagram of the Reliability TF	17
Figure 6: RF inference logs / feature scaling output.....	20
Figure 7: DNN inference output.....	20
Figure 8: Example inference logs of the High LoTw models on multi-layer telemetry records.....	21
Figure 9: Correlation coefficients between “frames displayed” and the four QoS metrics across the 15 UEs [3]	28
Figure 10: Representative session showing the temporal evolution of frames displayed, CQI, SINR and end-to-end delay [3]	28
Figure 11: Euclidean distances between attack classes and the benign class (class-similarity matrix) [1]	30
Figure 12: LightGBM model pipeline [1].....	32
Figure 13: Untuned sentence transformer embeddings with RF model pipeline [1].....	32
Figure 14: Fine-Tuned LLM with classifier head model pipeline [1]	33
Figure 15: Fine-Tuned LLM embeddings with RF classifier [1]	33
Figure 16: Average malicious recall across LLM-based models [1].....	34
Figure 17: Malicious recall per attack type, per model and per input size [1]	34
Figure 18: Average accuracy per attack type, per model and per input size [1]	35
Figure 19: MediaMTX architecture [6].....	36
Figure 20: Bar chart comparing accuracy, precision, recall, and F1 score per model [4].....	38
Figure 21: (a) ROC and (b) Precision-Recall curves of the DNN, aggregated across all six scenarios [4]	39
Figure 22: Per-scenario plots of predicted versus actual reliability labels together with average memory-usage ratio (a) One streamer with 100 MiB memory constraints. (b) One streamer with 200 MiB memory constraints. (c) Two streamers with 100 MiB memory constraints. (d) Two streamers with 200 MiB memory constraints. (e) Three streamers with 100 MiB memory constraints. (f) Three streamers with 200 MiB memory constraints [4].....	40
Figure 23: MediaMTX architecture for attack detection [2].....	41
Figure 24: MediaMTX architecture for multi-layer attack detection [2]	42

List of TABLES

Table 1: Reliability TF Requirements	13
Table 2: Multi-Layer Telemetry Metrics	23
Table 3: Number of samples for each class [1]	29
Table 4: Class distribution across the six validation scenarios. Values are expressed as the number of samples, with percentages reported in parentheses [4]	37
Table 5: Optimal hyperparameters for each ML model and scenario [4]	38
Table 6: Model Performance Metrics Over Nine Random Seeds (Mean $\pm$ Standard Deviation) [2]	43

1 INTRODUCTION

The SAFE-6G project adopts a holistic and user-centric approach to ensure trustworthiness in sixth-generation (6G) networks. As communication systems evolve toward intent-driven, Artificial Intelligence (AI)-native, and highly decentralised architectures, trust must be continuously evaluated and dynamically adapted to user needs. Within this broader vision, the Reliability Trust Function (TF) ensures that services delivered across the edge-cloud continuum maintain predictable, stable and user-aligned behaviour under diverse operational conditions, with particular emphasis on satisfying the Quality of Experience (QoE) offered to the users during applications such as real-time video streaming and immersive ones.

This deliverable is developed within **Work Package 4 (WP4)**, and specifically **Task 4.6**, which focuses on reliability threats, performance degradation causes, and AI-driven mechanisms for proactive and adaptive reliability control. WP4 defines the operational and functional behaviour of the TFs, while WP3 provides the underlying technical foundation, including multi-layer monitoring, Machine Learning Operations (MLOps) mechanisms, and the infrastructure components that generate the telemetry consumed by the Reliability TF. The design of the Reliability TF took into account all functional and non-functional requirements specified in Deliverable 2.1. Subsection 4.1 provides further details on these requirements, how they are addressed by the Reliability TF, and how their implementation is reflected in the current deliverable. The integration of the Reliability TF into the SAFE-6G platform will be carried out in WP5, where the cognitive loop and cross-TF orchestration are validated end-to-end.

In accordance with the project objectives, this deliverable:

- defines the role of the Reliability TF within the SAFE-6G cognitive trustworthiness framework,
- presents its architecture, layers and operational logic,
- describes the AI/ Machine Learning (ML) models used for reliability inference and anomaly/ out-of-distribution (OOD) detection,
- details the multi-layer monitoring and datasets used for training and validation,
- and reports the evaluation results that demonstrate the TF's effectiveness in user-centric reliability assurance.

1.1 THE RATIONALE BEHIND THE STRUCTURE

The structure of this deliverable follows a logical progression that reflects the development, specification and evaluation of the Reliability TF within the SAFE-6G framework. After the introduction, the document is organised as follows:

- **Section 2 - Related Work:** Provides an overview of the scientific background relevant to user-centric reliability, QoE-driven optimisation, multi-layer monitoring and AI-native

management in 6G systems. It also summarises external datasets and standardisation efforts that inform the design of the Reliability TF.

- **Section 0 - Role of the TF in SAFE-6G:** Describes how the Reliability TF fits into the SAFE-6G cognitive trustworthiness framework and its interactions with other TFs and the Cognitive Coordinator (CoCo).
- **Section 0 - Reference Architecture:** Presents the architectural principles, requirements and design considerations that guide the implementation of the Reliability TF across the edge-cloud continuum.
- **Section 0 - TF Layers:** Details the three layers of the TF (Local AI Agent, Vertical Application (vApp), Network Application (nApp)), their responsibilities, Level of Trustworthiness (LoTw)-driven flavour selection, and the AI/ML models executed within each layer.
- **Section 0 - Trust Evaluation & Enforcement Logic:** Explains how reliability is evaluated, how user characteristics influence reliability decisions, and how enforcement actions are triggered. Includes validation using external datasets and SAFE-6G multi-layer telemetry.
- **Section 7- Future Research Pathways:** Presents forward-looking research directions that naturally emerge from the scientific results of the deliverable and extend beyond the scope of SAFE-6G.
- **Section 0 – Conclusions:** Summarises the main outcomes and contributions of the Reliability TF and outlines its readiness for integration into the SAFE-6G platform.

2 RELATED WORK

The reliability of 6G wireless systems has become the center of attention for a large number of journal and conference publications due to the increasing demand for Ultra-Reliable and Low-Latency Communications (URLLC) to support mission-critical applications. The vision and the key requirements for 6G systems are described in [3], [5], [11], [12], [13], [14], where the need for this type of communication is emphasised. The authors of these works discuss the challenges in achieving these goals and provide insights into various technological advancements, such as AI/ML tools, that can contribute to the design of reliable 6G networks [3].

The evolution toward user-centric and AI-native 6G networks is reshaping how reliability is conceptualised, measured, and enforced. As traditional network-centric reliability indicators, such as packet loss, throughput, or link stability, prove insufficient for emerging 6G services like immersive video streaming, Extended Reality (XR), and real-time media pipelines operating across heterogeneous and dynamic edge-cloud environments, reliability must instead be interpreted through the lens of QoE, service continuity, and personalised performance expectations. Current research reinforces this shift toward a more user-centric network architecture tailored to individual user requirements, contrasting with the service-oriented approach of 5G, where the primary focus lies in guaranteeing Quality of Service (QoS) for predefined service categories, independent of the specific users consuming each service [5]. This distinction highlights a critical gap in existing reliability approaches, which focus primarily on network-centric metrics and fail to capture the subjective and contextual dimensions of the user experience [4].

The vision for a user-centric approach is relatively new to the research community. Only a few research works have been published related to this approach, including [15], [16], [17], [18], [19], [20], [21], where the shift from network-function focus to user focus is analysed. Specifically, in [15] an AI-native user-centric network architecture is proposed that exploits ML algorithms to optimise network resource allocation and management of user services. The authors in [16] discuss future developments in native AI within the User-Centric Access Network architecture. Further, the authors of [17], [18], [19] propose changes that should be made to current network management and orchestration to migrate to a user-centric approach, while [20], [21] present the contribution of AI technologies such as Federated Learning (FL) towards trusted, intelligent, zero-touch service management. Overall, most existing studies mainly focus on the core network or specific segments (edge or cloud) without considering the whole continuum and the specific user characteristics. Therefore, a comprehensive examination that integrates the entire 5G/6G core and edge-cloud continuum, considering a user-centric reliability approach, still remains underexplored [3].

Reinforcement Learning (RL) and FL have been widely discussed in the literature as enablers of intelligent, decentralised and user-centric 6G management. RL-based approaches aim to optimise resource allocation and service reliability through continuous interaction with the environment, enabling adaptive decision-making under dynamic network conditions [22], [23], [24]. FL contributes to privacy-preserving collaborative learning across distributed edge nodes, allowing models to be trained without sharing raw user data [23], [25], [26], [27], [28], [29], a key requirement for

user-centric trustworthiness. As highlighted in [5], these techniques form part of the broader AI-native vision for 6G networks, supporting zero-touch automation and personalised service optimization [23], [25], [30], [31], [32]. However, existing studies also reveal practical limitations: RL often suffers from high training overhead [22], [23] and instability in non-stationary environments [24], while FL introduces communication overhead [28], [29] and model-drift challenges [27] in heterogeneous edge deployments [23]. These constraints motivate the development and training of alternative AI/ML methods in the Reliability TF, where low-latency inference and multi-layer telemetry fusion are essential.

External reliability threats have been also documented in recent studies. Inter-operator interference mediated through Reconfigurable Intelligent Surface (RIS) has been shown to degrade Signal-to-Interference-plus-Noise Ratio (SINR), Channel Quality Indicator (CQI), and ultimately QoE in URLLC and multimedia services, reinforcing the need for reliability-aware orchestration in multi-tenant environments [33], [34]. Although these works do not constitute reliability mechanisms themselves, they provide essential context for understanding the conditions under which a Reliability TF must operate.

Building on the identified gaps in the literature, the work carried out within Task 4.6 has produced a coherent set of scientific contributions that directly address the requirements of a user-centric Reliability TF. These contributions, developed by the UNIWA team in the context of the present task, cover multi-layer monitoring, QoE-driven reliability inference, proactive scaling, and advanced anomaly/OOD detection, and are fully aligned with the objectives stated in the project.

Specifically, the following peer-reviewed publications and openly accessible datasets have been produced:

1. A comprehensive survey that systematically analyses the shift from network-centric to user-centric reliability in 6G and identifies the missing building blocks for personalised, QoE-aware reliability control [5].
2. A user-driven AI approach that jointly profiles service characteristics, user behaviour and network conditions to support personalised reliability decisions across the edge-cloud continuum [3].
3. An AI-driven reliability mechanism for real-world video streaming that combines multi-layer monitoring across the edge-cloud continuum, application-layer metrics and 5G-core performance data, with classical ML and DNNs to predict QoE degradation and trigger proactive actions [4].
4. An open-source framework based on tools such as Prometheus, Thanos and Kepler collects telemetry from the application layer and the containerised infrastructure. The associated labelled dataset (17 metrics across multiple memory allocations and streamer scenarios) [7] enables proactive scaling decisions and the prediction of system breakpoints that cause QoS/QoE violations. A novel indicator, the Time Till First Major Failure (TTFMF), has been introduced to evaluate the framework [6].
5. OOD attack detection using LLMs has been shown to improve the identification of previously unseen malicious behaviours. By converting network-traffic samples into textual feature-value representations and employing a leave-one-attack-out evaluation protocol on the 5G-NIDD

dataset [10], LLM-based methods achieve up to 5% improvement in malicious recall over a non-LLM baseline (LightGBM) [1].

6. A transformer-assisted multi-layer attack detection for cloud-native video streaming fuses network-flow (IP Flow Information Export - IPFIX), application and continuum telemetry. The combined records are serialised into deterministic feature-value sequences and processed with a fine-tuned DistilBERT encoder. Experimental results demonstrate that transformer-assisted models consistently outperform a tabular Random Forest (RF) baseline, improving attack-class recall by 6.15% while maintaining near-perfect precision [2].

Collectively, these results demonstrate that the Reliability TF developed in Task 4.6 is not an ad-hoc collection of algorithms, but the concrete realisation of a scientifically grounded, user-centric reliability approach that fills the gaps identified in the state of the art and remains fully consistent with the low-TRL research objectives of the project.

2.1 OPENLY ACCESSIBLE DATASETS

A number of datasets that can be used to train ML methods to ensure reliability and QoE in 5G/6G networks have been published in open-access repositories [9], [35], [36]. Datasets relating key performance metrics, such as packet loss rate, packet delay, with application type, category and slice type support ML-based network-slice prediction even under failure conditions [35], [37]. Large-scale YouTube QoE traces collected over commercial 4G/5G networks associate stall events, quality shifts and channel-level indicators with Key Performance Indicators (KPIs) [38]. End-to-end emulation datasets inject anonymised commercial cellular traffic over emulated radio-access and 5G-core platforms, enabling analysis of novel network-automation architectures [39]. Further, a Simu5G-based dataset includes 14 metrics from 15 User Equipments (UEs) across 24 sessions and, importantly, user-unique metrics such as position and velocity that directly affect CQI, SINR and end-to-end delay, which in turn influence QoE (e.g., percentage of video frames arriving on time) [9]. Another dataset relates Central Processing Unit (CPU) usage and traffic details (number of active UEs, Next Generation Node B (gNB) downlink bit rate, outbound traffic) for a 5G Real-Time Streaming Protocol (RTSP) video-streaming use case [36]. Complementary work has highlighted the need for collecting data from all architectural layers (application, virtual, physical) in order to ensure reliability while optimising resource utilisation [40], [41].

More recent openly available resources further expand the design space for QoE prediction, reliability assessment, multi-layer monitoring and XR/video-streaming studies. An XR Traffic-QoE dataset provides network traffic measurements together with perceptual QoE annotations spanning Virtual, Augmented and Mixed Reality applications under controlled bandwidth and refresh-rate conditions [42]. The 5G-QoERA dataset integrates real user mobility traces, radio-map information, scheduling decisions and application-level details for video-streaming QoE assessment in 5G New Radio [43]. Additional datasets address beyond-Enhanced Mobile Broadband (eMBB) QoS/QoE performance in real 5G networks [44], and end-to-end 360-degree video service measurements collected over mobile networks [45].

While these datasets are not directly used for the primary SAFE-6G validation (which relies on project-generated multi-layer telemetry), they form a valuable reference base for designing, training and benchmarking the Reliability TF under diverse operational and application conditions.

In addition to the external resources summarised above, two openly accessible multi-layer datasets were developed and released. These datasets constitute a core scientific outcome of the Reliability TF and directly support the training and validation of the AI/ML models described in later sections.

The first dataset [7] was collected from a MediaMTX-based video-streaming testbed instrumented with Prometheus, Thanos and Kepler. It contains 17 labelled metrics spanning application-layer (RTSP sessions, bytes sent/received), container/virtualisation and physical infrastructure (CPU, memory, energy) under varying memory allocations and streaming workloads. The dataset is publicly available on Zenodo and was used to train and evaluate the RF and DNN models for proactive scaling and QoE-degradation prediction [6].

The second dataset [8] focuses on transformer-assisted multi-layer attack detection. It fuses network-flow (IPFIX), application-layer and continuum telemetry collected under normal operation, high-load benign conditions and four controlled attack scenarios (SYN flooding, RTSP session flooding, slow RTSP and brute-force authentication). The combined records are serialised into deterministic feature-value sequences suitable for DistilBERT-based models. This dataset underpins the High-LoTw flavour of the Reliability TF and has been released as an open-access resource accompanying the corresponding publication [2].

Together, these two project-generated datasets provide the realistic, multi-layer telemetry required for reproducible research on user-centric reliability, proactive scaling and OOD/anomaly detection in 6G video-streaming environments.

2.2 STANDARDIZATION EFFORTS

Standardisation activities play a crucial role in shaping reliability and QoE requirements for next-generation networks. Relevant ETSI and 3GPP specifications were seriously considered during the design of the Reliability TF:

ETSI GR NFV-REL 014 V5.1.1 (October 2023) defines reliability evaluation approaches for cloud-native Virtualised Network Functions (VNFs). The specification examines key use cases, such as containerised VNF software modification, scaling-out and scaling-in, and discusses functionalities that support resiliency assurance, using Kubernetes as a reference containerisation technology [46]. These principles directly influence the design of reliability mechanisms in distributed 6G systems, especially in scenarios involving dynamic scaling, multi-tenant infrastructures and edge-cloud coordination. At the time of writing, V5.1.1 remains the latest published version.

Complementarily, **ETSI GS NFV-IFA 055 V5.3.1** (August 2025) specifies non-functional reliability requirements for parts of the Network Functions Virtualization Infrastructure (NFVI) and NFV-MANO, including physical and virtualised resources, the Virtualised Infrastructure Manager (VIM) and Container Infrastructure Service Management (CISM) [47].

ETSI TR 126 909 / 3GPP TR 26.909 V19.0.0 (October 2025, Release 19) provides guidelines for improved QoE reporting in multimedia services [48]. The specification analyses metrics, reporting mechanisms (including QoE Measurement Collection) and measurement methodologies that support reliability assessment from a user-centric perspective. These guidelines are highly relevant for immersive applications, video streaming and latency-sensitive services, where reliability is closely tied to user-perceived performance.

Additional related standardisation activities, related to the reliability in XR services include:

- **3GPP TR 26.928 V19.0.0** (Extended Reality in 5G) [49] and **3GPP TR 26.926 V19.0.0** (Traffic Models and Quality Evaluation Methods for Media and XR Services) [50], which address QoE metrics, traffic models and quality evaluation for XR and media services;
- **3GPP TS 28.404** (QoE measurement collection - Concepts, use cases and requirements) [51];
- Studies on Virtual Reality (VR)/XR QoE metrics (**3GPP TR 26.929** [52]) and policy management for virtualised network functions (**3GPP TS 28.311** [53]).

Together, these standards provide a foundational reference for aligning Reliability TF with industry-accepted reliability and QoE principles, ensuring interoperability and compliance within the broader 6G ecosystem.

3 ROLE OF THE TRUST FUNCTION IN SAFE-6G

The Reliability TF is one of the five user-centric TFs defined in SAFE-6G and is responsible for assessing and maintaining the reliability of services delivered across the edge-cloud continuum. Within the SAFE-6G cognitive framework, the Reliability TF interprets reliability as a dynamic, user-centric characteristic that reflects service continuity, QoE stability, and predictable behaviour under varying operational conditions.

This TF operates as part of the broader cognitive loop orchestrated by the CoCo. It receives trust-related inputs, such as the non-calibrated LoTw (nLoTw) – which conveys information about the user intent and contextual user metadata, and multi-layer telemetry, and transforms them into reliability-aware decisions and corresponding actions performed in the different planes of the SAFE-6G ecosystem to ensure reliability of applications. These decisions contribute to the computation of the calibrated LoTw (cLoTw), which the CoCo uses to coordinate actions across all TFs. The Reliability TF does not override or duplicate the responsibilities of other TFs; instead, it provides reliability-specific assessments and enforcement signals that complement safety, security, privacy, and resilience mechanisms.

Conceptually, the Reliability TF processes three categories of trust signals:

1. **Service-level signals**, such as QoE indicators, multi-layered metrics, and application-level behaviour.
2. **Infrastructure-level signals**, including resource utilisation, performance degradation patterns, and anomaly indicators.
3. **User-centric signals**, reflecting user intent, user metadata, and tolerance to service variability.

These categories align with the user-centric reliability principles highlighted in recent research [3], [4], and with the multi-layer monitoring and anomaly detection approaches explored in SAFE-6G-related studies [1], [6].

Based on these inputs, the Reliability TF produces three types of outputs:

1. a **reliability trust score**, representing the TF's assessment of current reliability conditions,
2. a **reliability outcome score**, representing the TF's assessment of current reliability conditions after the conflict resolution of the CoCo and
3. **reliability enforcement actions** (for example scaling, resource re-allocation or traffic-filtering decisions), which guide the behaviour of the vApp and nApp layers in subsequent sections.

The continuous LoTw supplied by the CoCo is discretised into three operational flavours, which are the Low, Medium and High. Each flavour corresponds to a different computational budget and detection capability of the AI/ML models hosted in the vApp layer, ranging from lightweight classical models (Low LoTw) to multi-layer fusion models (Medium LoTw) and transformer-/LLM-based detectors (High LoTw). The concrete mapping of LoTw ranges to models, the associated telemetry requirements and the resulting enforcement actions are detailed in Section 0.

The interaction boundaries of the Reliability TF are strictly defined in D4.1 [54]. The TF exchanges trust-related information with the CoCo, collaborates with other TFs through the shared trust-evaluation cycle, and interfaces with the underlying infrastructure through abstracted mechanisms exposed at the vApp and nApp layers. These interactions remain conceptual in this section; the detailed architecture is presented in Section 0, while the layer-by-layer responsibilities, model implementations and operational flows are specified in Section 0.

4 REFERENCE ARCHITECTURE

The Reliability TF builds upon the TF principles established in D4.1 [54], the intent-processing mechanisms described in D2.1 [55]/D2.2 [56], and the multi-layer monitoring and AI-driven reliability concepts explored in recent research [2], [4], [5], [6]. It consists of three layers: the Local AI Agent, which orchestrates the TF performance based on the received trustworthiness score; the vApp, which makes trust-related decisions; and the nApp, which interfaces with the network and service environment. Together, these layers enable the Reliability TF to interpret reliability signals, compute trust scores, and apply reliability-aware actions in alignment with user intent and contextual conditions.

In Figure 1, the high-level overview of the Reliability TF is featured.

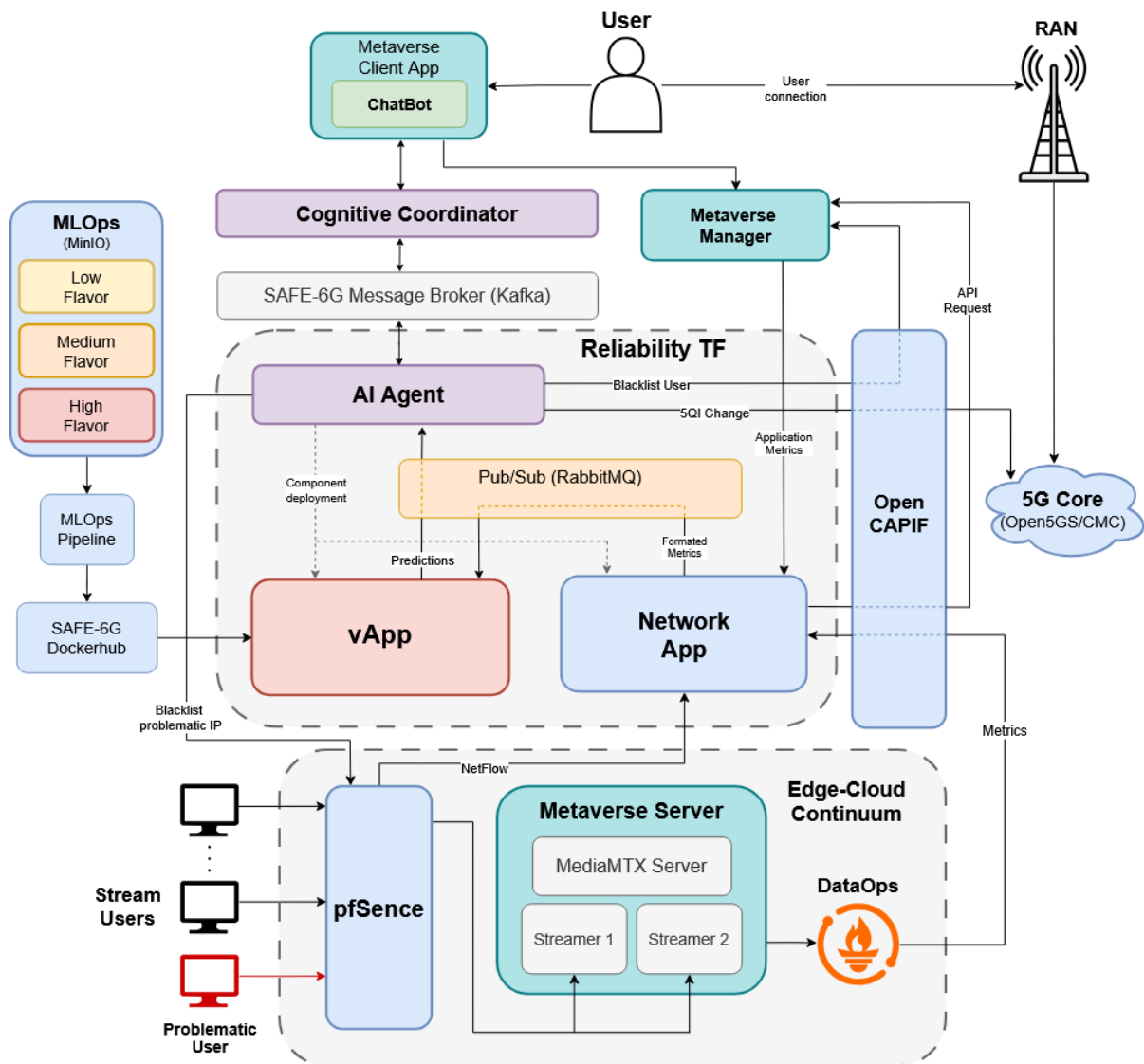


Figure 1: High-level overview of the Reliability TF

Each component within Figure 1 is described below:

- **User:** The User, wearing the XR headset, connects wirelessly to the Metaverse Client App and communicates with the Chatbot and views a walkthrough video streamed by the Metaverse Server.
- **Metaverse Client App:** It hosts the XR Metaverse environment in which the User interacts. It provides a way for the user to interact with the walkthrough video of Use-case 1 (UC1) – Factory Digital Twin and the Chatbot. Further information about the Use Cases and the Metaverse Applications can be found in D2.3 [57].
- **Chatbot:** Allows the User to interact with the system and translates the User’s intents to a reliability request to the CoCo. Details regarding the Chatbot are available in D2.4 [58] and D3.1 [59].
- **CoCo:** The CoCo is responsible for interpreting the User’s intents to a LoTw and coordinates all the TFs by resolving any conflicts between them. Further information about the CoCo can be found in D2.2 [56] and D2.4 [58].
- **Reliability TF:**
 - **Local AI Agent:** Communicates with the CoCo through the SAFE-6G Message Broker, exchanging messages to receive the nLoTw and the cLoTw. The Local AI Agent deploys the components with the corresponding flavour to the nLoTw through the Meta-OS platform. Further information about the Local AI Agent can be found in Section 5.3.
 - **RabbitMQ:** Acts as the intermediary of the messages exchanged between the Reliability TF components.
 - **vApp:** Receives AI/ML model(s) from the MLOps platform, performs inference using the selected model(s) and sends those predictions to the Local AI Agent. Further information about the vApp can be found in Section 5.4.
 - **nApp:** The nApp gathers multi-layered metrics across the Edge-Cloud Continuum and sends them to the vApp. Further information about the nApp can be found in Section 5.5.
- **MLOps:** Contains the AI/ML models corresponding to each flavour. The models are served through the MLOps pipeline and SAFE-6G DockerHub, using the MinIO platform. Further information about the MLOps platform and pipeline can be found in D2.2 [56], D2.4 [58], D3.1 [59], D3.5 [60].
- **Edge-Cloud Continuum:**
 - **DataOps:** Provides access to the multi-layered real-time data collected across the Edge-Cloud Continuum through Prometheus. Further information about DataOps can be found in D3.1 [59].
 - **Metaverse Server:** Hosts a MediaMTX setup that can stream video content to stream users. Moreover, the streamed video is a walkthrough video of the Metaverse Application.
 - **PfSense:** Acts as a firewall and router, managing all the traffic flowing to the MediaMTX Streamers.
- **Open CAPIF:** Provides secure, standardized, and scalable API exposure to the Reliability TF. Further information about the Open CAPIF can be found in D3.1 [59], D3.4 [61].

- **Radio Access Network (RAN):** Provides mobile access between the User and the 5G Core. It enables the transmission of Metaverse traffic over the 5G network and is affected by QoS adjustments performed by the Reliability TF to the 5G Core.
- **5G Core:** The 5G Core deployment [62] in SAFE-6G is based on two configurations, the Open5GS and CMC Core, both interacting with adjustments made by the Reliability TF using the “*AsSessionWithQoS*” API through CAPIF. Further information about the 5G Core APIs can be found on D3.4 [61].
- **Metaverse Manager:** Serves and exposes all the API endpoints and manages all the requests exchanged to the Metaverse Environment. More information about the Chatbot can be found on D3.1 [59], D3.6 [63].
- **Stream users:** Represent users connected to the Metaverse Server to consume video content. There are two types of users: “**normal**” and “**problematic**”. A **normal user** is one that does not unnecessarily stress the video streaming service. A **problematic user** actively stresses the service in various ways; this results normal users’ QoE of degrading or even for the service to be severed. More details about the problematic user will be covered in D5.3 deliverable.

4.1 REQUIREMENTS

The design of the Reliability TF is driven by the functional and non-functional requirements defined in Deliverable D2.1 [55] (Section 5.8.2). In addition to the common requirements applicable to all TFs (Section 5.1.2 of D2.1 [55]), the following Reliability-specific requirements were established in Table 1:

ID	Short Description	Realization in Reliability TF	Addressed in
REQ-FREL-NET-F-M-01	Interface to communicate with the cognitive SAFE-6G coordination component	The TF exposes an interface that enables bidirectional communication with the CoCo, allowing the exchange of trust-related events, the reception of the nLoTw and the reporting of the calibrated reliability trust score [5].	Sections 0, 0, 5.2 and 5.3
REQ-FREL-AI-NF-M-02	Training of reliability AI models	The TF trains AI/ML models capable of predicting potential service breaking points (e.g., resource exhaustion leading to QoE degradation) and of identifying abnormalities or malicious actions that threaten service reliability [1], [6].	Sections 5.4, 6.2 and 6.3
REQ-FREL-AI-NF-M-03	Model selection per LoTw	The TF selects and deploys different AI/ML models according to the requested LoTw, ranging from lightweight classical models (Low LoTw) [4] to multi-layer fusion and transformer-based models (High LoTw) [2].	Sections 5.1, 5.3 and 5.4
REQ-FREL-RES-NF-O-04	Provision of High-Performance Computing (HPC) infrastructure for	When the complexity of the selected models (number of features, volume of training data) requires it, the TF leverages HPC resources to accelerate training and inference.	Sections 5.4

	training of ML models		
REQ-FREL-RES-NF-M-05	Function interoperability	The TF consumes streaming multi-layer monitoring data (infrastructure, virtualisation and application-level metrics) provided by the SAFE-6G monitoring system [1], [6].	Sections 5.5, 6.3 and Table 2
REQ-FREL-NET-F-R-06	Restrict unauthorized access	The TF executes in a secure environment that is not directly accessible from public networks.	Sections 4.2 and 5.3
REQ-FREL-NET-NF-R-07	Evaluation report	At the post-deployment phase, the TF interacted with the MLOps platform to generate an evaluation report containing relevant performance metrics (e.g., attainable accuracy, training time, inference latency).	Sections 5.4 and 6.3

Table 1: Reliability TF Requirements

4.2 ARCHITECTURAL CONSIDERATIONS OF THE USER-CENTRIC RELIABILITY FUNCTION

The architectural design of the Reliability TF reflects the shift toward user-centric, AI-native, and distributed 6G systems. Rather than treating reliability as a static network property, the TF interprets it as a dynamic, context-aware dimension of trust that evolves with user behaviour, service conditions, and infrastructure state. A key architectural consideration is the **alignment with user intent**. The TF must incorporate user-level preferences and QoE expectations into its reliability assessment, ensuring that trust decisions reflect what the user perceives as acceptable service performance [5]. The architecture also emphasises **cross-layer visibility**, leveraging multi-layer telemetry to capture reliability-relevant signals from both infrastructure and application components. This enables the TF to detect degradation patterns that may not be visible at a single layer, consistent with findings from multi-layer monitoring research [6]. Another consideration is **AI-driven inference**, which allows the TF to move beyond threshold-based reliability models. ML-based and transformer-based approaches support predictive reliability assessment, anomaly detection, and adaptive enforcement user-centric. The architecture must also support **distributed operation** latency-sensitive services, while broader reliability trends may be - in the cloud. This distributed design aligns with the SAFE-6G. Finally, the Reliability TF is designed to be **modular and interoperable**, integrating seamlessly with other TFs through the CoCo-reliability specific trust signals without duplicating the responsibilities of privacy, safety, security, or resilience functions, ensuring a coherent and unified trustworthiness framework.

5 TRUST FUNCTION LAYERS

The Reliability TF is structured according to the WP4 reference architecture defined in D4.1 [54]. The three cooperating layers enable the TF to interpret multi-layer reliability signals, compute trust scores, and apply reliability-aware actions aligned with user intent and contextual conditions. The codes of the deployment phase are in the SAFE-6G Gitlab [64].

The following subsections describe the responsibilities, interactions, and high-level logic of each layer, without delving into deployment-specific details.

5.1 MAIN FUNCTIONALITIES AND ML MODELS ACCORDING TO THE LoTw

Based on the high-level architecture of the Reliability TF (Figure 1), the Local AI Agent is responsible for the orchestration of the other two components: it selects, deploys and validates the appropriate flavour of the vApp and nApp on the designated cluster according to the LoTw received from the CoCo. In addition, the Local AI Agent can directly apply selected enforcement actions on the 5G Core (for example, modification of the 5QI value or blacklisting of a problematic user) when instructed by the reliability predictions of the vApp.

The Reliability TF adapts its behaviour based on the received LoTw by the CoCo. The LoTw determines the flavour of the TF that is deployed, influencing the complexity of the AI/ML models used, the depth of telemetry processing, and the strictness of reliability enforcement actions. Three operational flavours are defined.

5.1.1 Low LoTw (1–40 %)

In Figure 2, the Low flavour realises a lightweight reliability assessment that is suitable for resource-constrained environments. It employs classical ML models, principally RF and, optionally, k-Nearest Neighbours (k-NN) or simplified neural networks. The models operate on a reduced set of telemetry features drawn primarily from the Edge-Cloud Continuum and the Metaverse Server, such as RTSP session metrics and memory-utilisation statistics in Table 2 [4]. The primary outputs are a binary reliability prediction (reliable / unreliable) and a basic reliability trust score. Based on these outputs the Local AI Agent may trigger simple scale-up actions on the MediaMTX streamer instances in order to restore or improve the QoE of the delivered walkthrough video to the user.

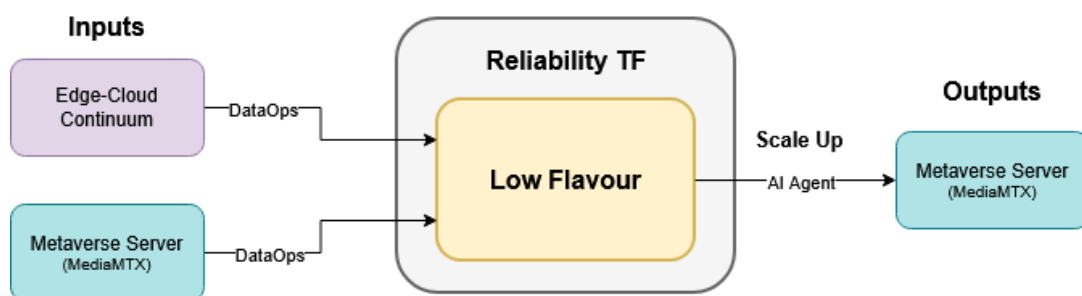


Figure 2: Low Flavour

5.1.2 MEDIUM LoTW (41–80 %)

In Figure 3, the Medium flavour extends the Low flavour by employing a more expressive DNN that fuses multi-layer telemetry. The DNN architecture consists of three dense hidden layers of 16 neurons with ReLU activations and a dropout of 0.1; training is performed with the Adam optimiser and early stopping [4]. The input feature vectors combine application-level, container-level, infrastructure-level and selected network-flow metrics in Table 2. In addition, the Reliability TF obtains information about the active user through the QoS API (AsSessionWithQoS). The model produces a higher-accuracy reliability prediction and an associated confidence score. The Local AI Agent uses these predictions to trigger proactive vertical or horizontal scaling of the MediaMTX streamers, and or a QoS adjustment utilizing a 5QI change to the 5G Core before a major QoE degradation becomes visible to the end user.

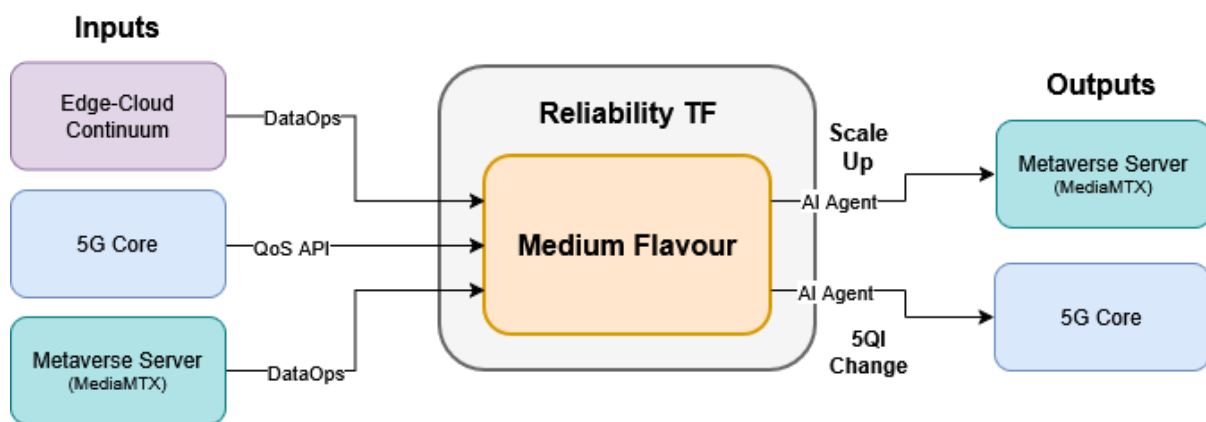


Figure 3: Medium Flavour

5.1.3 HIGH LoTW (81–100 %)

In Figure 4, the High flavour further extends the monitoring and detection capabilities of the Reliability TF by incorporating network-layer telemetry (IPFIX / NetFlow) in addition to application and container metrics. Two complementary AI/ML pipelines are employed:

- a DistilBERT-based transformer model that performs binary multi-layer attack detection on deterministically serialised feature-value sequences, distinguishing benign traffic from known attack patterns (SYN flooding, RTSP session flooding, slow RTSP and brute-force authentication) [2];
- complementary LLM-based pipelines that support OOD attack detection under a leave-one-attack-out protocol [1].

The High flavour therefore provides both high-accuracy detection of complex multi-layer anomalies and the ability to recognise previously unseen attacks. Based on the model outputs, the Local AI Agent may trigger scale-up actions, adjust the 5QI value at the 5G Core, or apply a blacklist rule (via the Metaverse Manager and pfSense) in order to protect the QoE of normal users.

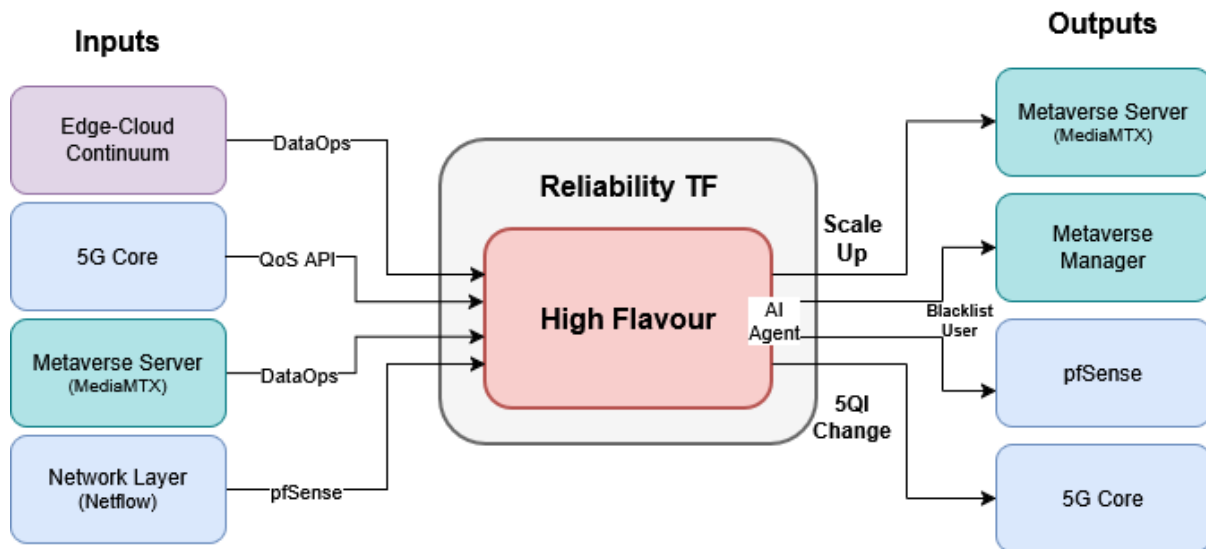


Figure 4: High Flavour

In all three flavours the Local AI Agent remains the sole decision-making entity: it receives the predictions from the active vApp, interprets them in the context of the current user intent and service conditions, and issues the corresponding enforcement actions. Training of the models is performed via the MLOps platform; the Local AI Agent itself does not perform training or inference. The detailed implementation of each flavour inside the vApp layer, together with the concrete telemetry interfaces provided by the nApp, is presented in Sections 5.4 and 5.5 respectively.

5.2 SEQUENCE DIAGRAM

The following sequence diagram describes the sequence of actions associated with UC1, as presented in D2.3 [57]. It provides a representation of the interactions and message exchanges between the relevant components. The Reliability TF follows the trust-evaluation cycle defined in D4.1 [54]. In Figure 5, the conceptual sequence is as follows:

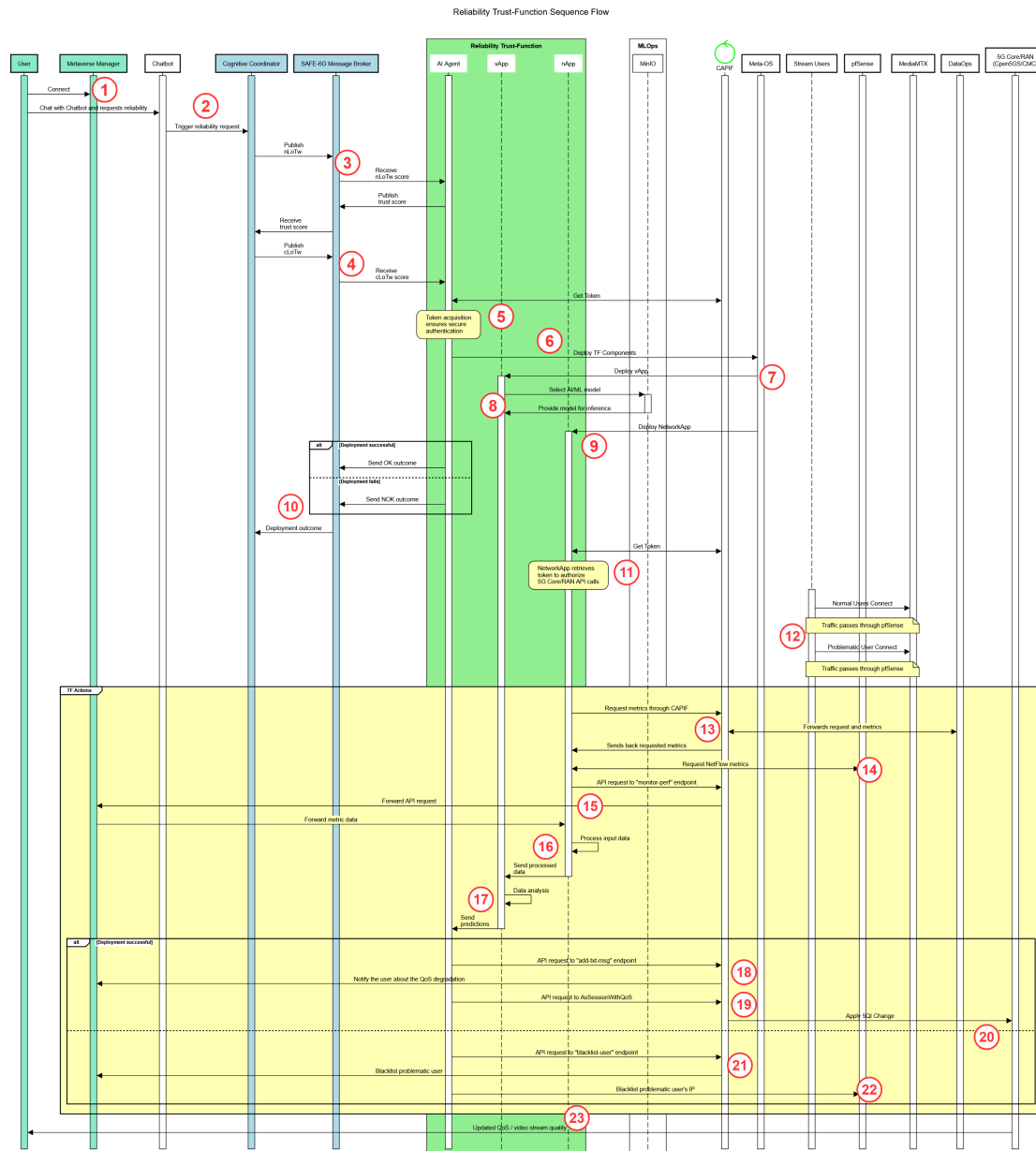


Figure 5: Sequence Diagram of the Reliability TF

1. The Reliability TF is being activated when a **user connects** to the **Metaverse Manager**, **interacts** with the **chatbot** in the Metaverse XR environment and requests for a reliable service.
2. The **chatbot** triggers a **reliability request** to the **CoCo**.
3. The **CoCo sends** the nLoTw to all relevant TFs (in this case the Reliability TF) then the Reliability TF (in parallel with any other relevant TFs) reports back the **trust score**.
4. The **CoCo** then sends the **cLoTw** to all relevant TFs.
5. The **Local AI Agent** requests an **authentication token** from CAPIF.
6. The **Local AI Agent** requests from the **Meta-OS** the **deployment of the vApp and nApp**.
7. The **Meta-OS deploys** the vApp flavour corresponding to the cLoTw.
8. The **vApp requests** the AI/ML model(s) from **MinIO** according to the selected flavour.

9. The **Meta-OS** deploys the nApp flavour corresponding to the **cLoTw**.
10. The **Local AI Agent** then (in parallel with the relevant TFs) **reports** back to the CoCo the deployment output.
11. The **nApp** retrieves a token to **authorize 5G Core/RAN API calls**.
12. The **“normal”** and the **“problematic”** users are connected to the MediaMTX streamers through the pfSense.
13. Then the **nApp gathers** performance metrics from **DataOps** through CAPIF, which forwards them back.
14. The **nApp requests** network-level data from the pfSense.
15. The **nApp requests** application-level metrics from the **“monitor-perf”** Metaverse Manager API endpoint through CAPIF, which forwards them back and then the **Metaverse Manager continuously sends** metrics to the nApp.
16. The **nApp formats and forwards** the data to the **vApp** which provides data to its AI/ML model (the models are dependent on the flavour of the vApp (see Section 5.4).
17. The **AI/ML model(s) continuously predict** whether a QoS adjustment is needed (**5QI change**) and/or a malicious user is being added to the blacklist, with its IP, if it is needed.
18. If a **QoS adjustment is needed**, the Local AI Agent uses **“add-txt-msg”** Metaverse Manager API through CAPIF to notify the user about the QoS degradation.
19. The **Local AI Agent requests** the **“AsSessionWithQoS”** API through CAPIF and **applies** QoS adjustment (e.g., increases the 5QI).
20. The **5G Core/RAN** applies the updated QoS parameters (**5QI change**).
21. If the **“problematic”** user is degrading the service quality, the AI/ML model predicts which one is it, and the **Local AI Agent adds** it to the blacklist by calling the **“blacklist-user”** Metaverse Manager API endpoint through CAPIF.
22. Also, the **Local AI Agent** adds it to the blacklist their IP as well through the pfSense.
23. The **User experiences** improved connection quality.

This conceptual flow aligns with the SAFE-6G trust-evaluation cycle and does not include implementation-specific details such as APIs, QoS parameters, or XR-specific logic (covered in Section 0).

5.3 LOCAL AI AGENT

The Local AI Agent operates as the core decision-making component of the Reliability TF, it orchestrates the TF’s behavior, selects the appropriate “flavour” based on LoTw from CoCo, interprets reliability predictions to produce enforcement actions from the vApp, and can also make changes directly to the 5G Core by utilizing 5QI value changes or blacklist problematic user from the service. The Local AI Agent has been designed with Meta-OS compatibility ensuring integration across the edge-cloud continuum.

5.3.1 TECHNICAL RESPONSIBILITIES

- **Trust score computation:** It processes and decides on a reliability trust score to send to the CoCo.

- **Flavour selection and deployment:** Based on the cLoTw, the Local AI Agent selects the appropriate reliability flavour (low, medium, high) and requests the deployment of the corresponding vApp and nApp components through Meta-OS [54].
- **Reliability inference orchestration:** The Local AI Agent receives predictions from the vApp's AI/ML model(s) and interprets them in the context of user intent and service conditions [4].
- **Enforcement decision-making:** It determines whether reliability-aware actions are required (e.g., resource adjustments, QoE-driven adaptations, anomaly responses) [4].

The predictions that the Local AI Agent receives from the vApp could be either a Boolean value or even an IP address of a user with suspicious behaviour. The actions that Local AI Agent can perform are horizontal and/or vertical scale up enforcement actions, executing an API request to **AsSessionWithQoS** (more information in D3.3 [62]) to change directly parameters in the 5G Core/RAN by utilizing 5QI value changes, it can also notify the user about QoS degradation by calling the **add-txt-msg** Metaverse Manager API endpoint. Furthermore, the Local AI Agent can blacklist a user using the **blacklist-user** Metaverse API endpoint and in parallel blacklist the user's IP address as well.

The Local AI Agent does not perform training or inference; training pipelines belong to MLOps and inference happens during the vApps runtime.

5.4 VERTICAL APPLICATIONS (VAPPS)

The vApp layer implements the core reliability logic of the TF and acts as the intermediary between the Local AI Agent and the nApp. Its role is to transform multi-layer telemetry into reliability-aware insights that reflect user intent, service behaviour, and contextual conditions across the edge-cloud continuum. The vApp receives structured telemetry metrics from the nApp. These metrics include application-level QoE metrics, container-level resource usage, and infrastructure-level performance indicators. The vApp normalises and aligns these signals to ensure consistent input for the AI/ML model [4], [6]. The vApp hosts the ML model corresponding to the LoTw flavour selected by the Local AI Agent.

5.4.1 LOTW FLAVORS

Each flavour reflects different computational budgets and reliability requirements:

The **Low LoTw flavor** implements a lightweight QoE prediction model, suitable for edge nodes with limited resources [4]. The model is based on RF Regressor, which is configured with a small number of trees and limited depth (for our studied cases with parameters `n_estimators=10`, `max_depth=15`) to achieve low computational costs and minimal inference latency. The model takes a subset of multi-layer telemetry, including CPU utilisation metrics, memory pressure, container-level indicators, and basic application-level metrics such as bit rate and stall events, as described in Table 2. Data is preprocessed with MinMax scaling in range [0,1], while training is performed by windowing 40 samples and train/test split 80/20. The evaluation of the model in the SAFE-6G multi-layer dataset [7] demonstrates stable performance with very low latency and high accuracy (see Section 6.3.1), making

it suitable for real-time reliability in Low LoTw conditions as illustrated in Figure 6. The choice of RF for this flavor is justified by the small model footprint, the direct loading to the vApp and its ability to operate reliably on heterogeneous edge infrastructures.

```

98 $ docker exec internal-container pytest tests/ -v -s --log-cli-level=INFO
99 ===== test session starts =====
100 platform linux -- Python 3.10.20, pytest-7.4.4, pluggy-1.5.0 -- /usr/local/bin/python3.10
101 cachedir: .pytest_cache
102 rootdir: /app
103 plugins: mock-3.14.0, anyio-4.6.0
104 collecting ...
105 collected 1 item
106 tests/test_model_api_aeros.py::test_model_inference_accuracy
107 ----- live log call -----
108 INFO root:test_model_api_aeros.py:106 Loaded 568 raw windows of shape (40, 1)
109 INFO root:test_model_api_aeros.py:119
110 ===== SCALED FEATURES FROM SERVER =====
111 INFO root:test_model_api_aeros.py:120 Shape: (568, 14)
112 INFO root:test_model_api_aeros.py:121 Min: 0.0000, Max: 1.0000
113 INFO root:test_model_api_aeros.py:122 First sample: [0.04415122 0.04214221 0.03768056 0.06302882 0.05492401 0.03978728
114 0.00088147 0.03604537 0.35673779 0.00086574 0.7274517 0.
115 0.01219526 0.35596225]
116 INFO root:test_model_api_aeros.py:123 Last sample: [0.65841271 0.65093188 0.65640711 0.5250655 0.64374211 0.64262891
117 0.16520104 0.08830209 0.32152806 0.26933953 0.75432551 0.
118 0.48878244 0.99771818]
119 INFO root:test_model_api_aeros.py:124 Feature means:
120 [0.3798947 0.35459042 0.46849181 0.2857104 0.32729982 0.40851182
121 0.21312294 0.05215161 0.47321183 0.28957445 0.75123994 0.0915493
122 0.24190257 0.96272149]
123 INFO root:test_model_api_aeros.py:125 =====
124 INFO root:test_model_api_aeros.py:139

```

Figure 6: RF inference logs / feature scaling output

The **Medium LoTw flavor** implements a DNN model, which utilizes full multi-layer telemetry to predict QoE degradation and reliability scores with higher accuracy. The model follows the architecture presented in the [4], consisting of three dense hidden layers (16 neurons) with ReLU activations and a dropout of 0.1 to avoid overfit. Training is performed with Adam optimizer (learning rate= 10^{-3}) and MSE loss, while the input data includes fused feature vectors from application-level, container-level, infrastructure-level and network-flow metrics, as described in Table 2: Multi-Layer Telemetry Metrics. DNN is trained with batch size 32 and early stopping, leveraging the SAFE-6G multi-layer dataset [7]. The evaluation results show a significant improvement in accuracy and F1-score over the RF baseline, particularly in cases of high bitrate and container pressure variability, as illustrated in Figure 7. Despite the increased computational costs, the inference latency remains at acceptable levels for Medium LoTw conditions, making DNN suitable for environments where a balance between accuracy and computational efficiency is required.

```

=== Fold 1 ===
Training on: 1 streamer 100MiB, 2 streamer 100MiB, 3 streamer 100MiB, 1 streamer 200MiB
Validation on: 2 streamer 200MiB
Testing on: 3 streamer 200MiB
DNN params: {'hidden_layers': 2, 'units': 4, 'dropout_rate': 0.1, 'learning_rate': 0.001, 'batch_size': 32, 'epochs': 100, 'fold': 'Fold 1 (Test: 3 streamer 200MiB)', 'model': 'DNN', 'best_f1_score': 1.0}
DNN accuracy: 0.9906, precision: 0.9916, recall: 0.9906, f1: 0.9908, AUROC: 0.9951, AUPRC: 0.9460, ECE: 0.0044
DNN per-class accuracy: Class 0 = 0.9906, Class 1 = 0.9906
DNN per-class precision: Class 0 = 1.0000, Class 1 = 0.8966
DNN per-class recall: Class 0 = 0.9898, Class 1 = 1.0000

=== Fold 6 ===
Training on: 3 streamer 200MiB, 1 streamer 100MiB, 2 streamer 100MiB, 3 streamer 100MiB
Validation on: 1 streamer 200MiB
Testing on: 2 streamer 200MiB
DNN params: {'hidden_layers': 2, 'units': 4, 'dropout_rate': 0.1, 'learning_rate': 0.001, 'batch_size': 32, 'epochs': 100, 'fold': 'Fold 6 (Test: 2 streamer 200MiB)', 'model': 'DNN', 'best_f1_score': 1.0}
DNN accuracy: 1.0000, precision: 1.0000, recall: 1.0000, f1: 1.0000, AUROC: 1.0000, AUPRC: 1.0000, ECE: 0.0151
DNN per-class accuracy: Class 0 = 1.0000, Class 1 = 1.0000
DNN per-class precision: Class 0 = 1.0000, Class 1 = 1.0000
DNN per-class recall: Class 0 = 1.0000, Class 1 = 1.0000

```

Figure 7: DNN inference output

The **High LoTw flavor** implements advanced transformer- and LLM-based models designed for high-accuracy multi-layer anomaly and attack detection. Two complementary capabilities are supported, both operating on fused telemetry from the application, container and network-flow (IPFIX) layers:

- **Multi-layer binary attack detection** [2]. A Distil-BERT encoder is adapted to process deterministic textual representations of the multi-layer telemetry. Data from the application, container and IPFIX layers (as described in Table 2) are converted into ordered feature-value sequences and fed to a fine-tuned Distil-BERT model. The model is trained with class-weighted cross-entropy loss with a learning rate 2×10^{-5} and performs binary classification, distinguishing benign traffic from malicious activity. The malicious class aggregates four attack scenarios: SYN flooding, RTSP session flooding, slow RTSP connections and brute-force authentication. Evaluation on the SAFE-6G multi-layer attack detection dataset [8] shows a substantial improvement in attack-class recall (approximately +6.15 percentage points) over a strong tabular RF baseline, while maintaining near-perfect precision.
- **OOD attack detection** [1]. Complementary LLM-based pipelines convert network-traffic samples into textual feature-value representations and are evaluated under a strict leave-one-attack-out protocol. These models enable the identification of previously unseen attack types and achieve up to a 5 percentage-point improvement in malicious recall compared with a non-LLM baseline (LightGBM) on the 5G-NIDD dataset [10].

Despite the higher inference latency relative to classical models, the High LoTw approaches are well suited to scenarios that demand maximum detection accuracy and robustness against both known multi-layer anomalies and previously unseen attacks. In Figure 8, the logs illustrate correct detection of attack flows and correct classification of benign traffic by the DistilBERT- and LLM-based pipelines.

```
flow #45451 [true label: Attack]
input (raw features): Bytes=182, Packets=1, IPTTL=0, rtsp_conns=608, rtsp_sessions_bytes_sent=3717139501, container_memory_usage_bytes=406192128, container_cpu_usage_cores=0.7944309958750737, container_memory_working_set_bytes_max=202612736, container_memory_failure_events_1m=727257.5132586919, container_restart_events_1m=0.0, flag_fin=0, flag_syn=0, flag_rst=0, flag_psh=1, flag_ack=1, flag_urg=0, DstPort=37324, SrcPort=8554, Proto=6
input (flow-text) : flow: bytes 182 packets 1 ttl 0 rtspconns 608 rtspbytes 3717139501 mem 406192128 cpu 0 memms 202612736 memfail 727257 restart 0 port ephemeral syn 0 rst 0 ack 1 proto 6
[OK] RF-tabular -> pred=Attack (conf=0.947)
[XX] RF-frozen-emb -> pred=Normal (conf=0.927)
[OK] RF-finetuned-emb -> pred=Attack (conf=1.000)
[OK] LLM-rawtext -> pred=Attack (conf=1.000)

flow #50000 [true label: Attack]
input (raw features): Bytes=181, Packets=2, IPTTL=0, rtsp_conns=708, rtsp_sessions_bytes_sent=5311149169, container_memory_usage_bytes=448733184, container_cpu_usage_cores=0.2502497753818509, container_memory_working_set_bytes_max=223883264, container_memory_failure_events_1m=128417.40469772815, container_restart_events_1m=0.0, flag_fin=0, flag_syn=0, flag_rst=0, flag_psh=1, flag_ack=1, flag_urg=0, DstPort=8554, SrcPort=60586, Proto=6
input (flow-text) : flow: bytes 181 packets 2 ttl 0 rtspconns 708 rtspbytes 5311149169 mem 448733184 cpu 0 memms 223883264 memfail 128417 restart 0 port rtsp syn 0 rst 0 ack 1 proto 6
[OK] RF-tabular -> pred=Attack (conf=0.547)
[XX] RF-frozen-emb -> pred=Normal (conf=0.867)
[OK] RF-finetuned-emb -> pred=Attack (conf=0.967)
[OK] LLM-rawtext -> pred=Attack (conf=1.000)

flow #13887 [true label: Normal]
input (raw features): Bytes=16931650, Packets=11677, IPTTL=0, rtsp_conns=43, rtsp_sessions_bytes_sent=18802181577, container_memory_usage_bytes=542556160, container_cpu_usage_cores=0.5818063537380973, container_memory_working_set_bytes_max=116477952, container_memory_failure_events_1m=87084.66546918376, container_restart_events_1m=0.0, flag_fin=0, flag_syn=0, flag_rst=0, flag_psh=1, flag_ack=1, flag_urg=0, DstPort=36272, SrcPort=8554, Proto=6
input (flow-text) : flow: bytes 16931650 packets 11677 ttl 0 rtspconns 43 rtspbytes 18802181577 mem 542556160 cpu 0 memms 116477952 memfail 87084 restart 0 port ephemeral syn 0 rst 0 ack 1 proto 6
[OK] RF-tabular -> pred=Normal (conf=1.000)
[OK] RF-frozen-emb -> pred=Normal (conf=0.983)
[OK] RF-finetuned-emb -> pred=Normal (conf=1.000)
[OK] LLM-rawtext -> pred=Normal (conf=1.000)
```

Figure 8: Example inference logs of the High LoTw models on multi-layer telemetry records

5.4.2 TECHNICAL RESPONSIBILITIES

The vApp performs inference continuously, producing predictions such as suggestions for optimal moments for scale up actions based on QoE degradation, and anomaly detection alerts. Additionally, it communicates reliability predictions, alerts, and policy-driven signals to the Local AI Agent. This communication follows the TF interface defined in D4.1 [54]:

- **Local AI Agent → vApp:** flavour selection, model activation
- **vApp → Local AI Agent:** reliability predictions, anomaly alerts, policy-triggered alerts

The vApp does not directly interact with network infrastructure; all enforcement actions are delegated to the Local AI Agent and executed through the nApp.

5.5 NETWORK APPLICATION (NAPP)

The nApp is the network-facing component of the Reliability TF. It provides multi-layer telemetry to the vApp and ensures that reliability inference is based on accurate, timely, and service-aligned monitoring data. The nApp abstracts heterogeneous monitoring sources into unified feature representations that support cross-layer reliability assessment.

5.5.1 TECHNICAL RESPONSIBILITIES

The nApp gathers metrics from all operational layers of the continuum [4], [6]:

- **Physical servers:** CPU load, memory usage, energy consumption
- **Virtualisation layers:** hypervisor metrics, container statistics, resource pressure
- **Application components:** RTSP session metrics, frame continuity indicators
- **Network infrastructure:** latency, throughput, QoS indicators, congestion signals

A detailed overview of all the metrics from each operational layer is shown in Table 2.

Layer	Source	Name	Description
Infrastructure / Physical	DataOps/ Kepler	kepler_container_core_joules_total	Measures the total energy in joules of the used CPU cores that the container runs on.
Infrastructure / Physical	DataOps/ Kepler	kepler_container_dram_joules_total	Measures the total energy consumption in joules, of the DRAM used by the container.
Virtualisation / Container Layer	DataOps/ Kepler	kepler_container_cache_miss_total	Counts the cache misses that occur during the container runtime. A cache miss occurs when the processor cannot find the requested data in the cache.
Infrastructure / Physical	DataOps/ Kepler	kepler_container_joules_total	Measures the total energy consumption in joules, of the container runtime.
Virtualisation / Container Layer	DataOps/ Kubernetes	container_memory_cache	Measures the cache memory used by the container.

Virtualisation / Container Layer	DataOps/ Kubernetes	container_memory_failures_total	Counts the memory allocation failures
Virtualisation / Container Layer	DataOps/ Kubernetes	container_memory_failure_events_1m	Counts the container memory failure events observed over a one-minute window.
Virtualisation / Container Layer	DataOps/ Kubernetes	container_memory_usage_bytes	Measures the memory used by the container in bytes.
Virtualisation / Container Layer	DataOps/ Kubernetes	container_memory_working_set_bytes	Measures the working set memory used by the container in bytes.
Infrastructure / Physical	DataOps/ Kubernetes	container_cpu_usage_seconds_total	Measures the cumulative CPU time consumed by the container across all CPU cores in seconds.
Infrastructure / Physical	DataOps/ Kubernetes	container_cpu_usage_cores	Measures the CPU usage of the container expressed in CPU cores.
Infrastructure / Physical	DataOps/ Kubernetes	container_restart_events_1m	Counts the restarted container events observed over a one-minute window.
Virtualisation / Container Layer	DataOps/ Kubernetes	pod_container_status_running	Indicates whether the container within the pod is running.
Application Layer	DataOps/ MediaMTX	rtsp_sessions	Counts the active RTSP streaming sessions.
Network Layer	DataOps/ MediaMTX	rtsp_sessions_bytes_sent	Counts the total amount of data transmitted from RTSP streaming sessions in bytes.
Network Layer	DataOps/ MediaMTX	rtsp_sessions_bytes_received	Counts the total received data from RTSP streaming sessions in bytes.
Application Layer	Metaverse Manager	framerate	Measures the frame rate of the client XR application, in frames per second.
Infrastructure / Physical	Metaverse Manager	GPU_percent_usage	Computes the percentage of Graphics Processing Unit (GPU) used by the XR headset.
Infrastructure / Physical	Metaverse Manager	CPU_percent_usage	Computes the percentage of CPU used by the XR headset.
Application Layer	Metaverse Manager	screen_tear_count	Counts the screen tearing events during the client XR application runtime.
Application Layer	Metaverse Manager	stale_frame_count	Counts the stale frames during the client XR application runtime.
Application Layer	Metaverse Manager	stale_frame_consecutive	Counts the consecutive stale frames during the client XR application runtime.
Application Layer	Metaverse Manager	max_repeated_frames	Counts the repeated frames during the client XR application runtime.
Network Layer	pfSense	NetFlow/IPFIX	Various IPFIX network flow data (SrcIP, DstIP, TCP-Flags, etc.).

Table 2: Multi-Layer Telemetry Metrics

After deployment, the nApp requests the authorization token from CAPIF to access the API Invoker and to start the monitoring process. It collects data from multiple layers of the infrastructure such as the Metaverse Manager by interacting and listening back to the **monitor-perf** Metaverse API endpoint with a webhook setup, the DataOps through Prometheus and network-level metrics from the pfSense.

It queries the data sources and collects the predefined set of timestamped metrics every second. The nApp formats the metrics and sends them periodically for a predetermined amount of time for the optimal operation of the AI/ML models in the vApp.

6 TRUST EVALUATION & ENFORCEMENT LOGIC

This section describes how the Reliability TF evaluates the reliability of services and implements the corresponding enforcement actions, taking into account both the technical data of the system and the characteristics of the user. The evaluation process is based on the continuous collection of multi-layer telemetry, the execution of the models implemented in the vApps (as analyzed in Section 5.4), and the dynamic adaptation of LoTw by the Local AI Agent.

Reliability assessment is carried out at two levels: (a) **evaluation of the current state of the system**, through ML/DNN/Transformer models that predict QoE, detect anomalies and recognize OOD behaviors; (b) **prediction and application of reinforcement actions**, such as scaling, resource reallocation, flavor change, or activation of protection mechanisms.

Reliability TF also incorporates user-centric factors, which directly influence the evaluation of reliability and the dynamic selection of the appropriate LoTw flavor. The remainder of the section is organised as follows. Section 6.1 analyzes the impact of user characteristics. Section 6.2 reports methodological validation on two openly accessible external datasets. Section 6.3 presents the core experimental validation performed on the two multi-layer datasets generated and released within Task 4.6, demonstrating that the developed models fulfil the proactive reliability management and attack-detection objectives set out in the project.

6.1 IMPACT OF USER CHARACTERISTICS

The reliability assessment in SAFE-6G is not limited to technical system metrics but explicitly incorporates the user's characteristics, preferences, and tolerances. This approach is fully aligned with the philosophy of user-centric trustworthiness, as defined in D4.1 [54] and the recent research projects on AI-native 6G architectures [15], [16], [17], [18], [19], [20], [21]. In Reliability TF, user characteristics influence both the evaluation of reliability and the selection of the appropriate LoTw flavor.

This effect manifests itself in three dimensions:

1. **User QoE Sensitivity and Tolerance to Variability:** Different users exhibit varying tolerance for QoE fluctuations, such as bit rate drops, minor delays, or occasional stalls. For example, users who consume immersive or latency-critical content (e.g., XR, real-time video analytics) require higher reliability and enable Medium or High LoTw flavor more frequently. In contrast, users with lower QoE sensitivity can be served with Low LoTw flavor, where the RF model provides fast and light evaluation. QoE sensitivity is integrated into the TF through user metadata and intent descriptors that the Local AI Agent receives from CoCo.
2. **User Intent and Service Profile:** Reliability TF takes into account the type of service consumed by the user. For services such as video streaming, TF emphasizes metrics such as frame delivery ratio, bit rate stability, and stall probability. For services with an increased risk of attacks or OOD behaviors (e.g., cloud-native pipelines, multi-tenant environments), TF enables the High LoTw flavor, leveraging the transformer-based model for multi-layer

anomaly detection. Thus, TF dynamically adjusts flavor selection, ensuring that credibility aligns with the user's true intent.

3. **User Device Capabilities and Edge Proximity:** Reliability is also influenced by the user's device characteristics and proximity to the edge. Users with low-power devices or with a greater distance from the edge may experience increased QoE variability. TF takes these conditions into account and adjusts the reliability assessment, selecting models that can operate effectively under constraints (e.g., Low LoTw flavor with RF). In contrast, users with high-power devices and low latency edge proximity can benefit from more complex models (Medium or High LoTw), which offer higher accuracy and the ability to detect anomalies.

This approach enhances predictability, reduces the likelihood of QoE degradation, and enables SAFE-6G to implement truly personalized reliability in distributed 6G environments.

6.2 VALIDATION EXPLOITING AN OPENLY ACCESSIBLE DATASET

Before evaluating the Reliability TF on project-generated data, two external openly accessible datasets were employed for methodological benchmarking. The concrete problem addressed in this subsection is the need to verify, under controlled and reproducible conditions, that the modelling paradigms selected for the Reliability TF are scientifically sound. In particular, two complementary questions were examined: (i) whether user-specific attributes exert a measurable influence on QoE-related metrics, thereby justifying their inclusion in the LoTw calculation, and (ii) whether LLM techniques can improve the detection of previously unseen (OOD) attacks that threaten service reliability. The approach consisted in applying statistical correlation analysis on the Simu5G dataset [9] and a leave-one-attack-out protocol on the 5G-NIDD dataset [10], comparing classical and LLM-based pipelines under identical feature-selection and evaluation conditions. The results confirmed both hypotheses and provided quantitative evidence that the chosen techniques are transferable to the SAFE-6G multi-layer setting.

6.2.1 SIMU5G-BASED QoE DATASET

A fundamental premise of the user-centric Reliability TF is that reliability cannot be assessed solely from network-centric metrics. Instead, it must incorporate the unique characteristics of each user (location, velocity, device capabilities, service preferences) because these attributes directly influence radio-level QoS indicators and, through them, the perceived QoE. If this premise is not quantitatively supported, the inclusion of user metadata in the LoTw calculation remains unjustified.

6.2.1.1 EVALUATION OF THE NEED FOR A USER-CENTRIC APPROACH

To quantify the impact of user-specific attributes on QoE, the openly available Simu5G dataset was analysed. The dataset contains fourteen metrics collected from fifteen user equipments across twenty-four sessions and includes both classical radio-level indicators (CQI, SINR, end-to-end delay, RTP inter-arrival time) and user-unique attributes (position, velocity). The analysis focused on the relationship between the application-layer metric “frames displayed” (the percentage of video frames arriving on time – a direct proxy of user-perceived quality) and four network/QoS metrics: average

downlink CQI (averageCqiDL), received downlink SINR (rcvdSinrDL), RTP inter-arrival time (interArrivalTimeRtp) and end-to-end delay (end2EndDelay) [3].

6.2.1.2 DATA ALIGNMENT AND CORRELATION METHODOLOGY

The different attributes in the dataset were neither collected at the same time nor with the same sampling rate. Consequently, a careful alignment procedure was applied for every pair of metrics. The two-timestamp series were first merged into a common timeline; missing values created by the non-overlapping timestamps were filled by forward propagation of the next valid observation. Once the series were of equal length, the Pearson correlation coefficient was computed according to

$$r = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2}}$$

where x_i and y_i are the paired samples of the two metrics and N is the number of aligned samples. The Pearson coefficient was selected because the metrics are continuous, approximately normally distributed and typically exhibit linear relationships; Spearman or Kendall rank correlations could also be applied but were not required for the present analysis. The procedure was repeated for all twenty-four sessions of each UE and the resulting coefficients were averaged to obtain a single representative value per UE and per metric pair [3].

6.2.1.3 RESULTS OF THE CORRELATION ANALYSIS

Once the series were of equal length, the Pearson correlation coefficient was computed for every session and then averaged across the twenty-four sessions of each UE. The Pearson coefficient was selected because the metrics are continuous, approximately normally distributed and typically exhibit linear relationships. The results, as illustrated in Figure 9, revealed a strong positive correlation between “frames displayed” and both average CQI and received SINR, together with a strong negative correlation with end-to-end delay and RTP inter-arrival time. The correlation coefficients remained consistent across almost all fifteen UEs, with only a few outliers. A representative session (UE #0) further illustrated that periods of high CQI/SINR and low delay coincide with near-100 % frames-displayed values Figure 10 [3], confirming that user-centric radio conditions directly influence the QoE metric that the Reliability TF aims to protect.

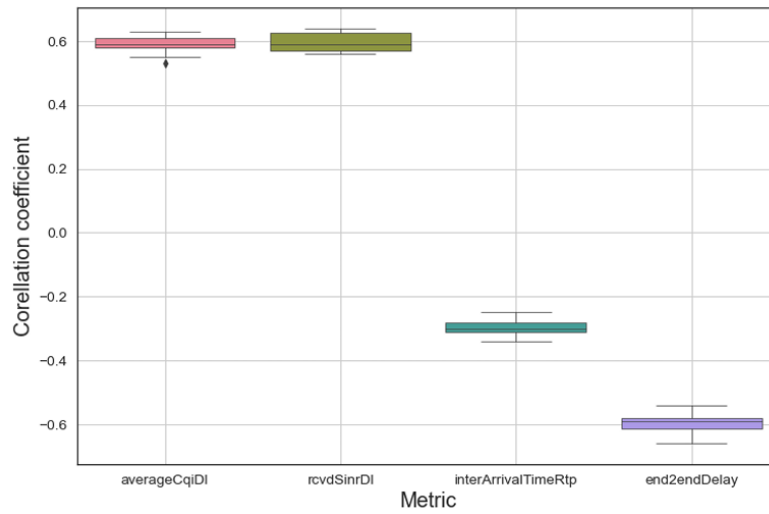


Figure 9: Correlation coefficients between “frames displayed” and the four QoS metrics across the 15 UEs [3]

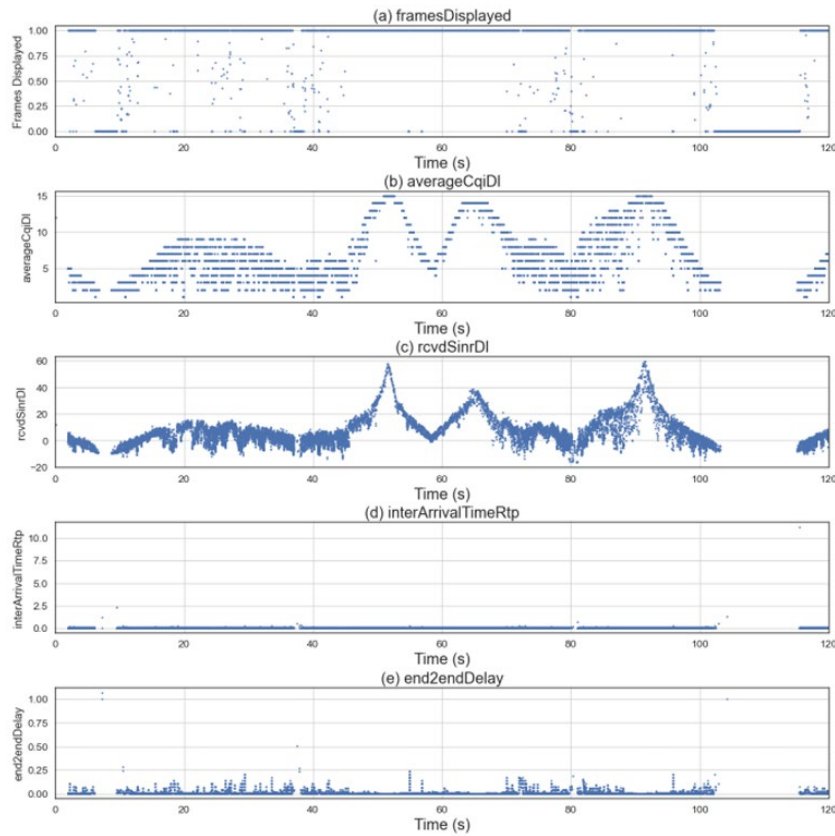


Figure 10: Representative session showing the temporal evolution of frames displayed, CQI, SINR and end-to-end delay [3]

6.2.1.4 IMPLICATION FOR THE RELIABILITY TF

The quantitative evidence obtained from the Simu5G dataset [9] demonstrates that user-specific attributes must be treated as first-class inputs inside the Reliability TF. They directly motivated the design decision to feed user metadata and intent descriptors from the CoCo into the Local AI Agent’s

LoTw-driven flavour-selection logic, thereby ensuring that reliability assessment and enforcement are truly user-centric rather than purely network-centric.

6.2.2 5G-NIDD DATASET FOR OUT-OF-DISTRIBUTION (OOD) ATTACK DETECTION

High-LoTw scenarios require the ability to detect previously unseen (zero-day / OOD) attacks. Classical supervised models trained on a fixed set of attack types typically fail when a novel attack appears. The Reliability TF therefore needs an OOD-capable detector for its High-LoTw flavour. The publicly available 5G-NIDD [10] dataset was selected for this purpose because it is a well-known, labelled dataset generated from a real 5G testbed at the University of Oulu and is suitable for AI/ML training.

6.2.2.1 DATASET DESCRIPTION AND ATTACK TYPES

The dataset contains 477,737 benign samples and 738,153 malicious samples belonging to eight distinct intrusion types: UDP Flood, HTTP Flood, Slowrate DoS, TCP Connect Scan, SYN Scan, UDP Scan, SYN Flood and ICMP Flood. After removal of the 5G-specific GTP encapsulation layer, each sample is described by 51 statistical and protocol-level features (e.g., Seq, sHops, Offset, dMeanPktSz, TcpRtt, AckDat, Proto, dDSb, State). The distribution of attack instances is highly imbalanced: UDP Flood alone accounts for 457,340 samples (37.61 % of the total), while ICMP Flood represents only 1,155 samples (0.09 %). This 396:1 ratio motivated the construction of balanced training sets with an equal number of samples per attack class. Table 3 shows the number of samples for each Attack Type, indicating a large disparity between the sizes of the different attack types [1].

Class	Number of samples
Benign	477,737
UDP Flood	457,340
HTTP Flood	140,812
Slowrate DoS	73,124
TCP Connect Scan	20,052
SYN Scan	20,043
UDP Scan	15,906
SYN Flood	9,721
ICMP Flood	1,155

Table 3: Number of samples for each class [1]

The eight attack types can be briefly characterised as follows. UDP Flood overwhelms the target by sending a very large number of UDP packets at high speed, exploiting the connectionless nature of the protocol. HTTP Flood is an application-layer attack that transmits HTTP requests at high rates toward resource-intensive endpoints. TCP SYN Flood exploits the three-way handshake by sending numerous SYN packets without completing the connections, thereby exhausting server resources. Slowrate DoS opens multiple slow-rate connections that remain open for prolonged periods. ICMP Flood sends large volumes of ICMP echo requests, forcing the target to reply and consuming bandwidth. TCP Connect Scan completes full TCP handshakes to identify open ports, while SYN Scan and UDP Scan probe ports with incomplete or protocol-specific packets [1].

6.2.2.2 CLASS SIMILARITY ANALYSIS

To quantify how challenging each attack is to detect, a class-similarity analysis was performed. For each class, summary statistics (min, max, median, standard deviation and quartiles) were computed on the most important features selected by ANOVA. Figure 11 illustrates the Euclidean distances that were then calculated between the median feature vectors of every pair of classes and between each attack class and the benign class. The closest pairs of attacks are HTTPFlood-SlowrateDoS (1.58), HTTPFlood-SynFlood (1.94), and SlowrateDoS-SynFlood (1.94), showing that these attacks share highly similar behavioral profiles. Another relatively close pair is SYNScan-UDPScan (2.39). In contrast, the most distant pair is TCPConnectScan-Benign (6.17), followed by SYNScan-Benign (5.47) and ICMPFlood-Benign (5.28). Based on these distances, we can group attacks into clusters that exhibit comparable network signatures, meaning they appear close to each other in the feature space [65]. One such cluster includes Flood-type DDoS attacks (ICMPFlood, SynFlood, HTTPFlood, SlowrateDoS), all of which have pairwise distances below 3. Similarly, Scan-type attacks (SYNScan, UDPScan) form another cluster with a distance of 2.39, clearly separated from the flood cluster. This separation is expected: flood attacks generate high-volume packet traffic, whereas scan attacks produce lower, more sporadic packet flows [10], [66]. Another notable observation is that TCPConnectScan stands out as the most distinct attack, being significantly distant from all other categories. This suggests that its behavior differs substantially from typical flood or scan patterns. Regarding the distance from benign traffic, the attack closest to the benign class is generally the hardest to detect. UDPFlood has the smallest distance (3.68), making it the most challenging to identify, followed by HTTPFlood (4.45) and SYN Flood (5.02). These results align with domain knowledge: HTTPFlood relies on legitimate-looking HTTP requests that resemble normal web traffic, while UDPFlood sends valid UDP packets. Since neither attack involves abnormal payloads or protocol violations, their traffic can easily blend with benign network activity [1].

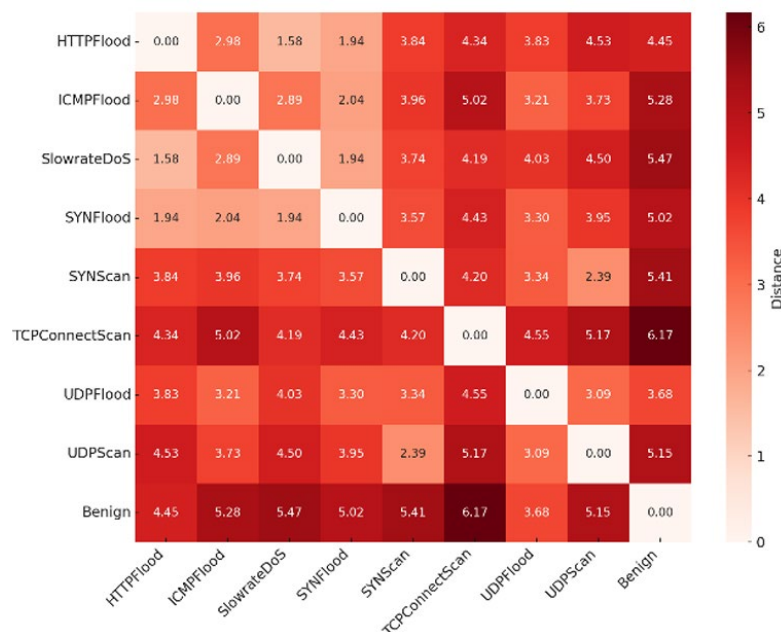


Figure 11: Euclidean distances between attack classes and the benign class (class-similarity matrix) [1]

6.2.2.3 TRAIN/VALIDATION/TEST SPLITTING

A strict leave-one-attack-out protocol was adopted. For a chosen dataset size n , nine equally sized slots were created (one per class). Each slot received $n/9$ samples from its class; the benign class contributed twice as many samples so that half could be reserved for the test set. When an attack class contained fewer samples than required (e.g., ICMP Flood), random oversampling with replacement was applied. For each run, one attack type was completely excluded from the training pool and placed, together with an equal number of benign samples, into the test set. The remaining seven attack classes plus the residual benign samples formed the training pool, which was further split into training and validation subsets with an approximate 75/25 ratio in a stratified manner. This procedure was repeated for every attack type and for three dataset sizes (10,000, 25,000 and 100,000 samples) [1].

6.2.2.4 DATA PREPROCESSING AND FEATURE SELECTION

During preprocessing the columns ‘Attack Type’ and ‘Label’, as well as any constant or null columns, were discarded. Missing numerical values were imputed with the mean computed exclusively on the training portion of the current fold; the same imputation parameters were then applied unchanged to the validation and test portions. All numerical features were z-normalised using training-set statistics, and categorical features were one-hot encoded [1].

Feature selection followed a three-stage supervised procedure. First, groups of highly correlated features (Pearson correlation > 0.9) were identified and only the member with the highest ANOVA F-score with respect to the binary label was retained. Second, the top-10 features ranked by ANOVA F-score were kept. Third, whenever a selected feature originated from one-hot encoding, all sibling one-hot columns belonging to the same original categorical variable were restored in order to preserve interpretability. The resulting feature set constituted the tabular input for LightGBM and the basis for the textual representations used by the LLM pipelines [1].

6.2.2.5 INPUT REPRESENTATIONS AND MODEL PIPELINES

Two complementary input representations were constructed from the same preprocessed records. For LightGBM a fixed-length numerical feature vector was formed from the selected numerical and one-hot features. For the LLM-based models each sample was converted into a human-readable textual string of feature-value pairs (e.g., “Proto:udp Seq:10 sTtl:64 ...”) [1]. Four models were compared:

a) LightGBM (Baseline ML Method)

As a baseline model, we employ LightGBM [67], a widely used gradient-boosting decision tree algorithm designed for structured tabular data. To ensure a fair comparison, we replicate the methodology of 5G-NIDD [10] and successfully reproduce its reported accuracy on the same dataset. Training uses early stopping after 50 rounds without improvement in validation loss, with *binary_logloss* (binary cross-entropy) [68] and binary error rate [69] as evaluation metrics. The pipeline shown in Figure 12 begins with raw data ingestion, followed by preprocessing, ANOVA-based feature selection, and construction of the final tabular dataset. This processed dataset is then fed into

the LightGBM classifier for binary classification. After training, the model predicts whether each test instance is benign or malicious, and these predictions are compared with ground-truth labels. Training settings include: binary classification objective, GBDT boosting, 64 leaves, maximum depth of 7, learning rate of 0.05, minimum child samples of 30, and enabled class-imbalance handling. All other parameters remain at their default values [1].

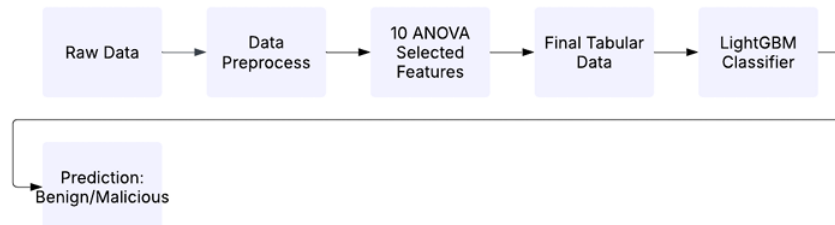


Figure 12: LightGBM model pipeline [1]

b) Untuned Sentence Transformer Embeddings with RF classifier

The first LLM-based approach uses a pre-trained sentence transformer (MiniLM-L6-v2, 22M parameters) [70], [71] without any fine-tuning; the model is applied directly with its original weights. As illustrated in Figure 13, after preprocessing and feature selection, the tabular samples are converted into textual descriptions. These text inputs are tokenized and encoded by the untuned SentenceTransformer, producing 384-dimensional mean-pooled embeddings. These embeddings act as high-level feature representations and are passed to a RF classifier, which outputs binary predictions (“0” for benign, “1” for malicious). The configuration includes: encoder *all-MiniLM-L6-v2*, encoding batch size 64, RandomForestClassifier with 600 trees and *random_state=42*, and test batch size 64 [1].

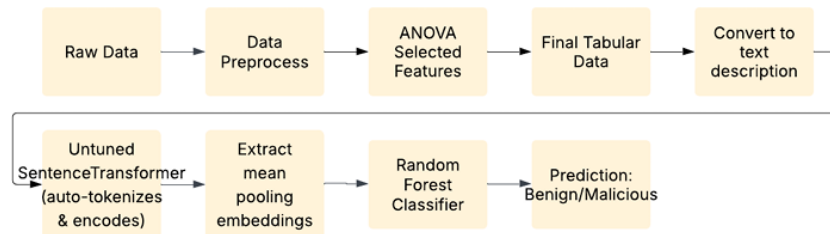


Figure 13: Untuned sentence transformer embeddings with RF model pipeline [1]

c) Fine-Tuned LLM with Classification Head

The second LLM approach fine-tunes the same MiniLM model using HuggingFace’s AutoModel for sequence classification [70], [71]. After preprocessing and feature selection, tabular samples are transformed into textual descriptions and tokenized using AutoTokenizer with a maximum length of 128 tokens. During training, both the encoder and the classification head are updated using the HuggingFace Trainer to classify samples as benign or malicious (Figure 14). Key parameters include: AutoTokenizer (*max_length=128*, *truncation=True*), batch size 16, 10 training epochs, evaluation and checkpointing at each epoch, loading the best model based on *eval_loss*, logging every 50 steps, *save_total_limit=1*, *seed=42*, AdamW optimizer with learning rate 5×10^{-5} , and EarlyStoppingCallback with *patience=2*. Testing uses batch size 32. Training is accelerated using a GPU [1].

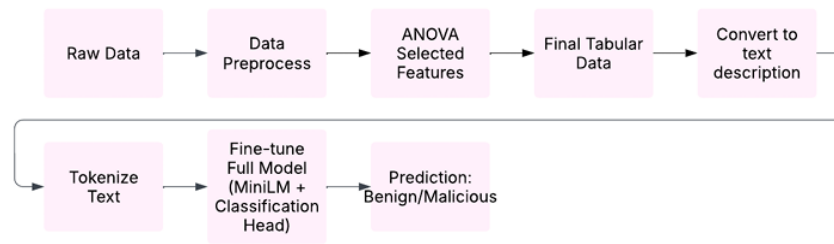


Figure 14: Fine-Tuned LLM with classifier head model pipeline [1]

d) Fine-tuned Sentence Transformer Embeddings with RF classifier

In the third LLM configuration, the preprocessed text is tokenized and fed into the MiniLM encoder for fine-tuning [70], [71]. After training, the classification head is removed, and only the fine-tuned encoder is retained to generate new embeddings. Each sample is tokenized again and passed through the encoder to obtain a fixed 384-dimensional mean-pooled representation. These embeddings are then used as input to a RF classifier trained to distinguish benign from malicious samples (Figure 15). The setup uses the best fine-tuned encoder (*finetuned_clf.base_model*), `token_max_length=128`, batch size 32, and a `RandomForestClassifier` with 600 trees and `random_state=42` [1].

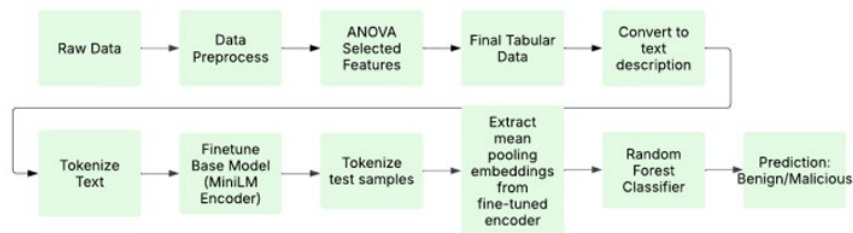


Figure 15: Fine-Tuned LLM embeddings with RF classifier [1]

6.2.2.6 EVALUATION METRICS AND RESULTS

Performance was measured by accuracy, precision, malicious (class-specific) recall, macro-recall and F1-score. Particular emphasis was placed on malicious recall, because the ability to detect previously unseen attacks is the primary requirement of the High-LoTw flavour.

Figure 16 shows that every LLM-based approach achieves higher malicious recall than LightGBM, meaning they correctly identify a larger share of attacks – although none of the LLMs clearly dominates the others. Interestingly, the untuned LLM slightly surpasses the fine-tuned variants. Fine-tuning may cause the model to overfit to the attacks seen during training, making it harder to distinguish malicious classes that closely resemble benign traffic (e.g., UDPFlood) and potentially causing it to miss attacks that differ substantially from both benign and other malicious patterns when excluded from training, such as ICMPFlood (Figure 11). Additionally, a 600-tree RF trained on frozen embeddings can be more effective when fewer samples are available. Overall, the key takeaway is that LLM-based models can achieve up to roughly 5% higher detection performance on unseen attacks compared to the non-LLM baseline. Still, a malicious recall of 76.3% indicates that there is room for further improvement [1].

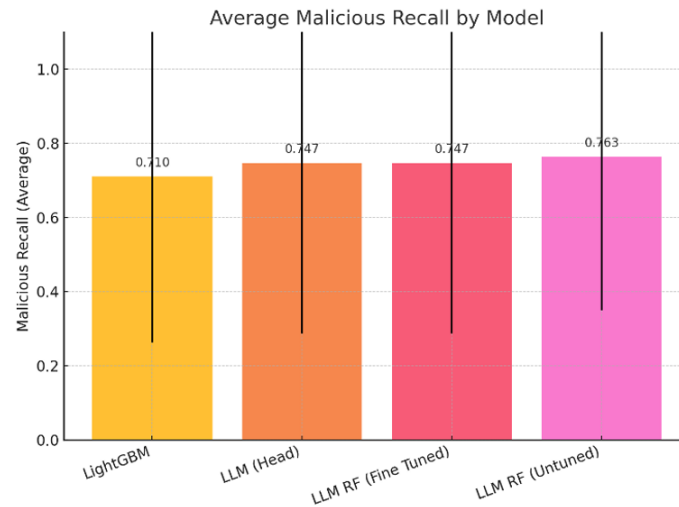


Figure 16: Average malicious recall across LLM-based models [1]

Figure 17 and Figure 18 provide a more detailed breakdown of performance across all attack types and different input sizes. In each heatmap, rows correspond to the attack type excluded during training, while columns represent the four models (LightGBM, untuned LLM with RF, fine-tuned LLM classifier, and fine-tuned LLM with RF). Lighter colors indicate higher malicious recall or accuracy. These figures show that input size only slightly affects the results. Moreover, HTTPFlood, SYNScan, SlowRateDoS, and UDPScan are consistently the easiest attacks to detect, with nearly perfect performance across all models, likely due to their high similarity, as shown in Figure 11 [1].

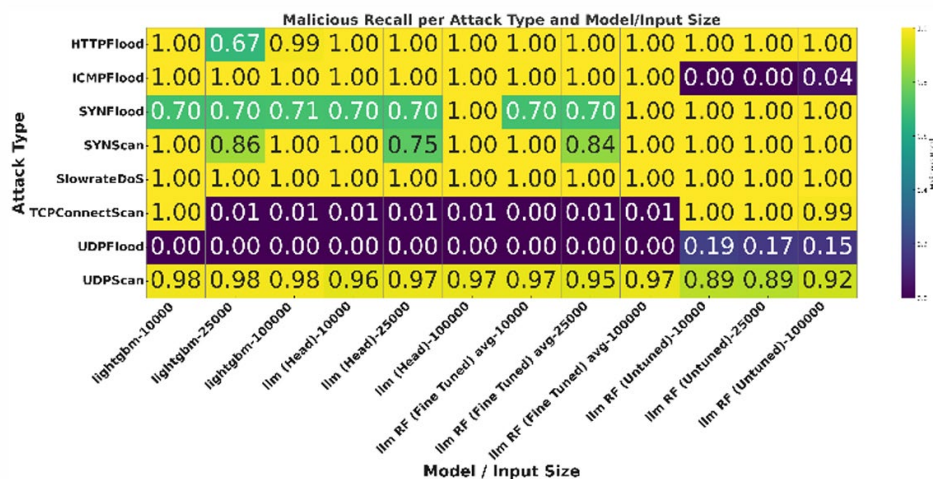


Figure 17: Malicious recall per attack type, per model and per input size [1]

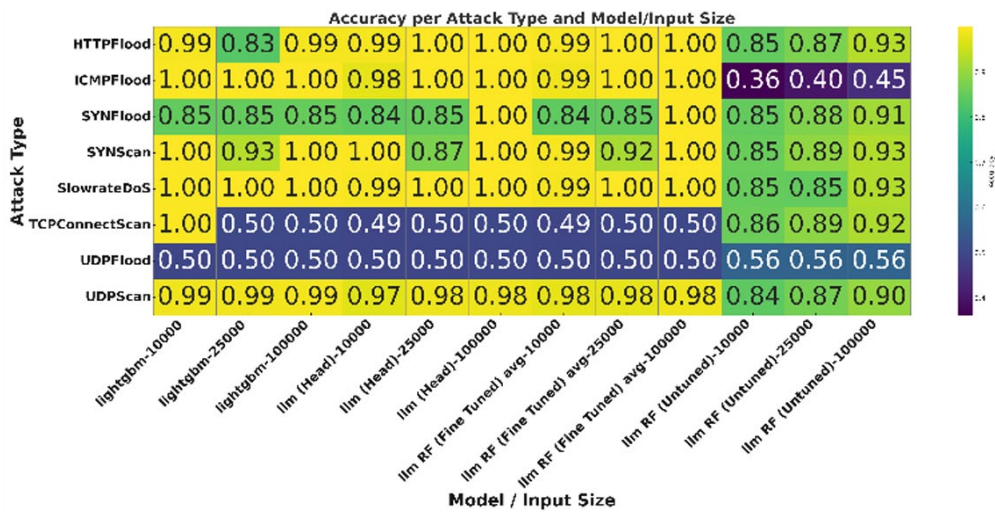


Figure 18: Average accuracy per attack type, per model and per input size [1]

The untuned LLM with RF achieves almost perfect malicious recall for TCPConnectScan and SYN Flood, outperforming the other models, but it completely fails to detect ICMP Flood. UDP Flood remains the hardest attack to identify: most models misclassify it entirely as benign, with the untuned LLM with RF reaching only about 17% malicious recall. This aligns with Figure 11, where UDP Flood appears most similar to benign traffic. Finally, although the untuned LLM with RF achieves the highest malicious recall overall, the other three models deliver better average accuracy, about 15% higher, indicating slightly stronger performance on benign samples [1].

6.2.2.7 IMPLICATION FOR THE RELIABILITY TF

These findings demonstrate that converting network-traffic samples into textual feature-value sequences and leveraging pre-trained language models yields a measurable improvement in the detection of previously unseen attacks. Consequently, LLM-based OOD detection is adopted as one of the two complementary capabilities of the High-LoTw flavour of the Reliability TF.

6.3 VALIDATION EXPLOITING SAFE-6G DATASET

The core experimental validation of the Reliability TF was performed on the two multi-layer datasets that were generated and released within the project. The concrete problem addressed in this subsection is the absence, at the start of the task, of realistic, labelled multi-layer telemetry that simultaneously captures application-layer QoE degradation, container/infrastructure resource pressure and network-flow behaviour under both benign high-load and adversarial conditions. Without such data it is impossible to train and rigorously evaluate the Low-, Medium- and High-LoTw AI/ML models required by the Reliability TF. The approach consisted in instrumenting a cloud-native MediaMTX video-streaming platform with a multi-layer monitoring stack, generating two complementary labelled datasets [7], [8], and training a progressive suite of models whose complexity matches the three LoTw ranges. The results show that the developed models achieve high accuracy, excellent recall of critical failure events and a substantial improvement in attack detection over strong tabular baselines, thereby fulfilling the proactive reliability-management and high-LoTw anomaly-detection objectives of Task 4.6.

6.3.1 MULTI-LAYER MONITORING AND PROACTIVE SCALING DATASET

The first dataset [7] addresses the concrete operational problem of predicting the transition from a reliable to an unreliable service state before a major failure occurs. In resource-constrained edge-cloud video-streaming environments, memory pressure is a dominant precursor of QoE degradation and service interruption. The Reliability TF must therefore be able to detect the onset of critical failures early enough to trigger proactive vertical or horizontal scaling.

6.3.1.1 EXPERIMENTAL SETUP AND ARCHITECTURE

A MediaMTX-based cloud-native streaming platform was instrumented with Prometheus, Thanos and Kepler, using the RTSP. The architecture consists of a Publisher, which acts as the source of the media stream, the Main Server which manages stream registration, routing logic, and coordination of Publishers and Streamers. The Streamers receive media streams from the Publisher and broadcast them to the End Clients, which consume the media content through the RTSP connections.

At the infrastructure and container layers, the monitoring system records CPU utilisation, memory usage, network throughput and energy consumption. At the application level MediaMTX exposes protocol-aware metrics, including the number of active RTSP sessions, bytes transmitted and received per session, frame-related indicators and failure counters [4], [6]. All metrics are presented on Table 2 and are exported as timestamped feature vectors in JSON format, enabling seamless integration with the AI/ML pipelines of the Reliability TF.

To evaluate performance across heterogeneous environments, the End Clients were deployed as three distinct instances across different environments, a Google Colab notebook, a virtual machine, and a physical machine as shown in Figure 19 [4], [6].

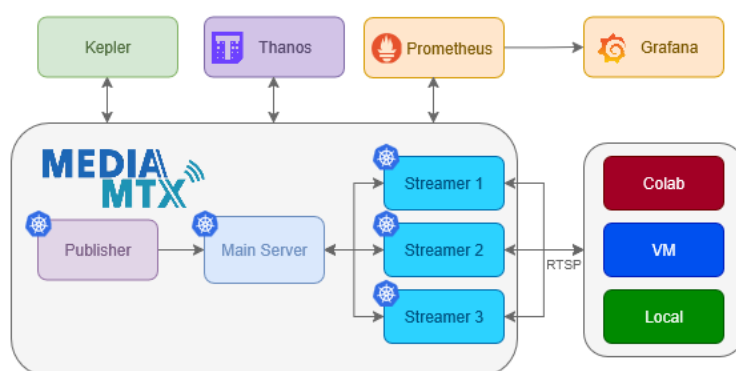


Figure 19: MediaMTX architecture [6]

6.3.1.2 VALIDATION SCENARIOS

Six experimental scenarios were defined by combining two memory allocations per streamer (100 MiB and 200 MiB) with one, two or three concurrent streamers. Client load was increased progressively up to thirty concurrent sessions per streamer instance, distributed across three independent execution environments (Google Colab, virtual machine and local physical machine) (see Figure 19: *MediaMTX* architecture [6] [6]). For each scenario the monitoring stack produced a labelled time series

in which every sample is annotated with a binary reliability label: the label equals 1 when at least one stream experiences a critical failure or severe QoE degradation (loss of active RTSP sessions or discontinuity in frame delivery) within the sampling interval, and 0 otherwise. As shown in Table 4, the resulting datasets exhibit moderate to strong class imbalance, with unreliable states representing between 8.1% and 27.0% of the samples, a realistic reflection of operational conditions in which major failures are relatively rare. In addition, the end clients connected to different streamers consumed video content of heterogeneous quality (distinct resolutions and bitrates), thereby emulating realistic user-centric demand diversity [4].

Scenario	Reliable, n (%)	Unreliable, n (%)	Total Samples, n
1 streamer – 100 MiB	759 (83.0)	156 (17.0)	915
2 streamers – 100 MiB	688 (73.0)	255 (27.0)	943
3 streamers – 100 MiB	716 (75.4)	234 (24.6)	950
1 streamer – 200 MiB	781 (87.0)	117 (13.0)	898
2 streamers – 200 MiB	792 (87.1)	117 (12.9)	909
3 streamers – 200 MiB	881 (91.9)	78 (8.1)	959

Table 4: Class distribution across the six validation scenarios. Values are expressed as the number of samples, with percentages reported in parentheses [4]

6.3.1.3 ML-BASED ANALYSIS - DATA PREPROCESSING AND FEATURE EXTRACTION

A sliding-window scheme of forty consecutive samples was applied to capture the temporal evolution of resource pressure. For each window fourteen statistical descriptors of memory utilisation (mean, standard deviation, minimum, maximum, skewness, kurtosis and selected percentiles) were computed and subsequently Min-Max scaled to the range [0, 1]. The binary reliability label of the window was determined by the presence of any failed stream inside that window. Although the monitoring framework provides a rich set of multi-layer metrics, memory utilisation was deliberately selected as the primary feature because memory exhaustion was empirically observed to be the dominant precursor of service degradation in the evaluated scenarios. The framework itself remains agnostic to the concrete metric and can incorporate additional network or RAN-level telemetry in future deployments [4].

6.3.1.4 CROSS-SCENARIO CROSS-VALIDATION STRATEGY AND MODELS

A rigorous leave-one-scenario-out was adopted: each of the six scenario files served once as the held-out test set, while the remaining files were used for training and validation. This design guarantees that every combination of memory allocation and number of streamers is observed both during training and during evaluation, but never simultaneously in the same role, thereby testing the ability of the models to generalise to unseen operating conditions. The five models were trained and compared, such as Support Vector Classifier (SVC), Decision Tree Classifier (DT), RF, k-NN, the DNN consisting of three dense hidden layers of 16 neurons with ReLU activations, dropout of 0.1, Adam optimiser and early stopping. Hyper-parameters of each model were activated on the validation fold by maximising the weighted F1-score and presented in Table 5 [4]. The DNN architecture and training procedure were chosen to balance predictive power with the computational constraints expected for a Medium-LoTw vApp.

Model	Hyperparameters
SVC	kernel = poly, gamma = scale, C = 3
DT	min-samples-split = 10, min-samples-leaf = 5, max-depth = 7
RF	n-estimators = 10, min-samples-split = 2, min-samples-leaf = 1, max-depth = 15
k-NN	weights = distance, n-neighbors = 7, metric = manhattan
DNN	hidden-layers = 3, units = 16, dropout-rate = 0.1, learning-rate = 0.001, batch-size = 32, epochs = 100

Table 5: Optimal hyperparameters for each ML model and scenario [4]

6.3.1.5 RESULTS

As illustrated in Figure 20, the DNN consistently achieved accuracy above 97% across all six scenarios, while the RF baseline remained above 85%. More importantly, the DNN exhibited the highest recall for the unreliable class and the lowest number of false negatives, a critical property for proactive reliability management, where missing an imminent degradation event is far more costly than a false alarm [4].

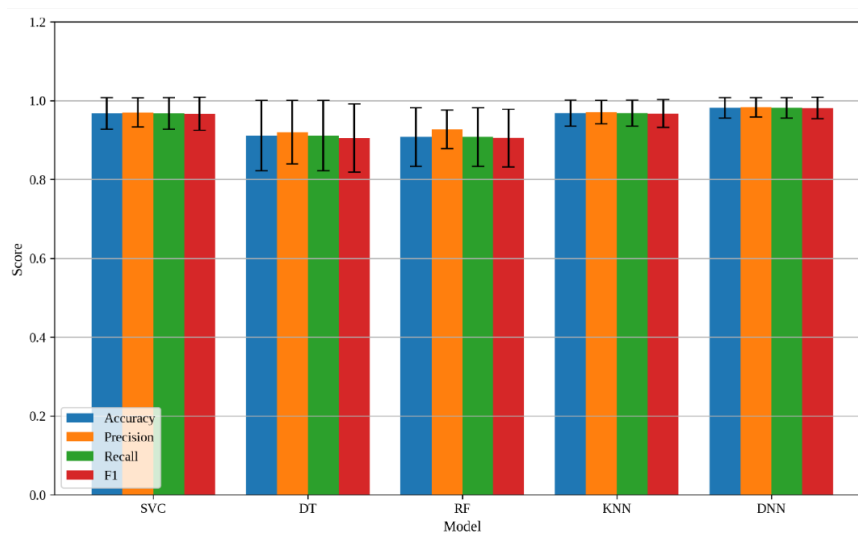


Figure 20: Bar chart comparing accuracy, precision, recall, and F1 score per model [4]

As illustrated in Figure 21, Receiver-Operating-Characteristic (ROC) and Precision-Recall (PRC) analysis yielded an AUROC of 0.9965 and an AUPRC of 0.9841 (far above the prevalence baseline of 0.1717), confirming excellent separability even under class imbalance. Expected Calibration Error averaged 0.029, indicating that the predicted probabilities are reliable enough to be consumed by the Local AI Agent [4].

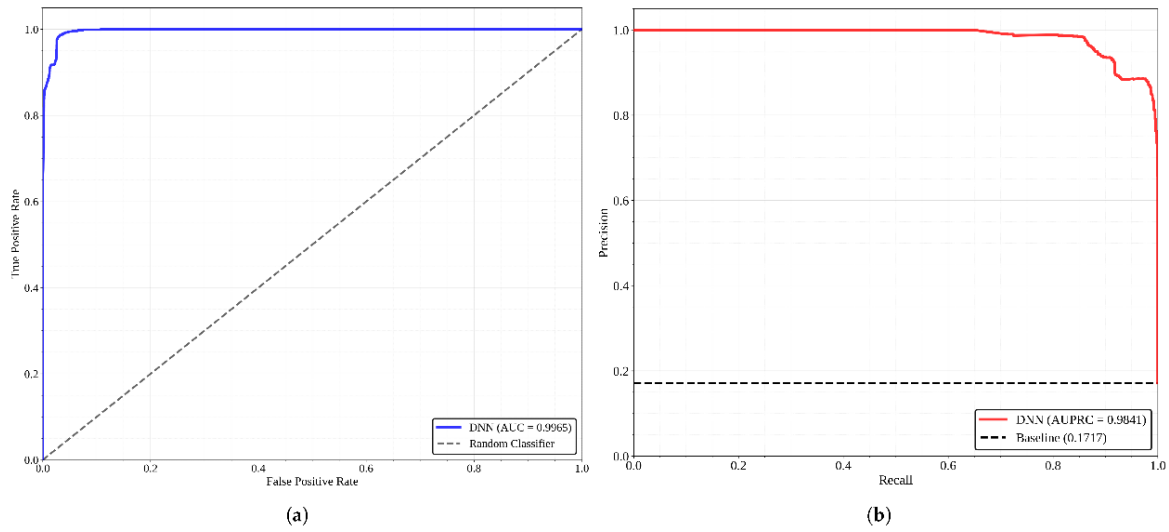


Figure 21: (a) ROC and (b) Precision-Recall curves of the DNN, aggregated across all six scenarios [4]

As illustrated in Figure 22, visual inspection of the predicted versus actual labels, plotted together with the average memory-usage ratio for each scenario, showed that the DNN correctly identifies the first transition into the unreliable state. Consequently, the model provides a practical estimate of the TTFMF at window granularity, enabling the Reliability TF to act before the end user experiences a visible QoE degradation. The DNN neither delays the detection nor exhibits premature triggering. Instead, it aligns almost perfectly with the actual onset of reliability degradation. Although the reliability label corresponds to the service state at the window endpoint, the temporal analysis shows that the model identifies the transition into the unreliable state at its earliest manifestation in the monitoring data. As a result, the effective forecast horizon can be interpreted as the number of windows by which degradation precursors are detected before a fully manifested failure. In the evaluated scenarios, this corresponds to a minimum lead time of one window before the first failure event. As such, the DNN not only classifies reliability states accurately, but also offers early situational awareness, enabling the reliability function to predict and mitigate upcoming QoE degradation. From a user-centric perspective, the ability to detect degradation early translates directly into maintaining perceived service continuity, particularly for users who consume higher quality video with a lower tolerance for interruptions [4].

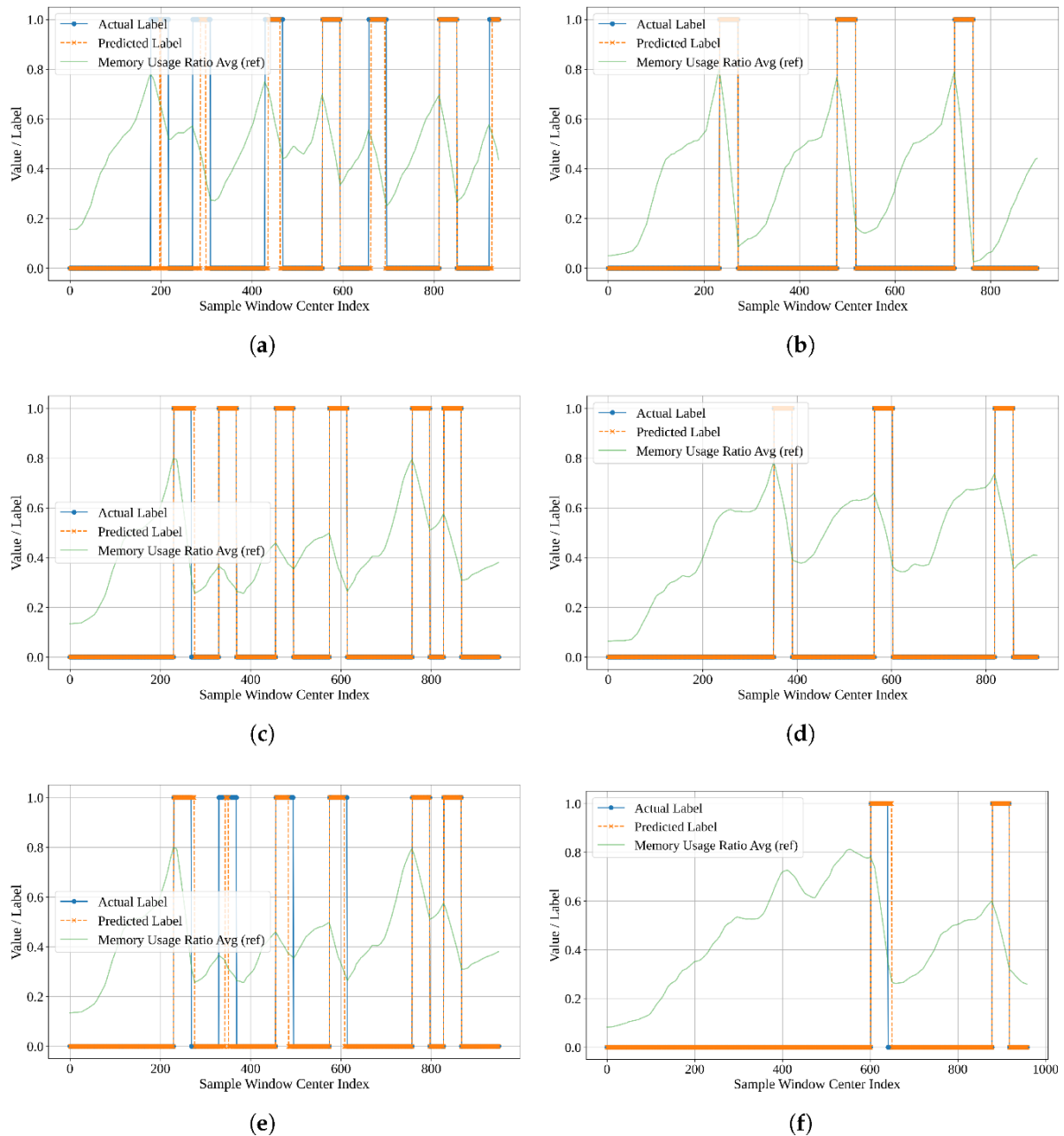


Figure 22: Per-scenario plots of predicted versus actual reliability labels together with average memory-usage ratio (a) One streamer with 100 MiB memory constraints. (b) One streamer with 200 MiB memory constraints. (c) Two streamers with 100 MiB memory constraints. (d) Two streamers with 200 MiB memory constraints. (e) Three streamers with 100 MiB memory constraints. (f) Three streamers with 200 MiB memory constraints [4]

These results demonstrate that multi-layer telemetry is sufficient to support accurate, early prediction of reliability degradation and that the Low-LoTw (RF) and Medium-LoTw (DNN) flavours of the Reliability TF can be deployed with confidence for proactive scaling decisions. The labelled failure events contained in the dataset constitute concrete evidence that the models solve the operational problem of anticipating major service interruptions before they become visible to the end user.

6.3.2 TRANSFORMER-ASSISTED MULTI-LAYER ATTACK DETECTION DATASET

The second dataset addresses the High-LoTw requirement of detecting complex multi-layer attacks that may remain invisible when only a single monitoring plane is examined. Approximately 210,000 fused telemetry records were collected and used to train and evaluate three complementary detection pipelines.

6.3.2.1 SYSTEM ARCHITECTURE AND DATA COLLECTION

The experimental platform is a cloud-native RTSP streaming system built on MediaMTX and deployed as containerised microservices, as shown in Figure 23. The testbed comprises the following components: Publisher, which receives video content from an external source and publishes it to the MediaMTX infrastructure; Main Server, which is central control and configuration node responsible for stream registration, authentication and routing logic; Streamers, which are distributed nodes that pull media streams from the publisher and redistribute them to clients; they can be horizontally scaled; Clients, who are legitimate consumers that request and play back media content over RTSP; Attacker, who is a malicious host that attempts either to exhaust networking and processing resources or to obtain unauthorised access by compromising the authorisation API [2].

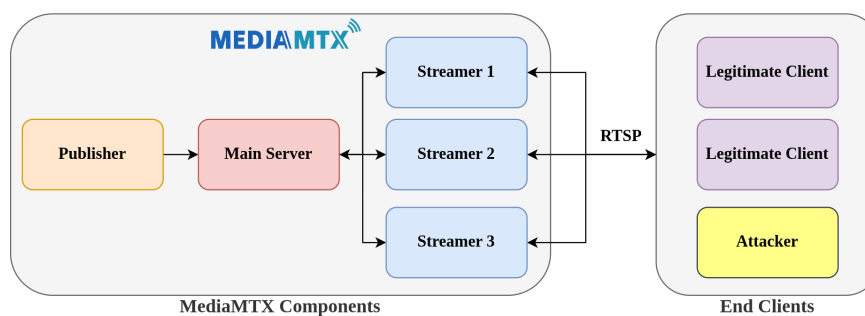


Figure 23: MediaMTX architecture for attack detection [2]

Normal operation involves approximately 10-15 concurrent client connections. A stressed-benign scenario increases the number of concurrent connections to more than 70 while preserving legitimate client behaviour. Four malicious scenarios are generated separately [2]:

1. **SYN attack** – high-rate stream of TCP SYN segments aiming to exhaust the server's connection backlog;
2. **RTSP session attack** – repeated establishment of a large number of full RTSP sessions, stressing session management and authentication;
3. **Slow RTSP (Slow-Read) attack** – numerous concurrent RTSP connections that are deliberately kept idle or streaming at a minimal rate;
4. **Brute-force attack** – high volume of authentication requests against the MediaMTX authorisation API.

Multi-layer telemetry is collected from three synchronised planes, as shown in Figure 24. At the network layer, traffic passes through a pfSense gateway on which softflowd generates NetFlow/IPFIX records that are forwarded to Akvorado; Akvorado relies on Apache Kafka for streaming transport and

ClickHouse for persistent storage. Each IPFIX record contains the first-packet timestamp, source and destination addresses and ports, transport protocol, byte and packet counts, TCP flags and IP TTL. At the continuum layer, Prometheus scrapes per-container CPU utilisation, memory usage, peak working-set size, OOM failure events and container-restart counts. At the application layer, MediaMTX exposes the number of active RTSP connections, bytes transmitted per session and RTSP failure counters. Each IPFIX flow is fused with the contemporaneous application and continuum measurements using its first-packet timestamp and a bounded 5-second alignment window [2]. All metrics are presented in Table 2.

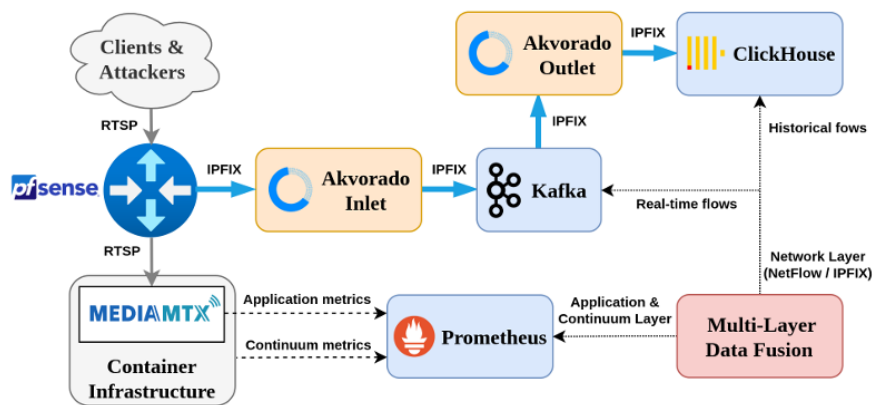


Figure 24: MediaMTX architecture for multi-layer attack detection [2]

6.3.2.2 METHODOLOGY

Three detection pipelines were compared under identical telemetry and chronological split: a classical RF operating on the sanitised tabular feature vector (RF-Tabular), an end-to-end fine-tuned DistilBERT model with a native classification head (DistilBERT-Head), and a hybrid pipeline that uses the fine-tuned DistilBERT encoder as a frozen embedding extractor followed by a RF classifier (DistilBERT-RF). The experiment was executed on three different days with slight variations in benign background traffic and attack rates, yielding approximately 210,000 telemetry records. The first execution was used exclusively for training, the second for validation and the third for testing. This chronological, execution-level split prevents temporally adjacent observations from appearing in multiple subsets and provides a more realistic estimate of future-traffic generalisation than random sample-level splitting [2].

Each fused record carries four binary attack indicators corresponding to the four attack scenarios. Because the evaluation concerns binary detection, all attack scenarios are aggregated into a single malicious class. Source and destination IP addresses, raw timestamps, session identifiers and connection identifiers are removed before model training in order to prevent the detector from memorising host identities or scenario-specific artefacts. Missing numerical values are imputed with statistics computed solely on the training execution; the same imputation parameters are then applied unchanged to the validation and test executions [2].

Two feature representations are constructed from the same sanitised records. The first is a conventional tabular vector containing the IPFIX, application, container and continuum features. The

second is a deterministic textual serialisation in which each record is converted into an ordered sequence of feature-value pairs (e.g., “proto=tcp src_port=554 dst_port=49152 ...”). The order of fields, feature names, missing-value markers and numerical formatting are fixed across all splits, ensuring that every pipeline receives equivalent information [2].

DistilBERT (approximately 66 million parameters) was selected for three reasons: its bidirectional self-attention can model dependencies among heterogeneous feature-value tokens; its distilled architecture reduces inference cost relative to full BERT; and its encoder-only design produces dense, deterministic embeddings without generative decoding. The model is fine-tuned for four epochs with class-weighted cross-entropy loss (learning rate 2×10^{-5}) on an NVIDIA GeForce RTX 3080 GPU (approximately 15 minutes per run). The validation execution is used for model selection. The selected classification head is evaluated as DistilBERT-Head; the corresponding encoder is frozen and used to extract embeddings for the DistilBERT-RF pipeline. All RF models employ 300 trees, balanced class weights and the run-specific random seed. Training is repeated over nine random seeds; results are reported as mean $\pm$ standard deviation [2].

6.3.2.3 RESULTS

RF-Tabular achieved a solid baseline (accuracy 0.9557 ± 0.0103 , attack recall 0.9088 ± 0.0212). Both transformer-assisted pipelines improved substantially. DistilBERT-RF reached an accuracy of 0.9820 ± 0.0076 and an attack recall of 0.9632 ± 0.0156 . DistilBERT-Head achieved the best overall performance with an accuracy of 0.9854 ± 0.0087 and an attack recall of 0.9703 ± 0.0181 , corresponding to an improvement of 6.15 percentage points in attack recall over the tabular baseline while maintaining near-perfect precision, as shows in Table 6 [2].

The relatively small gap between DistilBERT-Head and DistilBERT-RF indicates that most of the detection benefit is already captured by the fine-tuned embedding space. This offers a flexible deployment option: the encoder can be fine-tuned once and then reused as a frozen embedding extractor, while a lightweight RF performs the final binary decision, a design well suited to edge-cloud Reliability TF deployments.

Model	Accuracy	Macro F1-Score	Attack Precision	Attack Recall	Attack F1-Score
RF-Tabular	0.9557 $\pm$ 0.0103	0.9555 $\pm$ 0.0104	0.9997 $\pm$ 0.0000	0.9088 $\pm$ 0.0212	0.9520 $\pm$ 0.0116
DistilBERT-RF	0.9820 $\pm$ 0.0076	0.9820 $\pm$ 0.0076	0.9997 $\pm$ 0.0000	0.9632 $\pm$ 0.0156	0.9810 $\pm$ 0.0081
DistilBERT-Head	0.9854 $\pm$ 0.0087	0.9854 $\pm$ 0.0088	0.9996 $\pm$ 0.0001	0.9703 $\pm$ 0.0181	0.9847 $\pm$ 0.0093

Table 6: Model Performance Metrics Over Nine Random Seeds (Mean $\pm$ Standard Deviation) [2]

These results, obtained on a dataset developed within Task 4.6 and reported in a paper accepted for presentation at IEEE CloudNet 2026 [2], confirm that transformer-assisted multi-layer fusion provides a robust High-LoTw detection capability that classical tabular models cannot match. Together with the LLM-based OOD detection capability described in Section 6.2.2, the DistilBERT pipelines constitute the High-LoTw flavour of the Reliability TF.

7 FUTURE RESEARCH PATHWAYS

The work conducted in Task 4.6 establishes a complete and functional Reliability TF fully aligned with all related SAFE-6G objectives. Beyond the scope of the project, several promising research directions emerge from the scientific findings and the associated publications. These directions do not reflect open issues or incomplete work within SAFE-6G; instead, they highlight opportunities for advancing user-centric reliability in future 6G systems.

A first research pathway concerns the evolution of **transformer-based multi-layer fusion models**. The results obtained in SAFE-6G demonstrate that transformer encoders can effectively process heterogeneous telemetry records spanning application, network-flow and infrastructure layers [2]. Future work may explore larger foundation models capable of jointly encoding multi-modal reliability signals, enabling more generalisable inference across diverse deployments and dynamic service conditions.

A second direction arises from the findings on **OOD attack detection**. The use of textual feature-value representations and fine-tuned LLMs has shown clear benefits in identifying previously unseen malicious behaviours [1]. Extending this approach to cross-domain generalisation, multilingual traffic representations, and federated fine-tuning could enhance the robustness of reliability mechanisms in multi-operator or multi-tenant environments.

The **proactive scaling dataset** and the associated monitoring framework developed in SAFE-6G [4], [6] also open avenues for further research. Future studies may investigate predictive QoE-driven resource allocation for additional media types, XR workloads, multi-stream scenarios, and adaptive scaling strategies that respond to early indicators of reliability degradation.

Another promising direction relates to **inter-operator interference and external environmental factors**, such as RIS-mediated reflections, which have been shown to degrade SINR, CQI and QoE in multi-tenant deployments [33], [34]. Extending the Reliability TF to incorporate such external reliability determinants could enable more resilient inference and enforcement mechanisms in shared-spectrum or cross-operator 6G environments, aligning with emerging trends in cooperative radio resource management.

Finally, the user-centric perspective adopted in SAFE-6G suggests opportunities for developing **personalised reliability profiles**, integrating behavioural patterns, device capabilities and contextual information. This direction aligns with intent-driven networking and AI-native orchestration, enabling reliability mechanisms that adapt to individual user expectations and service contexts.

Collectively, these pathways represent natural extensions of the scientific work carried out in SAFE-6G and provide fertile ground for future research on trustworthy, adaptive and user-centric 6G systems.

8 CONCLUSIONS

This deliverable has defined, specified and evaluated the Reliability TF, completing Task 4.6 and building upon the general TF framework introduced in D4.1. Following the SAFE-6G architectural principles, the Reliability TF has been implemented across its three layers, Local AI Agent, vApp and nApp, each contributing distinct capabilities to user-centric reliability assessment and enforcement.

The nApp provides multi-layer telemetry collection across application, virtualisation and infrastructure domains, enabling fine-grained visibility into service behaviour. The vApp hosts the AI/ML models responsible for reliability inference, including lightweight classifiers, DNNs and transformer-based multi-layer fusion. The Local AI Agent orchestrates flavour selection based on the LoTw, ensuring that inference complexity adapts to user requirements and operational constraints.

The scientific work carried out in this task has resulted in **six peer-reviewed publications** and the creation of **two openly accessible multi-layer datasets**, which support proactive scaling, QoE prediction, anomaly detection and transformer-assisted OOD analysis. These contributions demonstrate the maturity of the Reliability TF and its alignment with the SAFE-6G vision for cognitive, intent-driven and trustworthy 6G networks.

The Reliability TF is fully implemented and ready for integration into the SAFE-6G platform. Its operational logic, interfaces and inference mechanisms will be incorporated into the end-to-end cognitive loop evaluated in WP5, where the TF will operate alongside the CoCo and the remaining TFs.

Beyond the scope of SAFE-6G, the scientific findings of this deliverable naturally point toward several **future research pathways**, including foundation-model-based reliability inference, federated OOD detection, predictive QoE-driven resource allocation and cross-operator reliability modelling. These directions represent opportunities for advancing trustworthy, adaptive and user-centric 6G systems in future research initiatives.

9 REFERENCES

- [1] E. Atsilimis, D. Uzunidis and P. Karkazis, "Large Language Models for Out-of-Distribution Attack Detection in 6G Networks," in *2025 IEEE Conference on Network Function Virtualization and Software-Defined Networking (NFV-SDN)*, Athens, Greece, 2025.
- [2] C. Betzelos, A. Vetsos, E. Atsilimis, D. Uzunidis and P. Karkazis, "Transformer-Assisted Multi-Layer Attack Detection for Cloud-Native Video Streaming in 6G Networks," in *Proc. IEEE Int. Conf. Cloud Networking (CloudNet)*, Athens, Greece, 2026.
- [3] C. Betzelos, D. Uzunidis, P. Karkazis and H. C. Leligou, "Enhancing Reliability in 6G Networks Based on User-Driven AI," *2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 21-23 October 2024.
- [4] C. Betzelos, D. Uzunidis, A. Vetsos and P. A. Karkazis, "AI-Driven Reliability in 6G Networks: Enhancing QoE of Real-World Video Streaming," *Telecom*, vol. 7, no. 2, p. 35, 2026.
- [5] S. F. Drampalou, D. Uzunidis, A. Vetsos, N. I. Miridakis and P. Karkazis, "A User-Centric Perspective of 6G Networks: A Survey," *IEEE Access*, vol. 12, pp. 190255 - 190294, 12 December 2024.
- [6] S. Drampalou, D. Uzunidis, A. Vetsos, H. Koumaras and P. Karkazis, "An Open-Source Framework and Dataset for Multi-Layer Monitoring and Proactive Scaling of 6G Video Streaming," in *2025 IEEE Conference on Network Function Virtualization and Software-Defined Networking (NFV-SDN)*, Athens, Greece, 2025.
- [7] S. Drampalou, D. Uzunidis, A. Vetsos, H. Koumaras and P. Karkazis, "An Open-Source Framework and Dataset for Multi-Layer Monitoring and Predictive Autoscaling of 6G Video Streaming [Data set]," Zenodo, 2025. [Online]. Available: <https://doi.org/10.5281/zenodo.17523181>.
- [8] C. Betzelos, A. Vetsos, E. Atsilimis, D. Uzunidis and P. A. Karkazis, "Transformer-Assisted Multi-Layer Attack Detection for Cloud-Native Video Streaming in 6G Networks," Zenodo, 14 September 2026. [Online]. Available: <https://zenodo.org/records/22748516>.
- [9] J. C. Bárcena, P. Ducange, F. Marcelloni, G. Nardini, A. Noferi, A. Renda, G. Stea and A. Virdis, "Towards Trustworthy AI for QoE prediction in B5G/6G Networks," in *First International Workshop on Artificial Intelligence in Beyond 5G and 6G Wireless Networks (AI6G 2022)*, Padua, Italy, 2022.
- [10] S. Samarakoon, Y. Siriwardhana, P. Porambage, M. Liyanage, S.-Y. Chang, J. Kim, J. Kim and M. Ylianttila, "5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network," *arXiv*, p. arXiv:2212.01298.

- [11] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland and F. Tufvesson, "6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1166 - 1199, July 2021.
- [12] International Telecommunication Union Radiocommunication Sector, "Framework and overall objectives of the future development of IMT for 2030 and beyond," ITU Publications, 2023.
- [13] D. Uzunidis, K. Moschopoulos, C. Papapavlou, K. Paximadis, D. Marom, M. Nazarathy, R. Muñoz and I. Tomkos, "A Vision of 6th Generation of Fixed Networks (F6G): Challenges and Proposed Directions," *Telecom*, vol. 4, no. 4, pp. 758-815, 2023.
- [14] I. Tomkos, C. Christofidis, D. Uzunidis, K. Moschopoulos, C. Papapavlou, C. Tranoris, D. M. Marom, M. Nazarathy, R. Muñoz, P. Famelis and D. Larrabeiti, "The "X-Factor" of 6G Networks: Optical Transport Empowering 6G Innovations," *IT Professional*, vol. 26, no. 1, pp. 32 - 39, 2024.
- [15] C. Li, W. Hua, A. Ming and S. Shaohui, "AI-native User-Centric Network for 6G," in *2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Sanshui, Foshan, China, 2022.
- [16] S. Chen, L. Chen, B. Hu, S. Sun, Y. Wang and H. Wang, "User-Centric Access Network (UCAN) for 6G: Motivation, Concept, Challenges, and Key Technologies," *IEEE Network*, vol. 38, no. 3, pp. 154 - 162, May 2024.
- [17] X. Yan, X. An, W. Ye, M. Zhao, Y. Xi and J. Wu, "User-Centric Network Architecture Design for 6G Mobile Communication Systems," in *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Gothenburg, Sweden, 2023.
- [18] J. E. Rivadeneira, J. Sá Silva, R. Colomo-Palacios, A. Rodrigues and F. Boavida, "User-centric privacy preserving models for a new era of the Internet of Things," *Journal of Network and Computer Applications*, vol. 217, p. 103695, 2023.
- [19] V. Stavroulaki, E. C. Strinati, F. Carrez, Y. Carlinet, M. Maman, D. Draskovic, D. Ribar, A. Lallet, K. Mößner, M. Tosic, M. Uitto, S. A. Hadiwardoyo, J. Marquez-Barja, E. Garrido, M. Stamatelatos, K. Sarayeddine, P. S. Vivas, A. Mämmelä and P. Demestichas, "DEDICAT 6G - Dynamic Coverage Extension and Distributed Intelligence for Human Centric Applications with Assured Security, Privacy and Trust: from 5G to 6G," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Porto, Portugal, 2021.
- [20] C. Benzaid and T. Taleb, "AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions," *IEEE Network*, vol. 34, no. 2, pp. 186-194, March/April 2020.
- [21] L. Yang, Y. Li, S. X. Yang, Y. Lu, T. Guo and K. Yu, "Generative Adversarial Learning for Intelligent Trust Management in 6G Wireless Networks," *IEEE Netw*, vol. 36, no. 4, pp. 134-140, July/August 2022.

- [22] C. Sandeepa, B. Siniarski, N. Kourtellis, S. Wang and M. Liyanage, "A Survey on Privacy for B5G/6G: New Privacy Challenges, and Research Directions," *arXiv:2203.04264*, p. 49, 14 December 2022.
- [23] B. Mao, J. Liu, Y. Wu and N. Kato, "Security and Privacy on 6G Network Edge: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1095 - 1127, Secondquarter 2023.
- [24] P. Wei, K. Guo, Y. Li, J. Wang, W. Feng, S. Jin, N. Ge and Y.-C. Liang, "Reinforcement Learning-Empowered Mobile Edge Computing for 6G Edge Intelligence," *IEEE Access*, pp. 65156 - 65192, 10 June 2022.
- [25] S. A. Abdel Hakeem, H. H. Hussein and H. Kim, "Security Requirements and Challenges of 6G Technologies and Applications," *doi.org/10.3390/s22051969*, 2022.
- [26] SAFE-6G, "A Smart and Adaptive Framework for Enhancing Trust in 6G Networks," Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101139031, 2023. [Online]. Available: <https://safe-6g.eu/>.
- [27] H. Su, J. Zhou, X. Niu and G. Feng, "TrustFed: A Reliable Federated Learning Framework with Malicious-Attack Resistance," 6 December 2023. [Online]. Available: <https://arxiv.org/abs/2312.04597>.
- [28] Y. Zuo, J. Guo, N. Gao, Y. Zhu, S. Jin and X. Li, "A Survey of Blockchain and Artificial Intelligence for 6G Wireless Communications," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2494-2528, 2023 May Fourthquarter 2023.
- [29] W. Li, Z. Su, R. Li, K. Zhang and Y. Wang, "Blockchain-Based Data Security for Artificial Intelligence Applications in 6G Networks," *IEEE Network*, vol. 34, no. 6, pp. 31 - 37, 2020.
- [30] The 5G IA, "European Vision for the 6G Ecosystem," *doi: 10.5281/zenodo.5007671*, 2021.
- [31] C. Li, W. Hua, A. Ming and S. Shaohui, "AI-native User-Centric Network for 6G," *2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, no. 10.1109/ICCCWorkshops55477.2022.9896699, pp. 494-499, 2022.
- [32] G. Gür, A. Kalla, C. de Alwis, Q.-V. Pham, K.-H. Ngo, M. Liyanage and P. Porambage, "Integration of icn and mec in 5g and beyond networks: Mutual benefits, use cases, challenges, standardization, and future research," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1382-1412, 2022.
- [33] N. I. Miridakis, T. A. Tsiftsis, P. A. Karkazis, H. C. Leligou and P. Popovski, "Impact of Inter-Operator Interference via Reconfigurable Intelligent Surfaces," *IEEE Wireless Communications Letters*, vol. 13, no. 9, pp. 2536 - 2540, September 2024.

- [34] N. I. Miridakis and P. A. Karkazis, "Spatially Correlated Multi-RIS Communication: The Effect of Inter-Operator Interference," *IEEE Transactions on Vehicular Technology*, vol. 75, no. 3, pp. 5103 - 5108, March 2026.
- [35] P. Meher, "Network Slicing Recognition," [Online]. Available: <https://www.kaggle.com/datasets/gauravduttakiit/network-slicing-recognition>.
- [36] S. Barrachina-Muñoz, E. Zeydan and L. Blanco, "Dataset of a 5G RTSP video streaming use case (Version v1) [Dataset]," Zenodo, [Online]. Available: <https://doi.org/10.5281/zenodo.7858064>.
- [37] "Network Slicing Recognition: LazyPredict," [Online]. Available: <https://www.kaggle.com/code/gauravduttakiit/network-slicing-recognition-lazypredict>.
- [38] R. U. Mustafa, "YouTube goes 5G: Benchmarking YouTube in 4G vs 5G Through Open Datasets," [Online]. Available: <https://github.com/razaulmustafa852/youtubegoes5g>.
- [39] D. C. Phung, N.-E.-H. yellas, S. B. Ruba and S. Secci, "An Open Dataset for Beyond-5G Data-driven Network Automation Experiments," in *2022 1st International Conference on 6G Networking (6GNet)*, Paris, France, 2022.
- [40] D. Uzunidis, P. Karkazis, C. Roussou, C. Patrikakis and H. C. Leligou, "Intelligent Performance Prediction: The Use Case of a Hadoop Cluster," *Electronics*, vol. 10, no. 21, p. 2690, 2021.
- [41] P. Karkazis, D. Uzunidis, P. Trakadas and H. C. Leligou, "Design Challenges on Machine-Learning Enabled Resource Optimization," *IT Professional*, vol. 24, no. 5, pp. 69 - 74, 2022.
- [42] Y. S. K. Manjunath, J. Gao and L. Zhao, "Extended Reality Network Traffic & Quality of Experience," *IEEE Dataport*, 18 November 2025.
- [43] E. Boffetti, A. Rago, G. Piro and G. Boggia, "5G-QoERA: An Integrated Dataset for QoE Assessment in 5G NR Based on User Mobility, Radio Map, Scheduling Decisions, and Application Details," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, Milan, Italy, 2025.
- [44] G. Caso, M. Rajiullah, A. Brunstrom, L. D. Nardis, Ö. Alay and M. Neri, "A Standard-compliant Assessment of Beyond-eMBB QoS/QoE in 5G Networks," in *2024 IEEE Conference on Standards for Communications and Networking (CSCN)*, Belgrade, Serbia, 2024.
- [45] S. Peñaherrera, C. Baena, S. Fortes and R. Barco, "E2E DATASET OF 360-DEGREE VIDEO SERVICE USING MOBILE NETWORKS," *IEEE Dataport*, 22 May 2024.
- [46] ETSI, "GR NFV-REL 014 V5.1.1: Network Functions Virtualisation (NFV) Release 5; Reliability; Report on evaluating reliability for cloud-native VNFs," October 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/NFV-REL/001_099/014/05.01.01_60/gr_NFV-REL014v050101p.pdf.

- [47] ETSI, "GS NFV-IFA 055 V5.3.1: Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Specification of non-functional requirements of reliability," August 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/055/05.03.01_60/gs_NFV-IFA055v050301p.pdf.
- [48] ETSI, "TR 126 909 V19.0.0: Universal Mobile Telecommunications System (UMTS); LTE; 5G; Study on improved streaming Quality of Experience (QoE) reporting in 3GPP services and networks," October 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/126900_126999/126909/19.00.00_60/tr_126909v190000p.pdf.
- [49] ETSI, "TR 126 928 V19.0.0: 5G; Extended Reality (XR) in 5G (3GPP TR 26.928 version 19.0.0 Release 19)," October 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/126900_126999/126928/19.00.00_60/tr_126928v190000p.pdf.
- [50] ETSI, "TR 126 926 V19.0.0: LTE; 5G; Traffic Models and Quality Evaluation Methods for Media and XR Services in 5G Systems (3GPP TR 26.926 version 19.0.0 Release 19)," February 2026. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/126900_126999/126926/19.00.00_60/tr_126926v190000p.pdf.
- [51] ETSI, "TS 128 404 V19.0.0: Universal Mobile Telecommunications System (UMTS); LTE; 5G; Telecommunication management; Quality of Experience (QoE) measurement collection; Concepts, use cases and requirements (3GPP TS 28.404 version 19.0.0 Release 19)," October 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128400_128499/128404/19.00.00_60/ts_128404v190000p.pdf.
- [52] ETSI, "TR 126 929 V19.0.0: 5G; QoE parameters and metrics relevant to the Virtual Reality (VR) user experience (3GPP TR 26.929 version 19.0.0 Release 19)," October 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/126900_126999/126929/19.00.00_60/tr_126929v190000p.pdf.
- [53] ETSI, "TS 128 311 V19.0.0: LTE; 5G; Management and orchestration; Network policy management for mobile networks based on Network Function Virtualization (NFV) scenarios (3GPP TS 28.311 version 19.0.0 Release 19)," October 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128300_128399/128311/19.00.00_60/ts_128311v190000p.pdf.
- [54] SAFE-6G Consortium, "Deliverable D4.1: Cognitive coordinator, AI agents and User-centric functions," 22 December 2025. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/02/D4.1_SAFE-6G_v1.1.pdf.

- [55] SAFE-6G Consortium, "Deliverable D2.1: Definition of Technical Requirements for User-centric 6G Trustworthiness," 18 December 2025. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/02/D2.1_SAFE-6G_v1.1.pdf.
- [56] SAFE-6G Consortium, "Deliverable D2.2: Overall SAFE-6G Framework and Reference Architecture Design," 18 December 2025. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/02/D2.2_SAFE-6G_v1.1.pdf.
- [57] SAFE-6G Consortium, "Deliverable D2.3: Metaverse use-cases definition with virtual assistant for user-centric configuration," 18 December 2025. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/02/D2.3_SAFE-6G_v1.1.pdf.
- [58] SAFE-6G Consortium, "Deliverable D2.4: 6G Trustworthiness KPI & KVI definition and validation methodology," 22 December 2025. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/02/D2.4_SAFE-6G_v1.1.pdf.
- [59] SAFE-6G Consortium, "Deliverable D3.1: User-centric Distributed 6G Core over Edge-Cloud Continuum with MLOps supporting Metaverse pilots," 30 April 2025. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2025/05/D3.1_SAFE-6G_v1.0.pdf.
- [60] SAFE-6G Consortium, "Deliverable D3.5: MLOps for User-centric 6G System with differential privacy," 30 June 2026. [Online]. Available: <https://safe-6g.eu/deliverables/>.
- [61] SAFE-6G Consortium, "Deliverable D3.4: Infrastructure Openness, Monitoring and Programmability," 17 July 2026. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/09/D3.4_SAFE-6G_V1.0.pdf.
- [62] SAFE-6G Consortium, "Deliverable D3.3: User-centric Distributed 6G Core," 17 July 2026. [Online]. Available: https://safe-6g.eu/wp-content/uploads/2026/09/D3.3_SAFE-6G_V1.0.pdf.
- [63] SAFE-6G Consortium, "Deliverable D3.6: Metaverse use-cases development and user-centric configuration," 30 June 2026. [Online]. Available: <https://safe-6g.eu/deliverables/>.
- [64] SAFE-6G Consortium, "Development Codes of the Reliability Function," [Online]. Available: <https://gitlab.com/safe-6g/development/reliability-function>.
- [65] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, L. C. Cordeiro, M. Debbah and T. Lestable, "Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices," *IEEE Access*, vol. 12, pp. 23733 - 23750, 06 February 2024.
- [66] D. Tymoshchuk, O. Yasniy, M. Mytnyk, N. Zagorodna and V. Tymoshchuk, "Detection and classification of DDoS flooding attacks by machine learning method," *arXiv*, p. arXiv:2412.18990v2 [cs.CR], 25 December 2024.

- [67] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017.
- [68] Y. Shen, "Loss Functions for Binary Classification and Class Probability Estimation," *University of Pennsylvania, United States -- Pennsylvania*, p. Order No. 3179814, 2005.
- [69] D. Ruby, P. Theerthagiri, D. Jacob and Dr.Y.Vamsidhar, "Binary cross entropy with deep learning technique for Image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, July-August 2020.
- [70] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *arXiv*, p. arXiv:1908.10084, August 2019.
- [71] Hugging Face, "Sentence-Transformer model "all-MiniLM-L6-v2"," Apache 2.0, [Online]. Available: <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>. [Accessed July 2025].