

УДК 004:656.2(477)

DOI: <https://doi.org/10.18664/ikszt.v31i3.371953>

KUSHCH M.Y., Master of Science,

ABBASOV O.O. Master of Science,

KANEVSKYI M.V., Master of Science

SHCHEBLYKINA O.V. PhD, Associate Professor, Department “Automation and computer telecontrol of trains”

(Ukrainian state university of railway transport)

A comparative analysis of edge and cloud computing architectures for rolling stock monitoring systems

Abstract



This paper presents a quantitative comparison of Edge- and Cloud-oriented processing of technical-condition monitoring data for high-speed railway (HSR) rolling stock, based on a verified discrete-time digital twin of a train. Unlike approaches that treat Edge and Cloud as separate, independent configurations, the model simultaneously computes both processing pipelines within a single run, including a stochastic radio channel (correlated fading, Doppler-induced degradation, HARQ with residual block error), priority-based scheduling of two traffic classes (safety-critical and monitoring), and a thermal model of the on-board computer with throttling. A hybrid architecture mode, in which Edge pre-filters monitoring traffic before forwarding it to the Cloud, is additionally implemented and empirically evaluated. The full factorial design shows that the two architectures are bottlenecked by fundamentally different mechanisms: the Cloud degrades due to the physical unavailability of the radio channel, whereas Edge degrades due to thermal throttling under insufficient cooling. The simultaneous action of both factors drives the share of the critical *Both Down* state to a substantial fraction of the total observation time, whereas increasing the thermal headroom of the hardware platform alone is sufficient to reduce this share several-fold. Hybrid filtering delivers a measurable reduction in network traffic both under nominal conditions and under active anomalous load, but the accompanying ML-inference compute load measurably erodes the Edge node's thermal headroom. Local computation proved substantially more energy-costly than radio transmission of the same volume of data. The results quantitatively substantiate the need for a redundant hybrid monitoring architecture in which no single subsystem constitutes a single point of failure.

Keywords: Edge computing, Cloud computing, digital twin, rolling stock monitoring, IoT, predictive maintenance, Age of Information, QoS, rail transport.

Introduction

Railway operation under rigid schedules of preventive maintenance has exhausted its efficiency potential. Under high-speed conditions, this approach becomes an economic burden: sound components are often replaced prematurely out of excessive caution, while critical defects develop in the “blind spots” between scheduled inspections [1]. This problem is pushing operators toward condition-based maintenance (Predictive Maintenance – PdM), where intervention decisions rely not on a calendar but on objective monitoring data. However, the technical implementation of PdM has created an unprecedented challenge for the data-transmission infrastructure. Modern rolling stock is saturated with heterogeneous sensing: from vibration accelerometers polling axle-box units at rates of up to 50 kHz, to thermal imagers and machine-vision cameras that

generate “heavy” video traffic [2]. In a single day, a single train can generate tens of terabytes of telemetry. This intensity moves the monitoring task into the realm of Big Data, where the critical parameter is not merely the volume of stored data but the speed of its processing (Velocity) [3]. Attempting to transmit this volume of information in real time runs into the physical limitations of the train-to-ground channel. The GSM-R standard is technologically obsolete for such tasks, while LTE/4G networks, despite their nominal throughput, demonstrate instability under specific railway conditions. High linear speed (over 300 km/h) causes the Doppler effect and frequent handovers between base stations, leading to packet loss and jitter. The situation worsens in tunnels and mountainous terrain, where radio-shadow zones arise that effectively isolate on-board systems from the outside world [4], [5]. This creates a classic engineering dilemma. A centralized cloud architecture (Cloud Computing)

©KUSHCH M, ABBASOV O, KANEVSKYI M, SHCHEBLYKINA O.,2026

offers powerful resources for in-depth analysis and model training but becomes helpless when the communication channel degrades. The alternative – edge computing (Edge Computing) – allows data to be processed directly on board, minimizing latency, but is tightly constrained by the power consumption and computing capacity of local nodes [6]. The research gap lies in the absence of formalized criteria for choosing between these architectures specifically for the railway context. Most studies examine them in isolation, ignoring the dynamics of the communication channel at high speeds.

The aim of this work is to quantitatively determine the bifurcation point: the network and load parameters beyond which the advantages of a centralized cloud are negated by latency, and the resources of local nodes are exhausted – providing an objective technical basis for migrating to adaptive hybrid architectures.

Literature Review and Problem Statement

2.1. Evolution of Monitoring in Intelligent Transportation Systems (ITS)

The development of ITS in the railway sector has progressed from periodic inspections and passive telemetry (event logs, SCADA, on-board recorders) to continuous IoT-based monitoring and Predictive Maintenance (PdM). A systematic review of railway condition-monitoring methods published in *Sensors* (2021) shows that recent decades have seen a shift from manual, periodic inspections to on-board and trackside sensor systems, as well as toward the analysis of “heavy” time-series data using machine learning and deep networks for anomaly detection and failure prediction [7].

A separate review of PdM specifically for the railway industry emphasizes that current approaches rely on massive volumes of data from vibration, acoustic, thermal, and current sensors, which generate continuous data streams for the running-gear, catenary, switch, and track subsystems, among others [8]. The authors note that the shift from preventive to predictive maintenance improves rolling-stock availability but sharply increases the demands on data collection, storage, and processing, giving rise to what might be called “Big Data on Rails.”

The role of open and industry-specific datasets is likewise growing. A review of public AI-oriented railway datasets documents the emergence of large collections of images, video, vibration signals, and telemetry for track, rolling-stock, and infrastructure diagnostics [9]. A significant share of these datasets contain millions of measurements, which indirectly confirms the industry-wide growth in data volumes. This is complemented by the publication of real operational data: for example, a dataset for Chinese high-speed rail combines detailed train-movement logs with weather data and is specifically oriented toward big-data analysis for operational planning and reliability [10].

In the broader context of ITS and road transport, the review “When Intelligent Transportation Systems Sensing Meets Edge Computing” (*Appl. Sci.*, 2021) describes how the expansion of sensing modalities (video, LiDAR, radar, mobile devices) has made it unrealistic to stream complete “raw” data flows to the cloud, given bandwidth and latency constraints [11]. The authors explicitly call for a shift toward distributed processing (edge/fog), where the cloud is used mainly for offline analytics and long-term storage.

Thus, in both the railway and automotive segments of ITS, the current monitoring paradigm involves high-intensity, continuous data streams (vibration, video, thermography, high-frequency telemetry) that are used not only for ex-post diagnostics but also for prediction and failure prevention. This creates a fundamental tension between data volumes and the limitations of transport channels, particularly in high-speed scenarios.

2.2. A Critical Analysis of Cloud-Centric Approaches

A substantial number of contemporary PdM architectures for the industrial and transport sectors treat the cloud as a natural center of gravity, owing to its practically unlimited computing and storage resources. For example, the TIP4.0 platform (*Sensors*, 2021) implements a fully cloud-centric approach to PdM: sensors transmit data to a scalable cloud platform, where stream-processing pipelines and deep neural networks perform diagnostics [12]. Similarly, for railway bridges, a 2024 study in *Sensors* presents a digital twin in which low-cost wireless accelerometers transmit complete time series to a cloud infrastructure for spectral analysis and anomaly classification [13].

“Railway Cloud: Management and Orchestration Functionality Designed as Microservices” (*Appl. Sci.*, 2024) generalizes the vision of a “railway cloud” that hosts virtualized control and monitoring functions as well as rolling-stock maintenance services; the authors highlight the benefits of scalability, resource reuse, and centralized service-lifecycle management [14]. Thus, a number of works present the cloud as the natural foundation for PdM and ITS analytics owing to its elasticity, simplified algorithm updates, and capacity for global data aggregation.

However, a review of 5G for Railways (5G-R) shows that under real high-speed-railway conditions, the cloud-centric approach runs into severe physical limitations of the communication channel. Reference [15] systematizes the requirements for FRMCS/5G-R: for operations- and safety-related services, the target end-to-end latency must be <100 ms at speeds up to 500 km/h and 99.9999 % reliability. For earlier technologies, the paper cites indicative end-to-end latency figures of roughly 500 ms for GSM-R and 200-500 ms for LTE-R,

versus 1-10 ms for 5G (under idealized core-access conditions). The work also identifies a number of fundamental radio-engineering constraints. The average penetration loss into the body of a high-speed train for frequencies up to 6 GHz is 25-35 dB, which sharply degrades the SNR for passenger and on-board services. At a carrier frequency of 2.1 GHz and a speed of 350 km/h, the Doppler shift exceeds 680 Hz, complicating demodulation and channelization, particularly under rapidly varying multipath conditions. The paper further emphasizes that, unlike GSM-R, the smaller cell sizes required by the higher frequencies of 5G-R lead to a substantially higher handover rate, including group handovers, which creates risks of delay and congestion in critical scenarios.

A study of radio-communication scenarios for 5G-R classifies 18 typical scenarios in detail (viaducts, cuttings, tunnels, platforms, on-board, depots, etc.) and defines target user-experience parameters for each. For most “linear” scenarios (viaduct, tunnel, urban, rural), the expected throughput is 0.1-1 Gbit/s with 10-100 ms latency, while platforms, stations, and on-board scenarios require 1-10 ms latency under high connection density [16]. The authors separately highlight compound scenarios (e.g., groups of tunnels), where rapid transitions between different radio environments produce “severe fading” and reduced link quality.

Regarding tunnels, [He et al., 2021] describes them as waveguide environments with numerous reflections, where leaky coaxial-cable coverage is often used; in compound “tunnel-group” scenarios, repeated tunnel entries and exits substantially complicate the maintenance of stable service quality. Analogous studies of urban metro systems at the 28 GHz mm-wave band show that tunnels exhibit significant additional loss and an attenuation profile that is highly sensitive to geometry, limiting achievable range and requiring dense placement of antennas or repeaters [17].

Even in road V2X scenarios supported by MEC infrastructure, where distances to edge nodes are considerably smaller, achieving stable URLLC latency remains difficult. For instance, a microservice-based MEC architecture for warning vulnerable road users achieved a maximum end-to-end latency of about 110 ms in real-world trials, which only marginally meets the requirements of active-safety systems [18]. This indicates that even with computation placed “at the edge,” achieving <100 ms latency is not straightforward; for railways, with their long routes to centralized cloud data centers, multiple handovers, and tunnels, reliance solely on cloud processing for real-time operation appears even less viable.

Finally, work on MEC-assisted high-speed railways largely focuses on maximizing throughput or minimizing energy consumption, but operates with idealized communication models. For example, a study on

maximizing the throughput of offloaded tasks in an MEC network for high-speed rail focuses on task upload/download routing and the impact of handovers on the number of served tasks, without a detailed analysis of end-to-end latency under prolonged coverage loss [19]. A similar situation is found in FRMCS-oriented work on tuning handover parameters to meet target latency requirements: control schemes are examined in detail, but comparative analyses of different splits of computation between the on-board unit and the cloud during prolonged channel outages are absent [20].

In summary, while the cloud remains indispensable for long-term storage, global analytics, and model training, the existing literature points to a fundamental mismatch between the URLLC/PdM requirements of high-speed rail and the typical latencies and instability of LTE-R/5G-R channels at speeds >300 km/h, particularly in tunnels and compound scenarios.

2.3. The Emergence of Edge and Fog Computing in Transport

Against the backdrop of the limitations of purely cloud-based solutions in transport, the edge/fog computing paradigm is actively taking shape. A review combining ITS sensing with edge processing [11] and a taxonomy of edge-cloud computing for ITS and connected vehicles [21] describe a multi-layered architecture: vehicles and roadside units (RSUs) act as edge nodes, above them sit municipal or operator-run MEC platforms, and above those, the centralized cloud. The key principle is that time-critical tasks (V2X cooperation, local obstacle detection, basic PdM) are performed at the nearest available level, while the cloud handles “heavy” model training, long-term analytics, and the aggregation of historical data.

Concrete examples demonstrate just how radically edge processing can reduce the load on backhaul channels. A 2023 Sensors study proposes an edge system for traffic-flow classification: video is processed at a node located near the camera, and only aggregated indicators are transmitted to the cloud. During field trials (≈10,000 vehicles per day), the average bandwidth required between the edge node and the cloud was only about 1.5 MB/hour, with a classification accuracy of 93.2 % and average speed-estimation accuracy of 82.9 % [22]. Such a result would be practically impossible under a purely cloud-based model, which would need to transmit full video streams.

Similar trends can be traced in V2X research. A review of edge computing for Vehicle-to-Everything shows that edge computing is proposed as a means of simultaneously reducing latency, addressing dynamic-handover issues, and lowering network load in scenarios involving large volumes of V2X data [23]. Individual studies on “electric, connected, autonomous, and shared” vehicles describe the vehicles themselves as edge

microservers on wheels, locally processing enormous streams of sensor data and exchanging already “condensed” information with other vehicles and infrastructure [24].

In the railway context, this logic has already begun to be implemented. A 2021 Appl. Sci. paper proposes a fault-diagnosis and visualization method for high-speed trains based on edge-cloud collaboration: a deep SAES-DNN model is trained in the cloud on large volumes of historical data and then deployed to on-board or trackside edge nodes for real-time execution, with model updates carried out via transfer-learning mechanisms [25]. In this scheme, the most computationally intensive operations (training) are performed in the cloud, while latency-sensitive inference is performed at the edge.

“High-speed Train Intelligent Maintenance Pattern and Edge-Computing Perception Model” (2021) describes a three-tier “Cloud-Edge-Terminal” architecture for intelligent maintenance of electric multiple units: the edge tier handles real-time, Finite State Machine (FSM)-oriented data processing, while the cloud aggregates information from the entire train fleet and performs large-scale analytics [26].

Frontiers (2021) presents the concept of Edge Intelligence in Private Mobile Networks for Next-Generation Railway Systems, in which local private 5G/FRMCS networks with embedded edge servers are used to implement “in-network intelligence” – local analytics for PdM, environment monitoring, and video analytics – without the need to transmit all data to a centralized cloud [27].

There are also instructive examples involving comparatively “lighter” sensor streams. In “Continuous Monitoring of Train Parameters Using IoT Sensor and Edge Computing” (2021), a MEMS sensor mounted on the rail locally processes vibration signals to estimate train speed and axle load, after which the results – rather than the “raw” signals – are transmitted over NB-IoT to the cloud; the system provides round-the-clock monitoring with low energy consumption and without occupying the track [28]. Similarly, a 3D-LiDAR detection system at level crossings performs complete point-cloud processing on an embedded edge platform, sending only the final obstacle warning to the train [29].

A separate line of research concerns computation-offloading tasks in the “railway IoT” (RIoT). A 2024 Electronics paper proposes a computation-offloading strategy for detection tasks in RIoT with integrated sensing, storage, and computing functions: both direct and relay transmission schemes using NOMA are considered, and the problem of minimizing the relative delay (delay ratio) is solved via a modified DDPG algorithm [30]. The authors show that an optimal mixed offloading solution (between local nodes and MEC

servers) substantially reduces average delay compared with simple policies, but the analysis is conducted mainly in terms of time and energy, without accounting for prolonged, complete connectivity outages.

A 2022 IEEE paper on multi-access MEC in RIoT proposes a hybrid DRL approach (DDQN + DDPG) for the joint optimization of sub-carriers, offloading ratios, transmit power, and computing resources, with the goal of minimizing a weighted sum of delay and energy consumption [31]. The authors demonstrate near-optimal solutions through policy learning; however, as in the previous case, a relatively stable channel is assumed, and “connectivity failure” is not treated as a central scenario.

Finally, a 2023 PeerJ paper on optimizing a railway logistics-center layout considers a “mobile cloud-edge computing” architecture in which tasks can be executed on on-board devices, on edge servers, or in the cloud; simulation shows that this two-tier scheme reduces latency by roughly 40 % compared with alternative task-allocation strategies [32]. However, that scenario is oriented more toward logistics than toward the safety-critical monitoring of rolling stock under harsh radio-communication conditions.

In sum, the literature demonstrates a clear trend toward the use of edge/MEC in transport (including railways) to reduce latency and network load, but these solutions are mostly developed either for isolated tasks (local detection, bridge SHM) or in the context of idealized communication models, without a full comparison of edge and cloud architectures under identical mobility and channel-constraint conditions.

2.4. Research Gap

The review of the literature above makes it possible to identify three relatively isolated strands of research: 1) PdM and monitoring in the railway industry, 2) the radio-engineering aspects of FRMCS/5G-R, and 3) MEC/edge computing in ITS and RIoT.

1. PdM and railway monitoring. Works [7] and [8] focus on data-analysis algorithms (ML/DL, anomaly detection, remaining-useful-life prediction), paying less attention to the placement of computing resources. Architectural assumptions are often implicitly cloud-centric – it is assumed that complete arrays of “heavy” data (including video and high-frequency vibration) can be delivered to a data center. Railway digital-twin applications confirm this approach: in [13], complete time series, after minimal pre-processing, are sent to the cloud for analysis, bypassing the channel constraints characteristic of high-speed operation.

2. LTE-R/5G-R radio channels. Work on 5G-R characterizes in detail the requirements for latency, reliability, and throughput, as well as the phenomena associated with high speed: Doppler shift, coverage gaps in tunnels, group handovers, and so on [15], [16].

However, these works essentially do not analyze how specific monitoring architectures (edge vs. cloud) behave under these conditions. For example, it is established that GSM-R yields ~500 ms of end-to-end latency, LTE-R yields 200-500 ms, and 5G-R is expected to deliver <100 ms for critical services, but no analysis is offered of what fraction of the time these requirements are actually met during operation at >300 km/h with repeated tunnel entries, nor of the implications for PdM systems that rely entirely on the cloud.

3. **MEC/edge in RIoT and ITS.** MEC solutions for railways mostly focus on formulating optimization/DRL problems for offloading and resource allocation, with the goals of minimizing latency or energy or maximizing throughput [19, 30, 31]. In doing so,

- **the channel is typically modeled as an ideally controlled resource** with known fading/noise statistics, but without the prolonged complete outages (blackouts) characteristic of tunnels, compound scenarios, and poorly covered segments;

- the analysis is conducted **at the level of average delay/energy**, without assessing the “survival” of safety functions in worst-case scenarios (for example, whether a crack/overheat detector remains able to function during a 5-30-second loss of network connectivity);

- comparisons are usually made between different offloading policies within a single MEC architecture, rather than between “pure edge” (all critical computation performed on board) and “pure cloud” (all “heavy” data offloaded to the center).

As for **heavy data types** (high-frequency vibration signals, high-resolution video, thermal images) for rolling stock at speeds >300 km/h, existing studies typically either:

- confine themselves to **local tasks** (monitoring of a single track segment [28] or a level crossing [29]), where the channel to the cloud is not a “bottleneck”; or

- consider **idealized 5G/5G-R conditions**, where throughput and latency correspond to nominal values (e.g., 0.1-1 Gbit/s and 10-100 ms for linear scenarios [16]), without capturing tunnel-waveguide effects, 25-35 dB penetration losses, frequent handovers, or the instability of group handovers [15].

This makes it possible to formulate the **key gap**:

- existing studies either focus on **processing/compression algorithms** (PdM, ML/DL), or on **radio-level/network technologies** (URLLC, handover, FRMCS), but **almost never combine** these two aspects within a single experimental or modeling framework capable of assessing the safety system’s ability to “survive” channel degradation or failure;

- there are practically no studies that **introduce reliability metrics for the monitoring system under connectivity-failure conditions** (for example, the probability of missing a critical defect under scenario X for architecture A vs. B), and that examine how the choice between edge and cloud affects these metrics for given channel characteristics (in particular, handover delays, Doppler-shift magnitudes, and tunnel blackout durations);

Problem statement. The scientific problem addressed by this paper is formulated as the **absence of a scientifically grounded, quantitative comparison of edge and cloud computing architectures for high-speed-train rolling-stock monitoring systems** that would:

- account for **the realistic constraints of railway communication channels** (GSM-R/LTE-R/5G-R, including tunnels, compound scenarios, group handovers, and high Doppler shift);

- operate with **heavy data types** (vibration, video, thermography) characteristic of modern PdM and safety systems;

- evaluate not only average latency and throughput but also **viability metrics for safety functions** under connectivity degradation/loss (for example, the maximum allowable connectivity-outage duration for which an edge architecture remains able to sustain the required level of safety, whereas a cloud architecture does not).

The comparison of “pure edge” and “pure cloud” proposed in this paper under such conditions will help close this gap, providing quantitatively substantiated arguments for choosing a rolling-stock monitoring architecture in the era of FRMCS/5G-R and railway big data.

Aim and Objectives of the Study

Aim: to quantitatively compare Edge- and Cloud-oriented processing of HSR rolling-stock monitoring data in terms of latency, reliability (Reliability, expressed as Packet Loss Ratio and the share of time spent in each of four operating states), data timeliness (Age of Information), and network-traffic volume, and to empirically assess the feasibility of a hybrid architecture in which the Edge node pre-filters data before forwarding it to the Cloud.

Objectives:

1. Develop a formalized data-flow model split into two QoS priority classes – safety-critical (vibration diagnostics, axle-box temperature) and monitoring (video) – reflecting the real heterogeneity of rolling-stock telemetry.

2. Build a mathematical model of end-to-end latency, packet-loss ratio, and age of information separately for the Edge and Cloud processing pipelines, including a stochastic radio-channel model (correlated fading, Doppler-induced degradation, HARQ) and a thermal model of the on-board computer with throttling.

3. Implement two architecture modes – direct duplication of the raw stream to both subsystems (baseline) and hybrid filtering, in which Edge forwards only a compressed version of the monitoring traffic to the Cloud – and quantitatively evaluate the resulting network-traffic savings.

4. Perform simulation modeling of system operation under unstable connectivity (tunnels, NLOS zones, handovers) and a varying thermal headroom of the on-board computer (three hardware profiles) within a full factorial experimental design.

5. Determine the boundary conditions (state bifurcations) at which each architecture loses stability, and quantitatively compare the physical energy consumption of local computation against the energy cost of radio transmission.

Both processing pipelines (Edge and Cloud) run **in parallel within a single digital twin** for every simulation run, and the hybrid architecture is **implemented and quantitatively evaluated** as part of this study. The simulation is implemented as a custom discrete-time simulator in Python with a fixed sampling step $\Delta t = [\Delta t = 0,01]$ s, ensuring synchronous updates of the two parallel processing pipelines at every tick.

Methodology and System Modeling

4.1. Formalization of the Cyber-Physical Monitoring Architecture

We consider a digital twin of a rolling-stock technical-condition monitoring system operating under high-speed conditions (High-Speed Railway, HSR). The architecture is formalized as a directed graph $G = (V, E)$, where $V = V_{sens} \cup V_{edge} \cup V_{cloud}$ is the set of nodes (vibration and temperature sensors, video cameras, the On-Board Unit computer, the cloud server), and E is the set of communication links (the internal OBU bus and the wireless uplink to the cloud).

Both processing pipelines are built **in parallel within a single digital twin** (the TrainDigitalTwin class): the volume of data generated at step t is simultaneously fed into both the local Edge processing queue and, after being transformed by the architecture mode (Section 4.7), the Cloud transmission queue. In addition, a **hybrid architecture mode** (ArchitectureMode.HYBRID_FILTERED) is provided, in which Edge pre-filters data before forwarding it to the Cloud, making it possible to evaluate not only fault tolerance but also the **actual traffic savings** achieved through local processing (Section 4.7).

Each unit of traffic belongs to one of two QoS priority classes (PriorityClass)

$$c(t) \in \{ \text{SAFETY}, \text{MONITORING} \}, \quad (1)$$

where SAFETY combines vibration diagnostics and axle-box temperature, while MONITORING denotes the video stream.

For each class, the latency, packet-loss ratio (PLR), and age of information (AoI) are computed separately in each subsystem; based on the aggregated (worst-per-class) values of these metrics, the system is classified at every step into one of four states (Section 4.8).

4.2. Mathematical Model of Traffic Generation

The input information flow of the SAFETY class is formed by vibration diagnostics (axle-box accelerometers) and axle-box temperature:

$$\lambda_{safety}(t) = \underbrace{N_{vib} \cdot f_s \cdot B_{vib} \cdot \xi_{vib}(t)}_{\text{vibration diagnostics}} + \underbrace{N_{temp} \cdot f_{temp} \cdot B_{temp} \cdot \xi_{temp}(t)}_{\text{axle-box temperature}}, \quad (2)$$

where $N_{vib} = 8$;

$$f_s = 50 \times 10^3 \text{ Hz};$$

$B_{vib} = 24$ bits (the product of $\approx 9,6$ Mbit/s – the nominal vibration-diagnostics bitrate);

$$N_{temp} = 8, f_{temp} = 10 \text{ Hz};$$

$B_{temp} = 16$ bits (a low-frequency channel, since the thermal process in the axle box is much slower than vibration);

$\xi_{vib}(t), \xi_{temp}(t) \sim U(0,95; 1,05)$ is multiplicative noise of the measurement paths.

Traffic of the MONITORING class (video) is modeled as a two-state Markov-modulated Poisson process (MMPP):

$$\lambda_{monitoring}(t) = N_{cam} \cdot R_{video}(t) \cdot k_{anomaly}(t), \quad R_{video}(t) = \begin{cases} R_{min} \cdot \xi_v(t), & \text{"low activity"} \\ R_{max} \cdot \xi_v(t), & \text{"high activity"} \end{cases}, \quad (3)$$

where $N_{cam} = 2$;

$$R_{min} = 4 \text{ Mbit/s};$$

$$R_{max} = 12 \text{ Mbit/s};$$

$\xi_v(t) \sim U(0,9; 1,1)$, state switching is a Bernoulli trial with probability $p_{MMPP} = 0,05$ at each step;

$k_{anomaly}(t)$ is the multiplier (4) active during an Anomaly Burst.

4.3. Model of Event-Driven Load (Anomaly Burst)

The binary anomalous-load process $A(t) \in \{0,1\}$ is activated by a Bernoulli trial with probability $p_a = 0,0008$ at each step of the inactive state and is held fixed for $T_a = 3,0$ s:

$$k_{anomaly}(t) = \begin{cases} k_v = 5,0, & A(t) = 1 \\ 1, & A(t) = 0 \end{cases} \quad U(t)|_{A(t)=1} \leftarrow \min(1, U(t) + \Delta U), \quad \Delta U = 0,4$$

where $U(t)$ is the CPU utilization of the Edge node (7).

The theoretical steady-state fraction of time spent in the active state, $P(\text{active}) \approx p_a T_a / (p_a T_a + \Delta t) \approx 19,4\%$ is used as an internal sanity check on the correctness of the random-event generator implementation.

4.4. Stochastic Model of the HSR Radio Channel

The communication channel accounts for three elements of realism essential to an HSR route: time-correlated fading, a triangular signal-level droop profile near the cell edge, and a probabilistic model of residual block error accounting for HARQ.

Doppler shift. At train speed v (km/h) and carrier frequency $f_c = 2,1$ GHz:

$$f_d = \frac{v/3,6}{c} f_c, \quad c = 3 \times 10^8 \text{ m/s.} \quad (5)$$

Correlated shadow fading.

The effective signal level is modeled by a Gauss-Markov AR(1) process in the dB domain, with autocorrelation dependent on the channel coherence time (rather than being redrawn independently at every tick, which would ignore the physical inertia of fading):

$$\tau_c \approx \frac{1}{k_{coh} \cdot f_d}, \quad \rho = e^{-\frac{\Delta t}{\tau_c}},$$

$$SNR_{dB}(t) = \mu_S + \rho(SNR_{dB}(t - \Delta t) - \mu_S) + \sigma_S \sqrt{1 - \rho^2} \mathcal{N}(0,1) \quad (6)$$

where $k_{coh} = 4$ (FADING_COHERENCE_FACTOR);

$$\mu_S = 29,5 \text{ dB}, \quad \sigma_S = 1,5 \text{ dB for LOS};$$

$$\mu_S = 19,0 \text{ dB}, \quad \sigma_S = 3,0 \text{ dB for NLOS}.$$

Triangular droop near the cell edge. A deterministic trend of position within the inter-handover interval is additionally taken into account:

$$\text{droop}_{dB} = -(1 - (1 - |2p - 1|)) \cdot \Delta_{edge}, \quad p = \frac{x - x_{HO}}{L_{HO}} \in [0,1] \quad (7)$$

where $\Delta_{edge} = 2$ dB (PATHLOSS_CELL_EDGE_DROOP_DB) is the maximum additional attenuation at the cell edge relative to its center.

Throughput with MCS derating. The effective SNR after Doppler-induced degradation is converted into throughput via the Shannon-Hartley theorem, discounted by the spectral-efficiency factor of the actual modulation-and-coding scheme:

$$SNR_{eff}(t) = 10^{\frac{(SNR_{dB}(t) + \text{droop}_{dB})}{10}} \cdot e^{-\frac{f_d}{f_{decay}}},$$

$$C_{real}(t) = \eta_{MCS} \cdot B_w \cdot \log_2(1 + SNR_{eff}(t)), \quad (8)$$

where $f_{decay} = 500$ Hz, $B_w = 20$ MHz, $\eta_{MCS} = 0,75$ (SPECTRAL_EFFICIENCY_FACTOR).

HARQ and residual BLER. Rather than a hard “channel available / channel unavailable” switch at the throughput level, a probabilistic block-error model is introduced as a sigmoid function of the effective SNR (dB), with HARQ combining across up to $N_{HARQ} = 2$ retransmissions:

$$BLER_1(t) = \frac{1}{1 + \exp\left(\frac{(SNR_{eff, dB}(t) - \theta_B)/s_B}{BLER_1(t)^{N_{HARQ}+1}}\right)}, \quad BLER_{HARQ}(t) = BLER_1(t)^{N_{HARQ}+1}, \quad (9)$$

$$C(t) = C_{real}(t) \cdot (1 - BLER_{HARQ}(t)), \quad \delta_{HARQ}(t) \approx T_{RTT} \cdot BLER_1(t) \quad (10)$$

where $\theta_B = 5$ dB;

$$s_B = 3 \text{ dB};$$

$T_{RTT} = 8$ ms (HARQ_ROUND_TRIP_S) is the expected additional delay due to retransmissions, added directly to the Cloud network-segment delay (14).

Tunnels and handover. If a segment is of type TUNNEL, $C(t) = 0$ for the entire duration of its traversal. A handover has **its own independent duration** $T_{HO} = 0,05$ s (HANDOVER_DURATION_S), independent of the sampling step Δt : as soon as the distance traveled since the previous handover reaches L_{HO} of the segment, the channel is forcibly switched into outage for the entire T_{HO} interval.

4.5. QoS Priority Queue, Thermal Model, and Edge-Node Energy Consumption

The OBU compute node is formalized as a queuing system with a shared buffer of limited capacity $K = 300$ Mbit, serving two priority classes (1) under a **strict-priority** discipline: SAFETY is served first, using the full available capacity $\mu(T_{cpu})\Delta t$, with the remainder allocated to MONITORING; when the buffer overflows, the excess is dropped first from MONITORING (tail-drop), and only as a last resort from SAFETY.

The thermal dynamics and throttling are described by the equations:

$$C_{th} \frac{dT_{cpu}(t)}{dt} = P_{active} \cdot U(t) - \frac{T_{cpu}(t) - T_{amb}}{R_{th}},$$

$$\mu(T_{cpu}) = \begin{cases} \mu_{max}, & T_{cpu} \leq T_{crit} \\ \mu_{max} e^{-\alpha(T_{cpu} - T_{crit})} \zeta(t), & T_{cpu} > T_{crit} \end{cases} \quad (11)$$

where $C_{th} = 40$;

$$P_{active} = 35 \text{ W};$$

$$T_{crit} = 75^\circ \text{C}, \quad \zeta(t) \sim U(0,9; 1,0);$$

the parameters R_{th} , T_{amb} , μ_{max} , α are determined by the edge profile (Section 4.9). The utilization $U(t)$ is additionally increased by (4) during an Anomaly Burst and by $\Delta U_{ML} = 0,10$ (ML_INFERENCE_EXTRA_UTILIZATION) when the hybrid-filtering mode is active (Section 4.7) – local ML inference carries a computational cost that cannot legitimately be ignored.

Age of information, per class. Since SAFETY is served practically instantaneously in the strict-priority queue, computing a single AoI based on the time of the most recently processed data packet (regardless of class) would mask the real staleness of the MONITORING queue behind the near-instant servicing of SAFETY. AoI is therefore computed separately for each class by definition $A(t) = t - U_{class}(t)$, where $U_{class}(t)$ is the generation instant of the most recent fully processed data unit of that class, and the aggregated subsystem indicator is the worse (maximum) of the two:

$$AoI_{edge}(t) = \max(AoI_{edge}^{safety}(t), AoI_{edge}^{monitoring}(t)) \quad (12)$$

Energy. Separately from the thermal equation (11) (so as not to alter the already-verified dynamics), the physical energy consumed by computation is tracked:

$$P_{edge}(t) = P_{idle} + P_{active} \cdot U(t),$$

$$E_{edge} = \int P_{edge}(t) dt + \sum_t \gamma_{ML} \cdot \frac{\text{volume}(t)}{10^6} [\text{ML active}], \quad (13)$$

where $P_{idle} = 5$ W (EDGE_IDLE_POWER_W);

$\gamma_{ML} = 0,02$ J/Mbit (ML_INFERENCE_ENERGY_J_PER_MBIT).

4.6. Network and Server Segments of the Cloud Subsystem

The Cloud subsystem consists of **two sequential segments**: a network segment (the radio channel with a deadline) and a server segment (backend processing).

Network segment – a queue with a deadline (TTL reneging) and QoS priority (1):

$$tx(t) = \min(Q_{net}(t), C(t)\Delta t), \quad L_{net}(t) = \frac{Q_{net}(t)}{C(t)} + \delta_{HARQ}(t). \quad (14)$$

A packet whose waiting time exceeds $TTL = 1,5$ s is dropped regardless of $C(t)$.

Server segment. Data that have successfully passed through the radio channel enter a queue of limited backend throughput $\mu_{srv} = 500$ Mbit/s (multi-tenant load from parallel trains), with an additional fixed processing delay of $\delta_{srv} = 20$ ms (writing to the message queue/database):

$$L_{server}(t) = \frac{Q_{server}(t)}{\mu_{srv}} + \delta_{srv}, \quad L_{cloud}(t) = L_{net}(t) + L_{server}(t). \quad (15)$$

$$= \begin{cases} (\lambda_{safety}\Delta t, \lambda_{monitoring}\Delta t), & \text{PARALLEL_RAW} \\ (\lambda_{safety}\Delta t, \rho_c \cdot \lambda_{monitoring}\Delta t), & \text{HYBRID_FILTERED, } A(t) = 0 \\ (\lambda_{safety}\Delta t, \lambda_{monitoring}\Delta t), & \text{HYBRID_FILTERED, } A(t) = 1 \end{cases} \quad (16)$$

where $\rho_c = 0,05$ (EDGE_TO_CLOUD_COMPRESSION_RATIO) is the share of video traffic forwarded as compressed features under normal conditions;

SAFETY is always forwarded in full (small volume, safety-critical), while during an Anomaly Burst MONITORING temporarily reverts to the full raw stream for dispatcher verification (“summary normally, raw evidence on alarm”). The PARALLEL_RAW mode serves as the baseline for comparison, in which Edge and Cloud process an identical full stream independently of one another.

Traffic savings are computed relative to the volume **queued for the channel** ($b_{attempted}$), rather than the volume actually transmitted after the channel, since the latter depends on link quality (losses in a tunnel) rather than on the filtering decision:

$$\text{savings} = 1 - \frac{\sum_t (b_{safety}^{cloud}(t) + b_{monitoring}^{cloud}(t))|_{HYBRID}}{\sum_t (\lambda_{safety}(t) + \lambda_{monitoring}(t))\Delta t|_{PARALLEL_RAW}}. \quad (17)$$

Table 1

Thermal and computational headroom parameters of the three hardware-platform edge profiles

Profile	$T_{amb}, ^\circ\text{C}$	R_{th}	$\mu_{max}, \text{Mbit/s}$	α
---------	---------------------------	----------	----------------------------	----------

The server segment is considered reliable (loss-free) – all losses occur in the network segment, where the TTL deadline applies. $AoI_{cloud}(t)$ is computed analogously to (12), separately per class, over packets that have passed through BOTH segments.

4.7. Hybrid Architecture and Traffic Savings

An ArchitectureMode parameter is introduced, defining the volume of data that Edge actually forwards to the Cloud:

$$(b_{safety}^{cloud}(t), b_{monitoring}^{cloud}(t)) =$$

4.8. System State Classification

A subsystem is considered healthy (OK) if the safety thresholds for latency and PLR are simultaneously satisfied:

$$OK(L, PLR) \Leftrightarrow (L < L_{safety}) \wedge (PLR < PLR_{safety}),$$

$$L_{safety} = 0,15 \text{ s}, \quad PLR_{safety} = 2 \%. \quad (18)$$

Applying (18) to the pairs (L_{edge}, PLR_{edge}) and (L_{cloud}, PLR_{cloud}) yields four mutually exclusive states: **Both OK, Cloud Down, Edge Down, Both Down** (Edge OK/not OK $\times$ Cloud OK/not OK).

4.9. Routes, Edge Profiles, and Factorial Design

The study uses four routes (Open terrain, 45 km, LOS, 350 km/h; Deep tunnels, 28 km, with two TUNNEL segments of 8 and 6 km; Urban development, 16 km, NLOS; Mixed stress test, 20 km, LOS+NLOS+TUNNEL) and three thermal/computational-headroom edge profiles (Table 1):

ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

Nominal	25	0,60	120	0,15
Degraded	45	1,40	90	0,30
Overload	50	1,80	60	0,45

Table 1 shows that the profiles differ not only in ambient temperature: the Overload profile combines the highest ambient temperature (50 °C), the largest thermal resistance (1.80), and the largest thermal performance-degrading coefficient (0.45) with the lowest maximum throughput (60 Mbit/s) – i.e., it has the smallest thermal headroom of all three profiles. The Nominal profile, by contrast, combines the lowest ambient temperature (25 °C) with the highest throughput (120 Mbit/s). It is precisely this difference that determines why, under identical load, the Overload profile systematically leads to thermal throttling of the Edge node, whereas the Degraded profile does not (details in Sections 5.3-5.4).

The factorial design combines three baseline scenarios (Nominal, without Anomaly Burst) and three aggravated ones (Overload/Degraded, with Anomaly Burst) – six scenarios, each executed under **two architecture modes** (16) with $N_{run} = 5$ replications, i.e., $6 \times 2 \times 5 = 60$ independent runs.

4.10. Software Implementation

The model is implemented as the `train_digital_twin` Python package (modular structure: `traffic.py`, `channel.py`, `edge_node.py`, `cloud_node.py`, `queueing.py`, `simulator.py`), verified by 29 automated tests (pytest) covering the correctness of the priority packet queue, the channel model (tunnel/handover/BLER), traffic savings, and integration scenarios. The simulation is discrete-time (Euler method, $\Delta t = 0,01$ s), with no

external DES frameworks. The results of the full factorial design (60 runs) are exported as an Excel report with native charts and formulas that reference the safety thresholds (18) – this report is the source of the figures in Section 5.

Code availability. The complete simulator source code, test suite, and factorial-design reproduction script are published in an open repository:

<https://github.com/nikolaykusch/tdt>

Results

The results were obtained by executing the full factorial design (Section 4.9) under two architecture modes (16): 6 scenarios $\times$ 2 modes $\times$ $N_{run} = 5$ replications = 60 independent runs, 2 374 920 simulation ticks in total. All figures in this section are native charts built directly from the resulting factorial-design data; dashed lines denote the safety thresholds (18). Tables 2-5 present the complete set of numerical results for all six scenarios, so that this section is self-contained and does not require reference to the simulator's source files.

5.1. Distribution of System States and Overall PLR

Fig. 1 summarizes the distribution of time the system spends in each of the four operating states (18) for the six factorial-design scenarios under the realistic HYBRID_FILTERED mode.

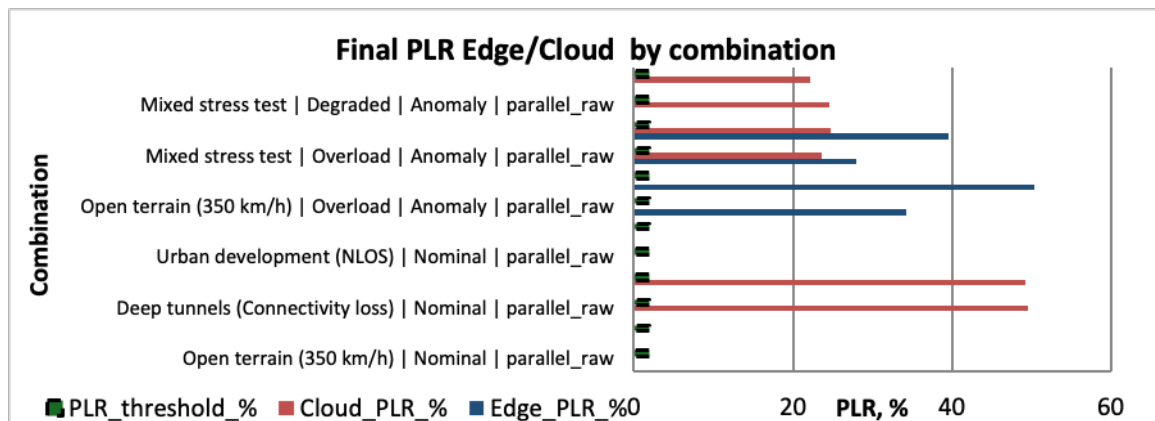


Fig. 1. Share of time the system spends in the Both OK / Cloud Down / Edge Down / Both Down states across the six factorial-design scenarios (HYBRID_FILTERED mode)

The dominant system state changes depending on which bottleneck is active (exact values in Table 4): *Both OK* - 99,8 % of the time in open terrain under

ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

nominal conditions, $99,7\%$ in urban development; *Cloud Down* - $82,2\%$ in the tunnel scenario; *Edge Down* - $87,9\%$ in open terrain under the Overload profile with Anomaly Burst enabled; *Both Down* - $74,7\%$ in the mixed stress test under the Overload profile. Fig. 2

supplements this with the final cumulative PLR of both subsystems across all 12 combinations (both architecture modes), with the safety-threshold line $PLR_{safety} = 2\%$ (18).

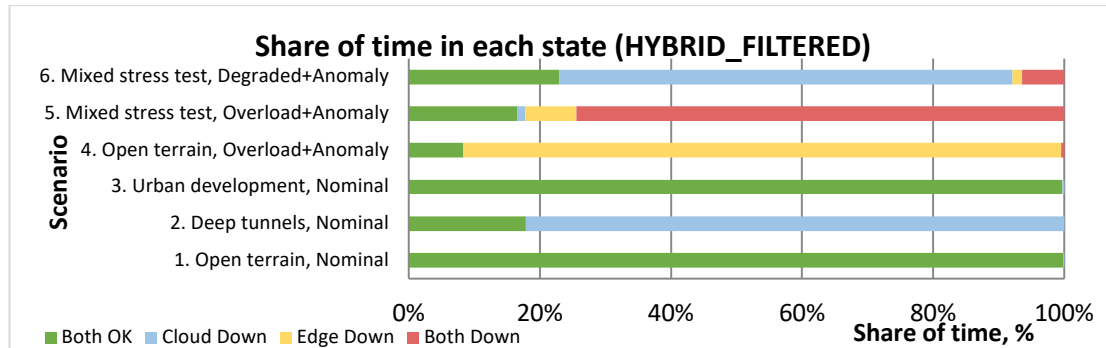


Fig. 2. Final cumulative Packet Loss Ratio of the Edge and Cloud subsystems across the 12 factorial-design combinations (6 scenarios $\times$ 2 architecture modes), with a dashed line marking the 2 % safety threshold

Neither subsystem exceeds the PLR_{safety} threshold in the nominal scenarios (1-3); Cloud exceeds the threshold in scenarios with channel degradation (tunnels, mixed stress test), while Edge exceeds it in scenarios with thermal degradation (Overload with Anomaly Burst); moreover, in HYBRID_FILTERED mode the peak Edge_PLR is systematically higher than in PARALLEL_RAW (details in Section 5.3).

Figs. 3-6 combine all six factorial-design scenarios on shared axes for each of the four key indicators: a solid line denotes Edge, a dashed line denotes Cloud (except for the thermal dynamics, where there is no Cloud component), color corresponds to the scenario, and the black dashed line marks the safety threshold (18). Each scenario's line stops where its route ends.

5.2. Combined Dynamics of Key Indicators Along the Route

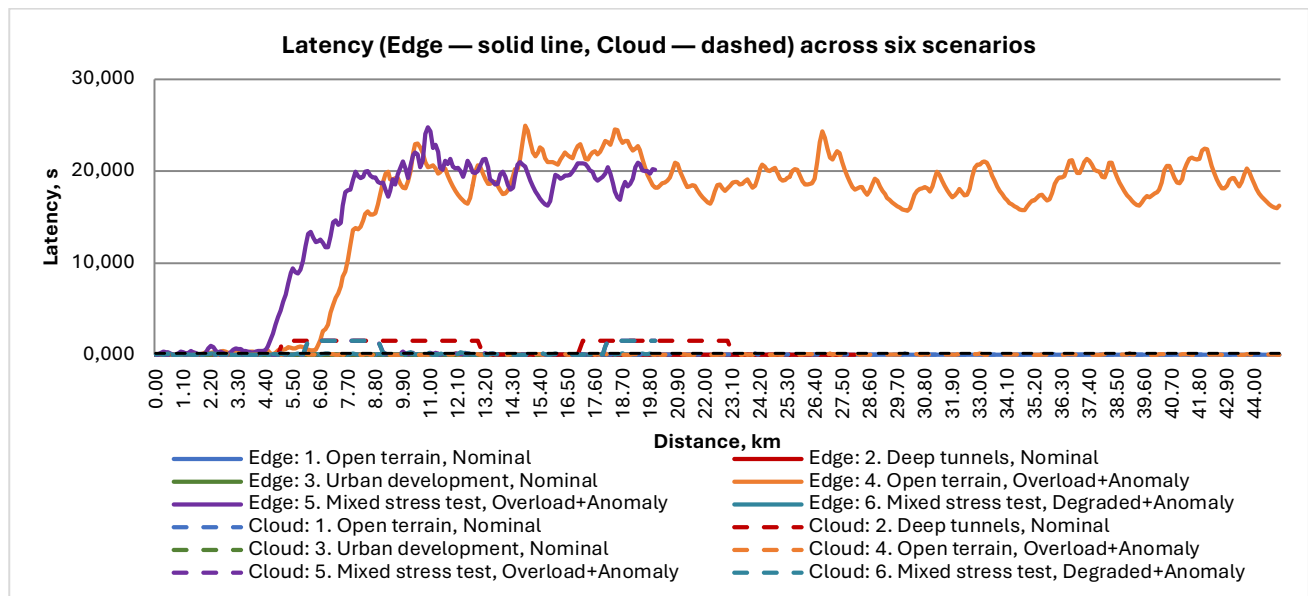


Fig. 3. Edge (solid line) and Cloud (dashed) latency along the distance for all six factorial-design scenarios

In Fig. 3, scenarios 1-3 (Nominal, without Anomaly Burst) show practically zero Edge latency along the entire route; the Cloud latency of scenario 2 (“Deep tunnels”) jumps to the $TTL = 1,5$ s deadline every time a

tunnel segment (8 and 6 km) is traversed, whereas scenarios 1 and 3 keep Cloud latency close to zero. Scenarios 4-5 (Overload, Anomaly Burst) show the opposite pattern: Edge latency rises to 15-20 s within the first few kilometers of the route and remains at that level,

ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

while the Cloud latency of the same scenarios stays close to zero (scenario 4, LOS channel) or is moderately elevated (scenario 5, mixed channel). Scenario 6 (Degraded, Anomaly Burst) is the only stress scenario in which Edge latency remains low (the thermal headroom

of the Degraded profile is sufficient), and only the Cloud component degrades, in sync with scenario 5, since both traverse the same route.

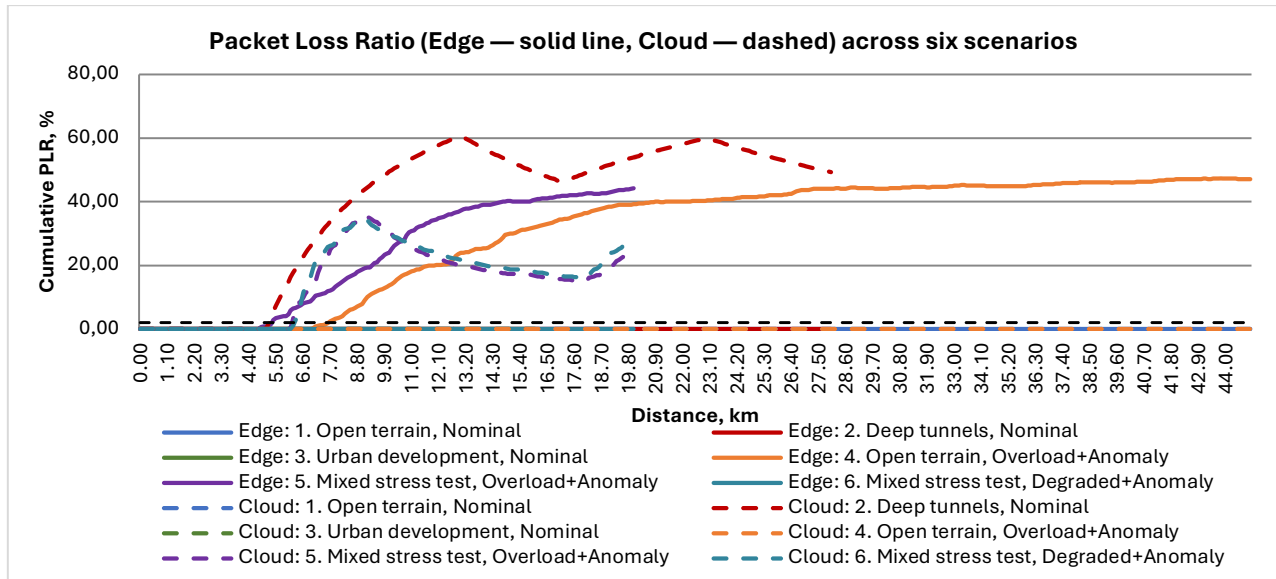


Fig. 4. Edge (solid line) and Cloud (dashed) Packet Loss Ratio along the distance for all six scenarios

Fig. 4 shows the cumulative nature of PLR: the curves only increase along the distance. Edge_PLR is noticeably non-zero only for scenarios 4 and 5 (Overload), reaching 40-50 %; for scenario 6 (Degraded), Edge_PLR remains at zero despite the same load level – a

direct consequence of the difference in thermal profiles (Section 5.3). Cloud_PLR increases stepwise on the tunnel and NLOS segments of scenarios 2, 5, and 6, remaining zero for scenarios 1, 3, and 4 (purely LOS routes or short nominal stretches).

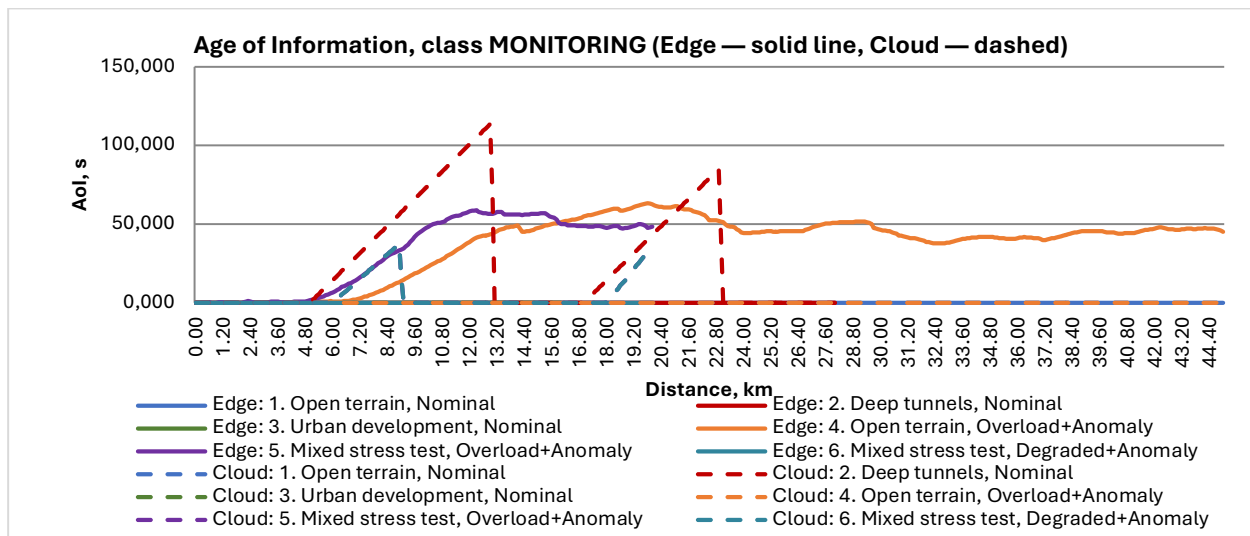


Fig. 5. Age of Information of the MONITORING class for Edge (solid line) and Cloud (dashed) along the distance for all six scenarios

Fig. 5 illustrates video-stream AoI. In scenario 2, $AoI_{cloud}^{monitoring}$ increases linearly during every tunnel traversal (the slope of the line corresponds to the rate at which data become stale when no packet is delivered) and resets to zero immediately after exiting the tunnel; the maximum practically coincides with the traversal time of

the longer (8-km) tunnel at 250 km/h. In scenarios 4-5, by contrast, $AoI_{edge}^{monitoring}$ increases – a mirror effect of the same nature, but caused by thermal throttling of the Edge queue rather than the absence of a channel.

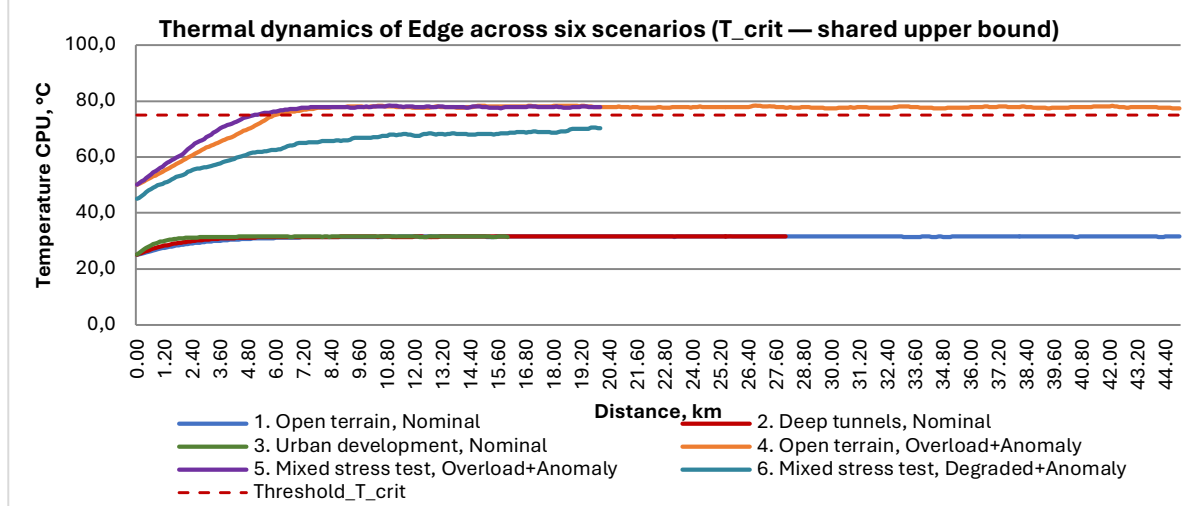


Fig. 6. Edge thermal dynamics for all six scenarios, with T_{crit} as a common upper bound

Fig. 6 directly compares the thermal profile across all six scenarios: scenarios 1-3 (Nominal) plateau at $\approx 32^\circ\text{C}$, substantially below T_{crit} ; scenarios 4-5 (Overload) cross $T_{crit} = 75^\circ\text{C}$ within the first few kilometers and stabilize at a plateau of $\approx 79^\circ\text{C}$; scenario 6 (Degraded, the same route and load as scenario 5) plateaus at $\approx 74^\circ\text{C}$ – below T_{crit} , so throttling (11) is practically never triggered. This difference precisely explains the divergence between the Latency and PLR curves of scenarios 5 and 6 in Figs. 3-4 under an identical input load.

5.3. Effect of Architecture Mode on Thermally Induced Degradation of the Edge Subsystem

A comparison of the PARALLEL_RAW and HYBRID_FILTERED modes on scenario 4 (open terrain, Overload profile, Anomaly Burst) quantitatively illustrates the cost of local ML filtering (Table 2). In PARALLEL_RAW mode, the average Edge latency is $11,66\text{ s}$ (maximum $27,12\text{ s}$), $Edge_PLR = 32,8\%$, peak temperature $78,53^\circ\text{C}$; in HYBRID_FILTERED mode, the average latency rises to $16,08\text{ s}$ (maximum $41,57\text{ s}$), $Edge_PLR = 47,1\%$, peak temperature $79,48^\circ\text{C}$. The difference is explained by the additional CPU utilization $\Delta U_{ML} = 0,10$ (11) from the local ML inference used for filtering, which is active only in HYBRID_FILTERED mode: the hybrid architecture, advantageous in terms of traffic (Section 5.5), carries a measurable computational cost that systematically erodes the Edge node's thermal headroom precisely when it is already close to the limit.

5.4. Non-Additivity of Failures in the Combined Stress Test

On the “Mixed stress test” route (LOS, NLOS, TUNNEL segments) under the **Overload** profile, the simultaneous action of thermal throttling and channel degradation produces a share of the critical state of *Both Down* $74,7\%$ (Fig. 1, Table 4), $Edge_PLR = 44,2\%$, $Cloud_PLR = 24,7\%$ (HYBRID_FILTERED). If the subsystem failures were statistically independent, with the probabilities observed in the isolated scenarios 2 and 4 ($P(\text{Cloud not OK}) \approx 0,822$, $P(\text{Edge not OK}) \approx 0,885$), the expected share of simultaneous failure would be $\approx 0,728$ – a value close to the actually observed $0,747$. By contrast, under the **Degraded** profile, on the same route and with the same Anomaly Burst, the maximum CPU temperature is $73,91^\circ\text{C}$ (HYBRID_FILTERED) – below T_{crit} , i.e., the thermal-throttling mechanism (11) is practically never triggered: $Edge_PLR \approx 0\%$, and the share of *Both Down* drops to $6,2\%$ – more than 12 times lower than under the Overload profile, with the route, channel, and anomaly-mode parameters unchanged. This quantitatively confirms that, under mixed channel conditions, it is the thermal headroom of the Edge node's hardware platform, rather than the radio-channel characteristics, that determines whether the system enters a catastrophic state.

5.5. Traffic Savings from Hybrid Filtering

Figs. 7 and 8 show the volume of data queued for the Cloud and the savings share calculated per (17): HYBRID_FILTERED mode versus the PARALLEL_RAW baseline.

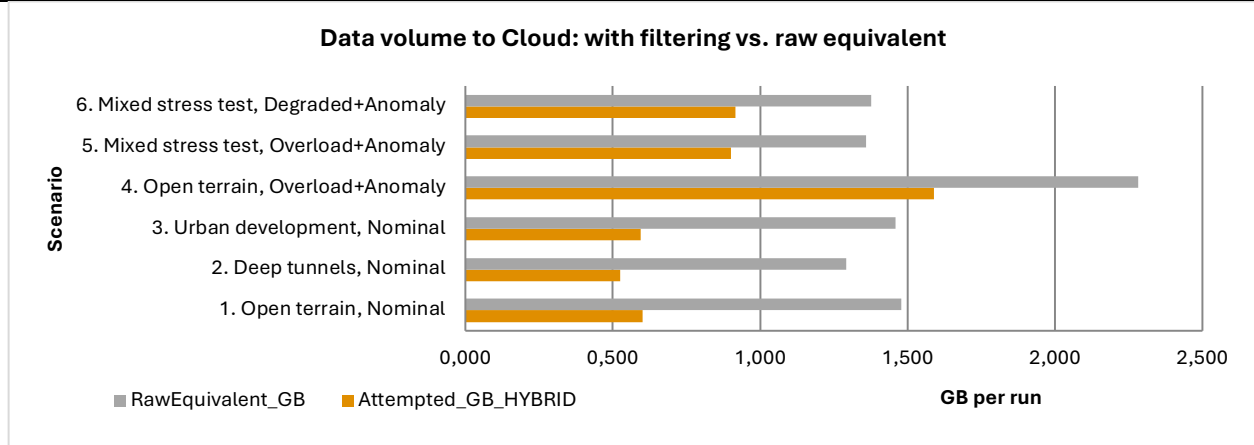


Fig. 7. Data volume queued to the Cloud per run: HYBRID_FILTERED (Attempted) versus the raw equivalent of PARALLEL_RAW (RawEquivalent), GB, across the six scenarios

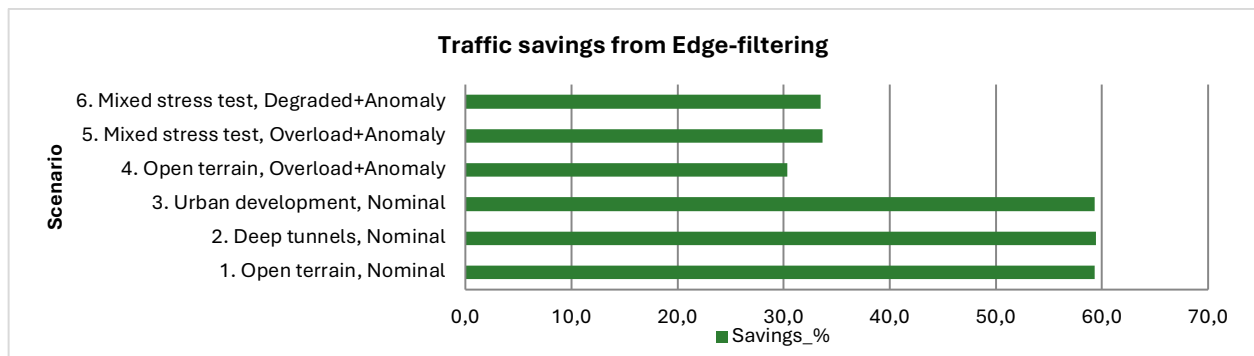


Fig. 8. Network-traffic savings from Edge filtering across the six factorial-design scenarios

Under nominal conditions (scenarios 1-3, without Anomaly Burst), the savings stably amount to $59,3 - 59,5 \%$; constant video compression (16) with a factor of $p_c = 0,05$ while the SAFETY class is still forwarded in full. Under an active Anomaly Burst (scenarios 4-6), the savings drop to $30,9 - 33,8 \%$, since during the anomaly the HYBRID_FILTERED mode (16) temporarily reverts to forwarding the full raw video stream for dispatcher verification, which partly offsets the filtering effect without eliminating it entirely, even given the observed anomaly-activity share of $15 - 21 \%$ of the time. The absolute volumes in Fig. 7 confirm the same effect in gigabytes: for scenario 1 (open terrain, Nominal), the volume of data queued to the Cloud drops from 1.479 GB (RawEquivalent) to 0.602 GB (Attempted) per run, and for scenario 4 (Overload+Anomaly) – from 2.282 GB to 1.516 GB (Table 5).

Table 2

Edge-subsystem indicators across the six scenarios, PARALLEL_RAW / HYBRID_FILTERED modes

Scenario	Mode	$\bar{L}_{edge}$, s	L_{edge}^{max} , s	Edge_PLR, %	T_{cpu}^{max} , s
----------	------	----------------------	----------------------	-------------	---------------------

5.6. Compute Energy Consumption Compared with Radio Transmission

Table 5 compares the physical energy (13) spent on Edge-node computation with the energy used by the radio transmitter to attempt data transfer to the Cloud (HYBRID_FILTERED). Across all six scenarios, the Edge node's compute energy ($1,85 - 2,89$ W·h per run) exceeds the radio-transmitter energy ($0,11 - 0,26$ W·h) by an order of magnitude: Edge accounts for $89 - 94 \%$ of the system's total energy consumption. The effect is amplified in the thermally loaded scenarios 4-5, where throttling extends the effective CPU active-work time per unit of usefully processed data.

5.7. Summary Result Tables

ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

					°C
1. Open terrain, Nominal	PARALLEL_RAW	0,000	0,000	0,0	29,72
1. Open terrain, Nominal	HYBRID_FILTERED	0,000	0,000	0,0	31,84
2. Deep tunnels, Nominal	PARALLEL_RAW	0,000	0,000	0,0	29,68
2. Deep tunnels, Nominal	HYBRID_FILTERED	0,000	0,000	0,0	31,85
3. Urban development, Nominal	PARALLEL_RAW	0,000	0,000	0,0	29,70
3. Urban development, Nominal	HYBRID_FILTERED	0,000	0,000	0,0	31,94
4. Open terrain, Overload+Anomaly	PARALLEL_RAW	11,655	27,124	32,8	78,53
4. Open terrain, Overload+Anomaly	HYBRID_FILTERED	16,082	41,574	47,1	79,48
5. Mixed stress test, Overload+Anomaly	PARALLEL_RAW	10,078	28,207	28,5	79,05
5. Mixed stress test, Overload+Anomaly	HYBRID_FILTERED	14,978	39,206	44,2	79,19
6. Mixed stress test, Degraded+Anomaly	PARALLEL_RAW	0,027	1,014	0,0	71,20
6. Mixed stress test, Degraded+Anomaly	HYBRID_FILTERED	0,030	1,057	0,0	73,91

Table 3

Cloud-subsystem indicators across the six scenarios, PARALLEL_RAW / HYBRID_FILTERED modes

Scenario	Mode	$\bar{L}_{cloud}$, s	Cloud_PLR, %
1. Open terrain, Nominal	PARALLEL_RAW	0,023	0,0
1. Open terrain, Nominal	HYBRID_FILTERED	0,023	0,0
2. Deep tunnels, Nominal	PARALLEL_RAW	0,772	49,5
2. Deep tunnels, Nominal	HYBRID_FILTERED	0,772	49,3
3. Urban development, Nominal	PARALLEL_RAW	0,025	0,0
3. Urban development, Nominal	HYBRID_FILTERED	0,025	0,0
4. Open terrain, Overload+Anomaly	PARALLEL_RAW	0,027	0,0
4. Open terrain, Overload+Anomaly	HYBRID_FILTERED	0,027	0,0
5. Mixed stress test, Overload+Anomaly	PARALLEL_RAW	0,441	26,4
5. Mixed stress test, Overload+Anomaly	HYBRID_FILTERED	0,441	24,7
6. Mixed stress test, Degraded+Anomaly	PARALLEL_RAW	0,429	27,1
6. Mixed stress test, Degraded+Anomaly	HYBRID_FILTERED	0,429	28,5

Table 3 confirms a mirror image of Table 2: the Cloud-segment latency remains low (0.02-0.44 s) under any Edge thermal profile, and the architecture mode (PARALLEL_RAW/HYBRID_FILTERED) has practically no effect on it – unlike Edge latency, which rises sharply under thermal throttling. The only factor that raises Cloud_PLR is radio-channel degradation: 49.3-

49.5 % in the tunnel scenario versus 0 % in open terrain and in the city, and these values likewise do not depend on the architecture mode, since both modes forward the SAFETY class in full. This confirms the conclusion of Section 6.1 that the Cloud bottleneck is external (the channel) rather than internal (compute), as it is for Edge.

Table 4

Share of time the system spends in each of the four states (18), %, HYBRID_FILTERED mode

Scenario	Both OK	Cloud Down	Edge Down	Both Down
1. Open terrain, Nominal	99,82	0,18	0,00	0,00
2. Deep tunnels, Nominal	17,83	82,17	0,00	0,00
3. Urban development, Nominal	99,71	0,29	0,00	0,00
4. Open terrain, Overload+Anomaly	11,46	0,02	87,91	0,61
5. Mixed stress test, Overload+Anomaly	15,84	0,93	8,51	74,72
6. Mixed stress test, Degraded+Anomaly	22,68	69,44	1,71	6,17

Table 5

Scenario	Attempted, GB	RawEquivalent, GB	Savings, %	Edge, W·h	Cloud TX, W·h	Edge share, %
1. Open terrain, Nominal	0,602	1,479	59,33	2,116	0,257	89,17
2. Deep tunnels, Nominal	0,371	0,915	59,45	1,847	0,112	94,29
3. Urban development, Nominal	0,213	0,524	59,32	2,089	0,253	89,19
4. Open terrain, Overload+Anomaly	1,516	2,282	33,56	2,889	0,257	91,83
5. Mixed stress test, Overload+Anomaly	0,671	0,971	30,93	1,921	0,122	94,03
6. Mixed stress test, Degraded+Anomaly	0,644	0,972	33,80	1,912	0,122	94,00

Discussion of Results

6.1. The Mirror-Image, Not Universal, Nature of Edge and Cloud Advantages

The results of Section 5 refute the simplified view of Edge computing as an unconditionally “faster” alternative to Cloud. A direct comparison of Sections 5.2 and 5.3 demonstrates a mirror-image symmetry of failures: in the tunnel scenario, Edge remains flawless ($Edge_PLR = 0\%$), while Cloud degrades catastrophically ($Cloud_PLR \approx 49\%$) due to the physical absence of a communication channel; in the thermal-overload scenario, the situation reverses – Cloud remains flawless, while Edge degrades ($Edge_PLR$ to 47%) due to nonlinear thermal throttling (11). The nature of the bottleneck is fundamentally different for each architecture: for Cloud it is an external, physically uncontrollable factor (radio-channel availability along the route), while for Edge it is an internal, hardware-determined factor (the thermal capacity and cooling power of the on-board computer). This has direct engineering implications: channel degradation cannot be eliminated by software means (its consequences can only be mitigated through buffering or proactive compression), whereas Edge degradation can be designed out entirely through an adequate thermal-power margin – which is directly confirmed by the contrast between the Overload and Degraded profiles (Section 5.4): under identical load, changing only the thermal headroom reduces the share of the critical state *Both Down* by more than a factor of 12.

The speed of Edge’s response (latency close to zero under nominal load) is physically due to the absence of an intermediate radio channel and server segment (15) in the processing chain: data are processed where they are generated. Cloud’s advantage, by contrast, lies not in speed but in the scalability of long-term analysis – the server segment (15), with a throughput of $\mu_{srv} = 500$

Mbit/s, introduced in the model as a multi-tenant resource, is able to accumulate and process historical data from many trains simultaneously, something a computationally and energy-constrained on-board node fundamentally cannot provide (Section 6.3).

6.2. The “Distributed Infrastructure versus Traffic Volume” Trade-off

The hybrid architecture (16) delivers a measurable reduction in network traffic ($59,3 - 59,5\%$ under nominal conditions, Section 5.5), which directly reduces the load on the radio channel and the network operator’s potential data-transmission costs. However, this saving is not free: local filtering requires running ML inference directly on board, which measurably raises CPU utilization ($4U_{ML} = 0,10$, equation (11)) and, as a consequence, the peak temperature and frequency of thermal throttling (Section 5.3). In scenario 4, this manifests quantitatively: the average Edge latency rises from $11,66$ s (PARALLEL_RAW) to $16,08$ s (HYBRID_FILTERED), and $Edge_PLR$ rises from $32,8\%$ to $47,1\%$. This is a classic engineering trade-off: the complexity (and energy cost, Section 6.3) of maintaining a distributed on-board computing infrastructure grows in proportion to the functionality delegated to it, and this cost must be weighed together with – not separately from – the choice of Edge-node hardware profile and the decision to deploy hybrid filtering.

6.3. The Paradox of Higher Energy Cost for Computation Compared with Radio Transmission

The result of Section 5.6 – that $89 - 94\%$ of the system’s total energy consumption is attributable to local Edge computation rather than to radio transmission – contradicts the common intuition that “data transmission is energy-expensive, so it is better to

compute locally.” Within the model adopted here, the opposite holds: computation is an order of magnitude more energy-costly than transmitting the same volume of data. This does not mean that Edge computation is architecturally disadvantageous – it remains necessary for latency and resilience to connectivity loss (Section 6.1) – but it does mean that the argument “Edge saves energy” is not supported by the present model and should not be used as a standalone justification for a hybrid architecture; the correct justification is precisely the traffic savings (Section 5.5) and resilience to channel degradation (Section 5.2), not energy efficiency.

6.4. Recommendation for a Redundant Hybrid Architecture

The absence of additivity in failures (Section 5.4: the observed share *Both Down* $\approx 0,747$ is close to the product of the isolated failure probabilities $0,822 \times 0,885 \approx 0,728$ rather than to their sum) means that, for end-to-end system reliability, simultaneous independent degradation is the worst-case scenario, whose probability grows multiplicatively. This leads directly to a practical conclusion: the monitoring system should treat Edge and Cloud as **parallel, mutually redundant channels** for delivering critical information, rather than as a sequential “primary-backup” pair. In practice, this means:

- SAFETY-class alerts (1) must be processed locally on Edge regardless of the state of the communication channel – this is the only way to guarantee a response while traversing tunnels (Section 5.2), when Cloud_PLR exceeds the safety threshold by more than a factor of 24;
- the Edge node must have a mechanism for proactively shedding load (temporarily reducing the video sampling rate, or a priority switch to a “SAFETY-only” mode) as T_{cpu} approaches T_{crit} since once the threshold is crossed, degradation becomes self-reinforcing (11): rising latency means a growing queue backlog, which sustains high utilization and temperature above T_{crit} ;
- the choice of the Edge node’s hardware thermal profile is a design decision at least as important as the data-processing algorithmic strategy (Section 5.4), and must account for the additional load from hybrid-filtering ML inference (Section 6.2), not merely the baseline monitoring traffic.

6.5. Impact on the Railway’s Server Infrastructure

Introducing the Cloud server segment (15) with a finite throughput μ_{srv} into the model reflects a real-world constraint: the monitoring system’s backend serves not a single train but an entire fleet simultaneously. The traffic savings from hybrid filtering (Section 5.5) directly reduce the load on this shared resource – scaled to a fleet of N trains, a proportional reduction of the input flow by a factor of $1 - \text{savings}$ makes it possible either to serve a larger fleet with the same backend or to relax its

throughput requirements. At the same time, since the cloud segment remains necessary for long-term trend analysis and retraining of predictive-maintenance models (unlike Edge, which is limited by compute power and buffer capacity), the hybrid model does not eliminate dependence on the server infrastructure but redistributes its load in favor of compressed, pre-filtered aggregated telemetry, reserving the full raw stream only for the verification of confirmed anomalies.

6.6. Study Limitations

The factorial design is a targeted, not an exhaustive, factorial sample (six of the nine $\{\text{route}\} \times \{\text{profile}\}$ combinations were examined); the model considers a single train in isolation, without inter-train interference under dense traffic; the thermal model is single-reservoir (lumped RC); the stochastic components (2)-(10) were chosen based on qualitative assumptions about a plausible range of variability rather than a calibrated dataset of real rolling-stock telemetry, so the absolute values should be regarded as comparative (for contrasting architectures under identical conditions) rather than as a precise forecast of the performance of a specific production system.

Conclusion

This paper builds and verifies a discrete-time digital twin of a train that makes it possible to quantitatively compare Edge- and Cloud-oriented processing of HSR rolling-stock monitoring data within a single, consistent set of stochastic conditions (radio channel, thermal regime, event-driven load). In contrast to qualitative claims about the merits of each architecture, concrete quantitative estimates were obtained: Edge delivers near-zero latency and complete independence from the radio channel (in the tunnel scenario, $\text{Cloud_PLR} \approx 49\%$ versus $\text{Edge_PLR} = 0\%$), whereas Cloud is independent of the on-board computer’s thermal state (in the thermal-overload scenario, Edge_PLR to 47% versus $\text{Cloud_PLR} = 0\%$). The simultaneous action of both degradation factors produces a share of the critical state of *Both Down* to $74,7\%$, whereas increasing the thermal headroom of the hardware platform alone is sufficient to reduce this share by more than a factor of 12 – this quantitatively confirms the hypothesis that the reliability of the monitoring system is determined not by choosing a single “better” architecture, but by the redundant interaction of both.

The empirically evaluated hybrid architecture, in which Edge pre-filters monitoring traffic, delivers network-traffic savings of $59,3 - 59,5\%$ under nominal conditions and $30,9 - 33,8\%$ under active anomalous load, confirming the initial hypothesis regarding the feasibility of hybrid computing for this class of systems. At the same time, two quantitative trade-offs were identified that require explicit consideration in system

design: (1) the computational load of local ML inference measurably degrades the Edge node's thermal headroom (peak temperature rises from $78,5$ to $79,5^{\circ}\text{C}$, and average latency from $11,66$ to $16,08$ s in the most heavily loaded scenario); (2) local computation proved to be an order of magnitude more energy-costly than radio transmission of the same volume of data ($89 - 94\%$ of the system's total energy consumption), which refutes energy efficiency as a standalone argument in favor of Edge and leaves traffic savings and resilience to connectivity loss as the only valid justification for the hybrid approach.

Future research should aim to extend the factorial design to a full enumeration of {route} $\times$ {profile} $\times$ {anomaly} combinations for rigorous variance decomposition of each factor's contribution; to validate the radio-channel and thermal-model parameters against real telemetry from operating rolling stock; to apply Federated Learning for distributed training of predictive-maintenance models without transmitting private raw train data to the Cloud, which naturally aligns with the conclusion that the system's dependence on constant availability of an external communication channel must be minimized; and to develop V2V (Train-to-Train) communications as a third, fully decentralized redundancy channel capable of detecting and propagating critical events even when both the Cloud and the nearest train's own Edge node are simultaneously unavailable.

Список використаних джерел

1. Railnova. (n.d.). *How edge computing revolutionises railway operations*. <https://www.railnova.eu/how-edge-computing-revolutionises-railway-operations/>
2. ICOS Project. (n.d.). *Cloud-edge-IoT solutions: Present and future for railway sector*. <https://www.icos-project.eu/blog/cei-in-railway-sector/>
3. He, R., Ai, B., Wang, G., Guan, K., Zhong, Z., Molisch, A. F., Briso-Rodríguez, C., & Oestges, C. (2016). High-speed railway communications: From GSM-R to LTE-R. *IEEE Vehicular Technology Magazine*, 11(3), 49–58. <https://doi.org/10.1109/MVT.2016.2564446>
4. Tata Consultancy Services (TCS). (n.d.). *IoT-based predictive maintenance for smart railroad asset management* [White paper]. <https://www.tcs.com/content/dam/global-tcs/en/pdfs/insights/whitepapers/Transforming-Railroad-Asset-Management.pdf>
5. Gbadamosi, A. Q., Oyedele, L. O., Delgado, J. M. D., Kusimo, H., Akanbi, L., Olawale, O., & Muhammed-Yakubu, N. (2021). IoT for predictive assets monitoring and maintenance: An implementation strategy for the UK rail industry. *Automation in Construction*, 122, 103486. <https://doi.org/10.1016/j.autcon.2020.103486>
6. IoT For All. (n.d.). *How the railroad industry is leveraging IoT edge computing*. <https://www.iotforall.com/how-the-rail-sector-is-leveraging-iot-edge-computing>
7. Kostrzewski, M., & Melnik, R. (2021). Condition monitoring of rail transport systems: A bibliometric performance analysis and systematic literature review. *Sensors*, 21(14), 4710. <https://doi.org/10.3390/s21144710>
8. Davari, N., Veloso, B., de Assis Costa, G., Mota Pereira, P., Ribeiro, R. P., & Gama, J. (2021). A survey on data-driven predictive maintenance for the railway industry. *Sensors*, 21(17), 5739. <https://doi.org/10.3390/s21175739>
9. Pappaterra, M. J., Flammini, F., Vittorini, V., & Bešinović, N. (2021). A systematic review of artificial intelligence public datasets for railway applications. *Infrastructures*, 6(10), 136. <https://doi.org/10.3390/infrastructures6100136>
10. Zhang, D., Peng, Y., Xu, Y., Du, C., Zhang, Y., Wang, N., Chong, Y., Wang, H., Wu, D., Liu, J., Zhang, H., Lu, L., & Liu, J. (2022). A high-speed railway network dataset from train operation records and weather data. *Scientific Data*, 9(1), 244. <https://doi.org/10.1038/s41597-022-01349-8>
11. Zhou, X., Ke, R., Yang, H., & Liu, C. (2021). When intelligent transportation systems sensing meets edge computing: Vision and challenges. *Applied Sciences*, 11(20), 9680. <https://doi.org/10.3390/app11209680>
12. Resende, C., Folgado, D., Oliveira, J., Pinto, F., Santos, M. Y., & Costa, C. (2021). TIP4.0: Industrial Internet of Things platform for predictive maintenance. *Sensors*, 21(14), 4676. <https://doi.org/10.3390/s21144676>
13. Armijo, A., & Zamora-Sánchez, D. (2024). Integration of railway bridge structural health monitoring into the Internet of Things with a digital twin: A case study. *Sensors*, 24(7), 2115. <https://doi.org/10.3390/s24072115>
14. Atanasov, I., Pencheva, E., Trifonov, V., & Kassev, K. (2024). Railway cloud: Management and orchestration functionality designed as microservices. *Applied Sciences*, 14(6), 2368. <https://doi.org/10.3390/app14062368>
15. He, R., Ai, B., Zhong, Z., Yang, M., Chen, R., Ding, J., Ma, Z., Sun, G., & Liu, C. (2022). 5G for railways: The next generation railway dedicated communications. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2207.03127>
16. He, R., Ai, B., Zhong, Z., Yang, M., Huang, C., Chen, R., Ding, J., Mi, H., Ma, Z., Sun, G., & Liu, C. (2023). Radio communication scenarios in 5G-railways. *China Communications*, 20(9), 235–246. <https://doi.org/10.23919/JCC.2023.0000xx>
17. Wang, C., Ji, W., Zheng, G., & Saleem, A. (2021). Analysis of propagation characteristics for various subway tunnel scenarios at 28 GHz. *International Journal of Antennas and Propagation*, 2021, 7666624. <https://doi.org/10.1155/2021/7666624>
18. Clérigo, A., Schrapel, M., Rito, P., Sargento, S., & Vinel, A. (2025). Microservice-based

- architecture for enhancing road safety with support for low-latency services. *Proceedings of the 2025 IEEE/IFIP Network Operations and Management Symposium (NOMS 2025)*, 1–7. <https://doi.org/10.1109/NOMS57970.2025.11073709>
19. Xu, J., Wei, Z., Lyu, Z., Shi, L., & Han, J. (2021). Throughput maximization of offloading tasks in multi-access edge computing networks for high-speed railways. *IEEE Transactions on Vehicular Technology*, 70(9), 9525–9539. <https://doi.org/10.1109/TVT.2021.3101571>
20. Atik, D., Gursu, M., Mehmeti, F., & Kellerer, W. (2024). Rural handover parameter tuning to achieve end to end latency requirements of future railway mobile communication systems. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2408.06957>
21. Arthurs, P., Gillam, L., Krause, P., Wang, N., Halder, K., & Mouzakitis, A. (2022). A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 6206–6221. <https://doi.org/10.1109/TITS.2021.3084396>
22. Khan, A., Khattak, K. S., Gul, H., Alqahtani, H. S., Alqahtani, A. S., & Gumaei, A. (2023). Edge computing for effective and efficient traffic characterization. *Sensors*, 23(23), 9385. <https://doi.org/10.3390/s23239385>
23. Abdullah, M. F. A., Yogarayan, S., Abdul Razak, S. F., Azman, A., Muhamad Amin, A. H., & Salleh, M. (2021). Edge computing for vehicle to everything: A short review. *F1000Research*, 10, 1104. <https://doi.org/10.12688/f1000research.73269.4>
24. Vermesan, O., John, R., Pype, P., Daalderop, G., Kriegel, K., Mitic, G., Lorentz, V. R. H., Bahr, R., Sand, H.-E., Bockrath, S., & Waldhör, S. (2021). Automotive intelligence embedded in electric connected autonomous and shared vehicles technology for sustainable green mobility. *Frontiers in Future Transportation*, 2, 688482. <https://doi.org/10.3389/ffutr.2021.688482>
25. Zhang, K., Huang, W., Hou, X., Xu, J., Su, R., & Xu, H. (2021). A fault diagnosis and visualization method for high-speed train based on edge and cloud collaboration. *Applied Sciences*, 11(3), 1251. <https://doi.org/10.3390/app11031251>
26. Sun, P., Zhang, W., & Jia, Z. (2021). High-speed train intelligent maintenance pattern and edge-computing perception model. *Proceedings of the 2021 IEEE 23rd International Conference on High Performance Computing and Communications; IEEE 19th International Conference on Smart City; IEEE 7th International Conference on Data Science and Systems (HPCC/DSS/SmartCity/DependSys 2021)*, 1331–1337. <https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00201>
27. Asad, S. M., Tahir, A., Bin Rais, R. N., Ansari, S., Abubakar, A. I., Hussain, S., & Imran, M. A. (2021). Edge intelligence in private mobile networks for next-generation railway systems. *Frontiers in Communications and Networks*, 2, 769299. <https://doi.org/10.3389/frcmn.2021.769299>
28. Zhao, Y., Yu, X., Chen, M., Zhang, M., Chen, Y., Niu, X., Sha, X., Zhan, Z., & Li, W. (2021). Continuous monitoring of train parameters using IoT sensor and edge computing. *IEEE Sensors Journal*, 21(14), 15458–15468. <https://doi.org/10.1109/JSEN.2020.3026643>
29. Wisultschew, C., Mujica, G., Lanza-Gutierrez, J. M., & Portilla, J. (2021). 3D-LIDAR based object detection and tracking on the edge of IoT for railway level crossing. *IEEE Access*, 9, 35718–35729. <https://doi.org/10.1109/ACCESS.2021.3062706>
30. Guo, Q., Xu, Z., Yuan, J., & Wei, Y. (2024). Computation offloading strategy for detection task in railway IoT with integrated sensing, storage, and computing. *Electronics*, 13(15), 2982. <https://doi.org/10.3390/electronics13152982>
31. Xu, J., Ai, B., Chen, L., Cui, Y., & Wang, N. (2022). Deep reinforcement learning for computation and communication resource allocation in multiaccess MEC assisted railway IoT networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(24), 23797–23808. <https://doi.org/10.1109/TITS.2022.3205175>
32. Zhang, X. (2023). Optimization design of railway logistics center layout based on mobile cloud edge computing. *PeerJ Computer Science*, 9, e1298. <https://doi.org/10.7717/peerj-cs.1298>

References

1. Railnova. (n.d.). *How edge computing revolutionises railway operations*. <https://www.railnova.eu/how-edge-computing-revolutionises-railway-operations/> [in English].
2. ICOS Project. (n.d.). *Cloud-edge-IoT solutions: Present and future for railway sector*. <https://www.icos-project.eu/blog/cei-in-railway-sector/> [in English].
3. He, R., Ai, B., Wang, G., Guan, K., Zhong, Z., Molisch, A. F., Briso-Rodríguez, C., & Oestges, C. (2016). High-speed railway communications: From GSM-R to LTE-R. *IEEE Vehicular Technology Magazine*, 11(3), 49–58. <https://doi.org/10.1109/MVT.2016.2564446> [in English].
4. Tata Consultancy Services (TCS). (n.d.). *IoT-based predictive maintenance for smart railroad asset management* [White paper]. <https://www.tcs.com/content/dam/global-tcs/en/pdfs/insights/whitepapers/Transforming-Railroad-Asset-Management.pdf> [in English].
5. Gbadamosi, A. Q., Oyedele, L. O., Delgado, J. M. D., Kusimo, H., Akanbi, L., Olawale, O., & Muhammed-Yakubu, N. (2021). IoT for predictive assets monitoring and maintenance: An implementation strategy for the UK rail industry. *Automation in Construction*, 122, 103486. <https://doi.org/10.1016/j.autcon.2020.103486> [in English].

6. IoT For All. (n.d.). *How the railroad industry is leveraging IoT edge computing*. <https://www.iotforall.com/how-the-rail-sector-is-leveraging-iot-edge-computing> [in English].
7. Kostrzewski, M., & Melnik, R. (2021). Condition monitoring of rail transport systems: A bibliometric performance analysis and systematic literature review. *Sensors*, 21(14), 4710. <https://doi.org/10.3390/s21144710> [in English].
8. Davari, N., Veloso, B., de Assis Costa, G., Mota Pereira, P., Ribeiro, R. P., & Gama, J. (2021). A survey on data-driven predictive maintenance for the railway industry. *Sensors*, 21(17), 5739. <https://doi.org/10.3390/s21175739> [in English].
9. Pappaterra, M. J., Flammini, F., Vittorini, V., & Bešinović, N. (2021). A systematic review of artificial intelligence public datasets for railway applications. *Infrastructures*, 6(10), 136. <https://doi.org/10.3390/infrastructures6100136> [in English].
10. Zhang, D., Peng, Y., Xu, Y., Du, C., Zhang, Y., Wang, N., Chong, Y., Wang, H., Wu, D., Liu, J., Zhang, H., Lu, L., & Liu, J. (2022). A high-speed railway network dataset from train operation records and weather data. *Scientific Data*, 9(1), 244. <https://doi.org/10.1038/s41597-022-01349-8> [in English].
11. Zhou, X., Ke, R., Yang, H., & Liu, C. (2021). When intelligent transportation systems sensing meets edge computing: Vision and challenges. *Applied Sciences*, 11(20), 9680. <https://doi.org/10.3390/app11209680> [in English].
12. Resende, C., Folgado, D., Oliveira, J., Pinto, F., Santos, M. Y., & Costa, C. (2021). TIP4.0: Industrial Internet of Things platform for predictive maintenance. *Sensors*, 21(14), 4676. <https://doi.org/10.3390/s21144676> [in English].
13. Armijo, A., & Zamora-Sánchez, D. (2024). Integration of railway bridge structural health monitoring into the Internet of Things with a digital twin: A case study. *Sensors*, 24(7), 2115. <https://doi.org/10.3390/s24072115> [in English].
14. Atanasov, I., Pencheva, E., Trifonov, V., & Kassev, K. (2024). Railway cloud: Management and orchestration functionality designed as microservices. *Applied Sciences*, 14(6), 2368. <https://doi.org/10.3390/app14062368> [in English].
15. He, R., Ai, B., Zhong, Z., Yang, M., Chen, R., Ding, J., Ma, Z., Sun, G., & Liu, C. (2022). 5G for railways: The next generation railway dedicated communications. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2207.03127> [in English].
16. He, R., Ai, B., Zhong, Z., Yang, M., Huang, C., Chen, R., Ding, J., Mi, H., Ma, Z., Sun, G., & Liu, C. (2023). Radio communication scenarios in 5G-railways. *China Communications*, 20(9), 235–246. <https://doi.org/10.23919/JCC.2023.0000xx> [in English].
17. Wang, C., Ji, W., Zheng, G., & Saleem, A. (2021). Analysis of propagation characteristics for various subway tunnel scenarios at 28 GHz. *International Journal of Antennas and Propagation*, 2021, 7666624. <https://doi.org/10.1155/2021/7666624> [in English].
18. Clérigo, A., Schrapel, M., Rito, P., Sargento, S., & Vinel, A. (2025). Microservice-based architecture for enhancing road safety with support for low-latency services. *Proceedings of the 2025 IEEE/IFIP Network Operations and Management Symposium (NOMS 2025)*, 1–7. <https://doi.org/10.1109/NOMS57970.2025.11073709> [in English].
19. Xu, J., Wei, Z., Lyu, Z., Shi, L., & Han, J. (2021). Throughput maximization of offloading tasks in multi-access edge computing networks for high-speed railways. *IEEE Transactions on Vehicular Technology*, 70(9), 9525–9539. <https://doi.org/10.1109/TVT.2021.3101571> [in English].
20. Atik, D., Gursu, M., Mehmeti, F., & Kellerer, W. (2024). Rural handover parameter tuning to achieve end to end latency requirements of future railway mobile communication systems. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2408.06957> [in English].
21. Arthurs, P., Gillam, L., Krause, P., Wang, N., Halder, K., & Mouzakitis, A. (2022). A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 6206–6221. <https://doi.org/10.1109/TITS.2021.3084396> [in English].
22. Khan, A., Khattak, K. S., Gul, H., Alqahtani, H. S., Alqahtani, A. S., & Gumaei, A. (2023). Edge computing for effective and efficient traffic characterization. *Sensors*, 23(23), 9385. <https://doi.org/10.3390/s23239385> [in English].
23. Abdullah, M. F. A., Yogarayan, S., Abdul Razak, S. F., Azman, A., Muhamad Amin, A. H., & Salleh, M. (2021). Edge computing for vehicle to everything: A short review. *F1000Research*, 10, 1104. <https://doi.org/10.12688/f1000research.73269.4> [in English].
24. Vermesan, O., John, R., Pype, P., Daalderop, G., Kriegel, K., Mitic, G., Lorentz, V. R. H., Bahr, R., Sand, H.-E., Bockrath, S., & Waldhör, S. (2021). Automotive intelligence embedded in electric connected autonomous and shared vehicles technology for sustainable green mobility. *Frontiers in Future Transportation*, 2, 688482. <https://doi.org/10.3389/ffutr.2021.688482> [in English].
25. Zhang, K., Huang, W., Hou, X., Xu, J., Su, R., & Xu, H. (2021). A fault diagnosis and visualization method for high-speed train based on edge and cloud collaboration. *Applied Sciences*, 11(3), 1251. <https://doi.org/10.3390/app11031251> [in English].
26. Sun, P., Zhang, W., & Jia, Z. (2021). High-speed train intelligent maintenance pattern and edge-computing perception model. *Proceedings of the 2021 IEEE 23rd International Conference on High Performance Computing and Communications; IEEE 19th International Conference on Smart City; IEEE 7th International Conference on Data Science and Systems (HPCC/DSS/SmartCity/DependSys 2021)*, 1331–1337.

<https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00201> [in English].

27. Asad, S. M., Tahir, A., Bin Rais, R. N., Ansari, S., Abubakar, A. I., Hussain, S., & Imran, M. A. (2021). Edge intelligence in private mobile networks for next-generation railway systems. *Frontiers in Communications and Networks*, 2, 769299. <https://doi.org/10.3389/frcmn.2021.769299> [in English].

28. Zhao, Y., Yu, X., Chen, M., Zhang, M., Chen, Y., Niu, X., Sha, X., Zhan, Z., & Li, W. (2021). Continuous monitoring of train parameters using IoT sensor and edge computing. *IEEE Sensors Journal*, 21(14), 15458–15468. <https://doi.org/10.1109/JSEN.2020.3026643> [in English].

29. Wisultschew, C., Mujica, G., Lanza-Gutierrez, J. M., & Portilla, J. (2021). 3D-LIDAR based object detection and tracking on the edge of IoT for railway level crossing. *IEEE Access*, 9, 35718–35729. <https://doi.org/10.1109/ACCESS.2021.3062706> [in English].

30. Guo, Q., Xu, Z., Yuan, J., & Wei, Y. (2024). Computation offloading strategy for detection task in railway IoT with integrated sensing, storage, and computing. *Electronics*, 13(15), 2982. <https://doi.org/10.3390/electronics13152982> [in English].

31. Xu, J., Ai, B., Chen, L., Cui, Y., & Wang, N. (2022). Deep reinforcement learning for computation and communication resource allocation in multiaccess MEC assisted railway IoT networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(24), 23797–23808. <https://doi.org/10.1109/TITS.2022.3205175> [in English].

32. Zhang, X. (2023). Optimization design of railway logistics center layout based on mobile cloud edge computing. *PeerJ Computer Science*, 9, e1298. <https://doi.org/10.7717/peerj-cs.1298> [in English].

Порівняльний аналіз архітектур периферійних і хмарних обчислень для систем моніторингу рухомого складу

Анотація

У статті наведено кількісне порівняння оброблення даних моніторингу технічного стану рухомого складу високошвидкісного залізничного транспорту (HSR), орієнтованого на Edge і Cloud, на основі верифікованого дискретно-часового цифрового двійника поїзда. На відміну від підходів, у яких Edge і Cloud розглянуто як окремі незалежні конфігурації, запропонована модель одночасно обчислює обидва конвеєри оброблення в межах одного запуску, включаючи стохастичний радіоканал (корельоване замирання, погіршення характеристик під впливом ефекту Доплера, HARQ із залишковою помилкою блоку), пріоритетне планування двох класів трафіку (критичного для безпеки та моніторингового), а також теплову модель бортового комп'ютера з механізмом тротлінгу. Додатково реалізовано та експериментально оцінено гібридний режим

архітектури, у якому Edge попередньо фільтрує моніторинговий трафік перед його передаванням до Cloud. Повний факторний план експерименту показує, що дві архітектури мають принципово різні механізми виникнення вузьких місць: Cloud деградує через фізичну недоступність радіоканалу, тоді як Edge деградує через тепловий тротлінг за недостатнього охолодження. Одночасна дія обох факторів зумовлює значну частку критичного стану *Both Down* від загального часу спостереження, тоді як лише збільшення теплового запасу апаратної платформи дає змогу зменшити цю частку в декілька разів. Гібридна фільтрація забезпечує вимірюване зменшення мережевого трафіку як за номінальних умов, так і за активного аномального навантаження, проте супутнє обчислювальне навантаження ML-інференсу помітно зменшує тепловий запас вузла Edge. Локальні обчислення виявилися суттєво більш енерговитратними, ніж радіопередавання такого самого обсягу даних. Отримані результати кількісно обґрунтовують необхідність резервованої гібридної архітектури моніторингу, у якій жодна окрема підсистема не є єдиною точкою відмови.

Ключові слова: Edge computing, Cloud computing, цифровий двійник, моніторинг рухомого складу, IoT, прогнозне технічне обслуговування, Age of Information, QoS, залізничний транспорт.

Mykola KUSHCH, Master of Science. ORCID: 0009-0009-3078-4671.

Ukrainian state university of railway transport, Kharkiv, Ukraine.

E-mail: dev.kusch@gmail.com

Oleh ABBASOV, Master of Science. ORCID: 0009-0004-1299-5903.

Ukrainian state university of railway transport, Kharkiv, Ukraine.

E-mail: abbasov.oleg@gmail.com

Olena SHCHEBLYKINA, PhD, Associate Professor, ORCID: 0000-0002-8304-2921.

Ukrainian state university of railway transport, Kharkiv, Ukraine.

E-mail: sov@kart.edu.ua

Maksym KANEVSKYI, Master of Science. ORCID: 0009-0007-6284-0311.

Ukrainian state university of railway transport, Kharkiv, Ukraine.

E-mail: kanevskij@gmail.com

Микола КУЩ, магістр. ORCID: 0009-0009-3078-4671.

Український державний університет залізничного транспорту, Харків, Україна.

E-mail: dev.kusch@gmail.com

Олег АББАСОВ, магістр. ORCID: 0009-0004-1299-5903.

Український державний університет залізничного транспорту, Харків, Україна.

E-mail: abbasov.oleg@gmail.com

Олена ЩЕБЛИКІНА, PhD, доцент кафедри автоматизації та комп'ютерного телекерування рухом поїздів.

. ORCID: 0000-0002-8304-2921.

Український державний університет залізничного транспорту, Харків, Україна.

E-mail: sov@kart.edu.ua

Максим КАНЄВСЬКИЙ, магістр. ORCID: 0009-0007-6284-0311.

Український державний університет залізничного транспорту, Харків, Україна.

E-mail: kanevskij@gmail.com

Стаття надійшла 19.08.26

Стаття прийнята до друку після рецензування 13.09.26

Стаття опублікована (оприлюднена) 06.10.26

Стаття поширюється на умовах ліцензії Creative Commons Attribution License International CC-BY.