Article

Not peer-reviewed version

A Multimodal TinyML-Based Predictive Maintenance Architecture for Industrial IoT in the 6G Era

Carlos Exequiel Garay , Fernando Alberto Miranda Bonomi , Gonzalo Nicolás Mansilla , Mariano Fagre , Sergio Gustavo Guzmán , Pablo Alberto Ritorto , Franco Ismael Perez , Marcos Katz *

Posted Date: 17 June 2026

doi: 10.20944/preprints202606.1304.v1

Keywords: TinyML; predictive maintenance; anomaly detection; multimodal sensing; thermography; 6G; URLLC; Industry 5.0; sensor fusion; experimental validation

 doi:10.20944/preprints202606.1304.v1

Article

A Multimodal TinyML-Based Predictive Maintenance Architecture for Industrial IoT in the 6G Era

Carlos Exequiel Garay [1,2], Fernando Alberto Miranda Bonomi [2], Gonzalo Nicolás Mansilla [1], Mariano Fagre [2,5], Sergio Gustavo Guzmán [3], Pablo Alberto Ritorto [3], Franco Ismael Perez [3] and Marcos Katz [4,*]

[1] CIASUR (Centro de Investigación de Atmósfera Superior y Radiopropagación), Facultad Regional, Tucumán (FRT), Universidad Tecnológica Nacional (UTN), Rivadavia 1050, San Miguel de Tucumán 4000, Argentina

[2] Laboratorio de Telecomunicaciones (LTC), Departamento de Electricidad, Electrónica y Computación, Facultad de Ciencias Exactas y Tecnología (FACET), Universidad Nacional de Tucumán (UNT), Tucumán 4000, Argentina

[3] Laboratorio Área IV – Termología, Departamento de Mecánica, Facultad Regional Tucumán (FRT), Universidad Tecnológica Nacional (UTN), Rivadavia 1050, San Miguel de Tucumán 4000, Argentina

[4] Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland

[5] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Tucumán 4000, Argentina

* Correspondence: marcos.katz@oulu.fi

Abstract

Predictive maintenance (PdM) is central to Industry 5.0 strategies for reducing unplanned downtime in rotating machinery. This work proposes and evaluates a multimodal edge architecture for PdM that combines TinyML inference at the sensor with industrial IoT connectivity, designed to remain stable across the application-plane evolution toward sixth-generation (6G) networks. Three complementary modalities are deployed on commercial off-the-shelf hardware: vibration, acoustic and thermography, each running local inference on a smart sensor node, with an embedded gateway bridging per-modality decisions to a serverless cloud back-end. On real vibration data from a controlled static-unbalance testbed, five anomaly-detection algorithms are benchmarked on the actual Cortex-M4F target: an INT8-quantized fully connected autoencoder reaches F1 = 0.9976 with 254 μs inference latency and a 6,056 B Flash footprint, well within the microcontroller budget. A preliminary intra-session late-fusion analysis suggests that a logistic-regression meta-learner over the three modality scores improves on single-modality baselines, motivating multimodal sensing; cross-session generalization is left to future work. An end-to-end latency experiment shows that the cloud-uplink leg dominates the budget (79–88 %), establishing edge-first inference as a necessary condition for 6G URLLC gains to be observable at the application level.

Keywords: TinyML; predictive maintenance; anomaly detection; multimodal sensing; thermography; 6G; URLLC; Industry 5.0; sensor fusion; experimental validation

1. Introduction

Maintenance has evolved from corrective (run-to-failure) to preventive maintenance (PM) and now to predictive maintenance (PdM), which anticipates degradation through continuous monitoring of physical variables. PdM extends asset life, lowers operating costs and reduces unplanned shutdowns by intervening only when evidence of deterioration justifies it. Relying on

 doi:10.20944/preprints202606.1304.v1

advanced sensing, IoT and machine learning, PdM analyses signals such as vibration, acoustic and temperature to detect early anomalies and trigger proactive interventions. PdM has been reported to reduce costs in rotating machinery such as AC motors and electric drives by anticipating bearing and gearbox wear [1–3], with multimodal sensor fusion [4] and embedded edge AI [5] identified as central avenues for Industry 5.0.

Traditional PdM systems typically stream raw sensor data to the cloud or high-performance servers for analysis [6]. TinyML, by deploying machine-learning models on resource-constrained microcontrollers, drastically reduces this upload, lowering latency, improving privacy and saving bandwidth [7]. Local edge processing enables decisions in the millisecond range, keeps sensitive vibration, acoustic and image data on-device, and mitigates security risks [6,8]. Within Industry 5.0, two generations of smart sensors can be distinguished: Sensors 1.0, where transducer, processor and ML model coexist in an embedded system, and Sensors 2.0, where the inference engine resides inside the sensor module itself, often on a dedicated application-specific integrated circuit (ASIC), delivering only the inference result rather than raw samples [9,10].

Wireless technologies have evolved in major decadal milestones. The ITU projects a hundred-fold increase in wireless traffic by 2030 [11]. Sixth-generation (6G) networks are expected to deliver terabit-per-second throughputs, sub-millisecond latencies, massive device densities and native AI integration [12]. The overall capability framework for IMT-2030 is consolidated in ITU-R M.2160 [13]; standardization in 3GPP is organized into parallel tracks within Release 20 (5G-Advanced and foundational 6G studies), with Release 21 triggering the normative phase [14]. In smart factories, 6G is expected to interconnect huge numbers of intelligent sensors, while edge-AI accelerators in base stations blur the boundary between communication and computation [15,16].

Within this framework, TinyML is complementary to, not a substitute for, 6G: it provides local decision-making in milliseconds, privacy by design, and degraded-mode operation when the link fails. Federated and split-federated learning find their natural place at the intersection of these two planes. The contributions of this work are threefold: (i) a microcontroller-grade benchmark of five anomaly-detection algorithms on real vibration data, evaluated in terms of F1-score, inference time, and both Flash and stack footprint on the actual Cortex-M4F deployment target; (ii) a multimodal TinyML sensing architecture combining vibration, acoustic and thermograph ic modalities on commercial off-the-shelf hardware, integrated through an IoT gateway and an AWS IoT cloud back-end, complemented by a preliminary intra-session late-fusion analysis providing initial evidence of inter-modality complementarity under controlled static-unbalance conditions; and (iii) an end-to-end URLLC latency experiment quantifying how the cloud-uplink leg dominates the budget, motivating edge-first inference as a design invariant for 6G.

The end-to-end deployment is then mapped onto the IMT-2030 usage scenarios and complemented with a controlled URLLC latency experiment.

The remainder of the article is organized as follows: Section 2 reviews TinyML, PdM and 6G; Section 3 presents the architecture; Section 4 describes materials and methods; Section 5 reports experimental results; Section 6 discusses implications, 6G adaptation, URLLC latency and limitations; Section 7 concludes.

2. Background and Related Work

TinyML denotes the deployment of ML models on tiny, severely resource-constrained devices, typically microcontrollers with CPUs in the tens of MHz, memory in the kilobyte-to-megabyte range and tight energy budgets [17]. Compact architectures (MobileNets, TinyCNN) and compression techniques such as 8-bit (or sub-8-bit) quantization, pruning and distillation enable a variety of AI tasks locally with minimal accuracy loss [7,16]. TensorFlow Lite for Microcontrollers [18] and end-to-end platforms such as Edge Impulse [19] provide the runtime and toolchain. Local processing also yields system-level energy savings: the radio typically dominates the energy budget of IoT nodes, so well-optimized TinyML sensors can run for months on a coin cell while transmitting only sporadically [16].

 doi:10.20944/preprints202606.1304.v1

PdM instruments machinery with accelerometers, microphones, temperature and pressure sensors and thermographic cameras to continuously collect data, inferring the health of bearings, motors and gearboxes from patterns and anomalies in these streams [6,20]. ML models fall into supervised and unsupervised families [6]; data-driven techniques span thresholds, PCA, Isolation Forest, k-NN, Naïve Bayes, SVM and deep networks (CNN, LSTM) [20,21]. Embedding intelligence directly in the sensors via TinyML overcomes the limitations of cloud-centric architectures: smart sensors emit only actionable information (an anomaly score, a class label), reducing network load and enabling sub-second responses such as automatic shutdowns without round-tripping to the cloud [20,22]. Distributed inference on low-power microcontrollers has been shown to achieve accuracies above 99 % in motor condition monitoring [23].

ITU-R M.2160 [13] reorganizes the 5G service space (eMBB, URLLC, mMTC) into six IMT-2030 usage scenarios: three evolved (Immersive Communication, Hyper-Reliable Low-Latency Communication or HRLLC, and Massive Communication) and three new (Ubiquitous Connectivity, Integrated AI and Communication, and Integrated Sensing and Communication or ISAC). Aggregate targets include peak rates near 1 Tb/s, user-plane latencies down to 0.1 ms in HRLLC, reliability above $1–10^{-7}$ and densities up to $10^6–10^7$ devices/km² [13]. Enabling technologies include sub-THz bands, the upper mid-band, ultra-massive MIMO, reconfigurable intelligent surfaces (RIS), edge-oriented architectures and an AI-native air interface, first prototyped in 5G-Advanced via TR 38.843 [24]. For industrial environments, the path to 6G starts not from the public mobile network but from the private and deterministic solutions that 3GPP developed for vertical industries: stand-alone non-public networks (SNPN), which run a fully independent private 5G core; public-network-integrated non-public networks (PNI-NPN), which host a dedicated industrial domain over a shared operator infrastructure; and 5G-LAN, which emulates local Ethernet-style connectivity among plant devices. These are combined with IEEE Time-Sensitive Networking (TSN), which adds bounded latency and clock synchronization for real-time control loops, as specified in TS 23.501 and TR 22.821 [25,26], with cyber-physical service requirements in TS 22.104 and TS 22.261 [27,28]. Integrated Sensing and Communication (ISAC) additionally allows the network itself to act as a distributed sensor, reusing the communication waveform to infer presence, motion and distance of objects in its environment [29,30].

3. System Architecture and Key System Components

The proposed architecture (Figure 1) emphasizes local processing at the sensor nodes, with an IoT network layer (and future 6G connectivity) tying the whole system together. The goal is to perform most filtering and inference at the edge itself, leaving the network for coordination, control, higher-level analytics and model updates.

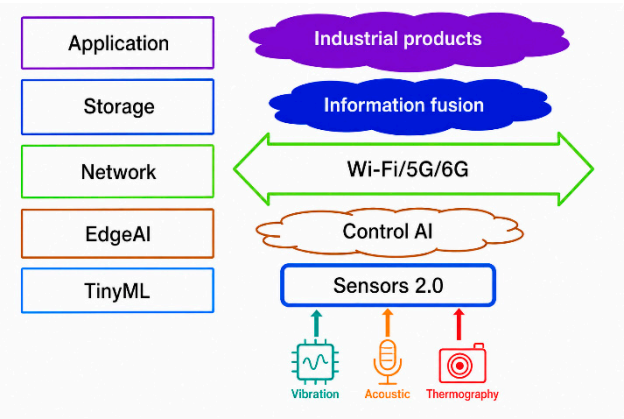


Figure 1. Layered TinyML architecture proposed for predictive maintenance. The bottom TinyML layer hosts smart sensor nodes (Sensors 2.0) running local inference; an edge-AI layer consolidates per-modality results and exposes them via Wi-Fi/MQTT; the network layer covers current Wi-Fi/5G and the path toward 6G; the storage

 doi:10.20944/preprints202606.1304.v1

and application layers host the cloud back-end, the digital twin and the industrial applications fed by information-level fusion.

3.1. Implemented Hardware Architecture

The specific hardware specification used in each modality is summarized in Table 1.

Table 1. Hardware specification used in each sensing modality, embedded processing and IoT gateway.

Function	Component	Model / Reference	Key features
Vibration	Smart node with triaxial MEMS accelerometer	Arduino Nicla Sense ME (Bosch BHI260AP)	Host MCU: nRF52832 (ARM Cortex-M4F at 64 MHz, FPv4-SP); runs TinyML vibration models locally
Acoustic	Acoustic node with neural decision coprocessor	Arduino Nicla Voice (Syntiant NDP120 ASIC)	Acoustic DSP + Syntiant Core 2 neural engine; in-sensor log-Mel classification
Thermography	Embedded thermal camera + IR microbolometer	OpenMV Cam H7+ with FLIR Lepton	Binary CNN inference on radiometric images using TensorFlow Lite
IoT gateway	Embedded gateway with wireless connectivity	Arduino Portenta H7	Dual-core MCU (Cortex-M7/M4); Eslov–Wi-Fi/MQTT bridge to AWS IoT
Inter-device link	I²C-based serial connector	Eslov (Arduino Pro)	Up to 3.4 Mbit/s in I²C High-Speed mode; far above accelerometer throughput
Cloud platform	Serverless IoT, storage and inference services	AWS (IoT Core, Lambda, DynamoDB, S3, SageMaker, TwinMaker, Grafana, SNS)	Storage, alerting, digital twin and centralized training

Each smart node carries a microcontroller capable of running ML inference, in line with the Sensors 2.0 paradigm [9,10]. A predictive-maintenance prototype was developed to monitor a laboratory rotating-machine testbed, targeting three failure modes: vibration anomalies, motor acoustic indicative of incipient damage (e.g., unbalance), and thermal signatures associated with imbalance.

Vibration: a smart node is mounted on the motor casing. The integrated Bosch BHI260AP triaxial MEMS accelerometer exposes preprocessed acceleration vectors via I²C to the host MCU of the Nicla Sense ME, an nRF52832 (ARM Cortex-M4F at 64 MHz, FPv4-SP, 512 KB Flash / 64 KB

 doi:10.20944/preprints202606.1304.v1

SRAM). The TinyML anomaly-detection models reported in Section 5.1 are deployed on this Cortex-M4F host, not on the Fuser2 core internal to the BHI260AP.

Acoustic: an acoustic node is placed next to the motor. A neural decision coprocessor extracts the acoustic features (log-Mel bins) internally, and an in-sensor model classifies them.

Vision: an embedded thermographic camera, coupled to an infrared microbolometer, is placed in front of the motor to capture heat maps (thermograms) of the casing, in which each pixel encodes a surface-temperature value (Figure 2).

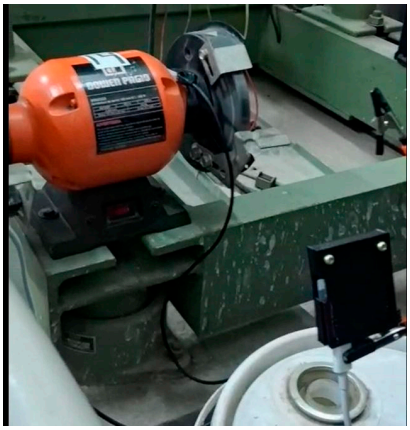


Figure 2. Smart sensor nodes deployed on the laboratory test motor. The vibration node (with the triaxial MEMS accelerometer) is mounted on the motor housing; the acoustic node faces the motor at short distance; the thermographic node is positioned in front of the motor. The test disc (Figure 5) is mounted on the rotating shaft to induce calibrated unbalance.

3.2. Network and Data Integration

The three sensor nodes feed their TinyML models locally and raise alarms when appropriate. They are connected through the Eslov connector, an I²C-based serial interface from the Arduino Pro family (not an industrial fieldbus such as Modbus or PROFIBUS), to an IoT gateway that manages the devices and acts as a Wi-Fi/MQTT bridge to a cloud platform deployed on AWS. The Eslov link supports up to 3.4 Mbit/s in I²C High-Speed mode. The BHI260AP runs the BSX virtual-sensor fusion pipeline internally and exposes preprocessed three-axis acceleration vectors to its host MCU at a configurable cadence; in this work, the application-layer cadence is set to 10 Hz of feature vectors, corresponding to approximately 480 bit/s of raw payload, many orders of magnitude below the I²C capacity (Figure 3).

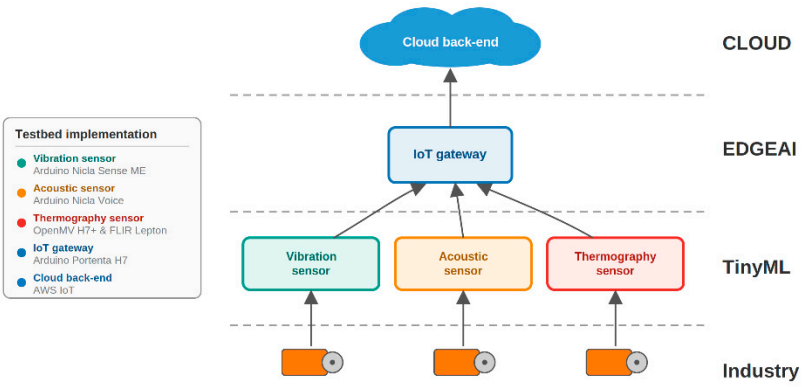


Figure 3. End-to-end network integration. The three smart sensor nodes (Nicla Sense ME for vibration, Nicla Voice for acoustic, OpenMV Cam H7+ with FLIR Lepton for thermography) execute TinyML inference locally;

 doi:10.20944/preprints202606.1304.v1

the Arduino Portenta H7 gateway aggregates their results and forwards them via Wi-Fi/MQTT to AWS IoT Core, where they are stored, visualized and dispatched as alerts.

3.3. Cloud Infrastructure

Amazon Web Services (AWS) is a widely adopted public-cloud provider that offers, on demand, a broad collection of managed services for storage, data processing, machine learning and IoT connectivity. The cloud platform of this work was developed on AWS under the serverless paradigm, in which the user specifies only which services are used and how they interact, while AWS manages the underlying infrastructure [8,31]. The platform persists data from the vibration and acoustic nodes together with the inference results from the thermographic images; trains, validates and deploys ML models; and serves a Grafana-based dashboard for real-time visualization. Figure 4 shows the platform architecture, detailing the main services and their interactions: IoT Core for MQTT and certificate-based device management, Lambda for event-driven processing, DynamoDB for structured records, S3 for the data lake [32], SageMaker for centralized training [33], TwinMaker for the digital twin, Glue for ETL and SNS for alerting [34].

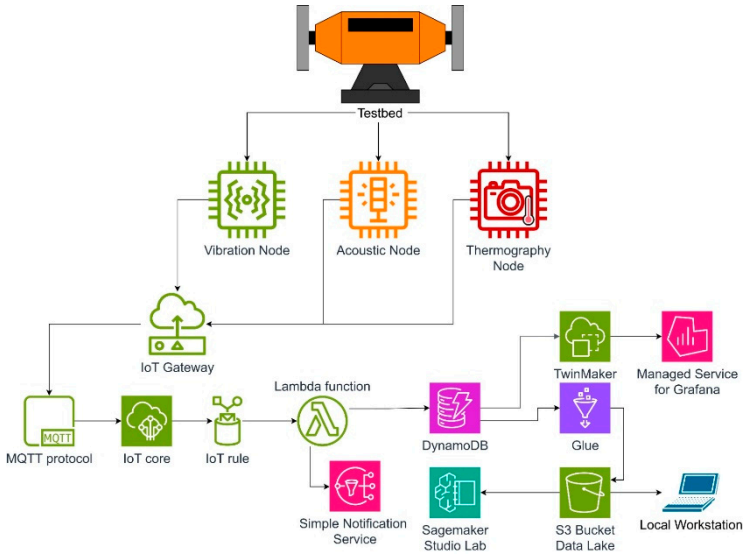


Figure 4. Cloud architecture deployed on AWS. Inbound MQTT messages from the IoT gateway are routed by IoT Core into a Lambda function that decodes, persists in DynamoDB and dispatches the payload. Detected anomalies trigger SNS notifications. A Data Lake on S3, fed by Glue, supports off-line training in SageMaker; TwinMaker and the Grafana-based dashboard visualize the digital twin.

4. Materials and Methods

4.1. Rotating Machine Testbed and Unbalance Protocol

In this section we describe the experimental testbed and the controlled unbalance protocol used to generate the labeled vibration dataset that underpins the anomaly-detection experiments of Section 5. A central difficulty in predictive-maintenance research is obtaining fault data: healthy operation is abundant, but faults are rare and costly to reproduce on production machinery. To address this, we designed and implemented a realistic testbed, built around a rotating machine, that incorporates a mechanism to artificially and repeatably induce abnormal operation, controlled static unbalance, so that both healthy and faulty states can be recorded on demand and at different severities.

Static unbalance occurs when the principal axis of inertia of the rotor is displaced parallel to the axis of rotation, generating a mass eccentricity that produces a non-zero net centrifugal force. The vibration amplitude is directly proportional to the unbalancing mass and its eccentricity, so adding

 doi:10.20944/preprints202606.1304.v1

mass at a given angular position deliberately increases vibration and generates fault signatures, while removing mass (or adding it at the opposite side) reduces vibration toward the balanced reference.

Testbed: A balancing disc of 200 mm diameter and 10 mm thickness was designed with 36 M12 threaded holes machined every 10° around the perimeter (Figure 5), allowing calibrated mass to be placed at any angular position. The disc was mounted on a Dowen Pagio AB150P bench grinder (250 W, 2950 rpm). The experiment proceeds in two stages: (i) unbalance induction, in which an M12 screw is mounted at a known angular position to introduce a calibrated eccentric mass; and (ii) unbalance correction, in which the magnitude and phase angle measured by the vibration instrument determine the compensation mass placed at the opposite angular position. The mass is then iteratively trimmed and the vibration re-measured until the residual amplitude is minimized. This protocol bidirectionally modulates the vibration level, yielding the dataset used in this work (balanced healthy and unbalanced faulty states with different severities).

Figure 5. Balancing disc used in the static unbalance protocol (200 mm diameter, 10 mm thickness, 36 M12 holes every 10°). M12 screws of calibrated mass are inserted at controlled angular positions to induce or compensate unbalance, enabling reproducible generation of vibration signatures across a range of severities.

4.2. Vibration-Based Anomaly Detection

Vibration patterns depend on equipment-specific parameters such as rotational speed and mounting [35–37]. Obtaining labeled fault data on production lines is impractical: a more useful strategy is to learn the vibration pattern under normal conditions and flag deviations indicative of malfunction, i.e., one-class anomaly detection [38,39], in which the training set contains a single class (healthy operation). Autoencoder reconstruction has proven particularly effective on industrial motors without labeled fault data [5,21].

In this work, ten statistical features are extracted from 6 second windows of the accelerometer signal (60 samples per window at the 10 Hz host cadence of Section 3.2). This feature set avoids expensive preprocessing such as the Fourier transform and, being frequency-independent, reduces the dependency on operating speed. The ten features (Table 2) are standard time-domain descriptors widely used in vibration-based condition monitoring [35–37] and recent TinyML-PdM publications [4,5]. With N = 60 samples per window, the higher-order moments (skewness and excess kurtosis) are estimated with acceptable variance; substantially shorter windows would render these fourth-order descriptors statistically unreliable.

Data are collected from the testbed with the disc in balanced and unbalanced configurations across several severities. Windows of 6 s are extracted contiguously without shuffling between recordings. Healthy-condition measurements are partitioned into 60 % for training, 20 % for validation, and 20 % for testing. The healthy test subset is then combined with unbalanced-condition measurements to form the final test partition used for anomaly-detection evaluation. The anomaly

 doi:10.20944/preprints202606.1304.v1

threshold is set at the 95th percentile of the reconstruction-error or projection-distance distribution computed on the healthy training windows, and is never tuned on the test partition.

Five anomaly-detection algorithms were evaluated: Principal Component Analysis (PCA), a fully connected autoencoder evaluated both in FP32 and INT8 (Q8INT) form via post-training quantization, One-Class SVM and Isolation Forest. Autoencoder hyperparameters were tuned via Random Search over 10 configurations (encoder/decoder depths ∈ {1,2,3}, layer widths 8–64, bottleneck 2–8, Adam, MSE loss, early stopping); the winning architecture is 32 → 8 → 8 → 16 → 10 with ReLU hidden layers and linear output. The OC-SVM uses an RBF kernel; the Isolation Forest uses 5 trees of maximum depth 8. Features are standardized to zero mean and unit variance using training-only statistics.

Table 2. Ten time-domain statistical features extracted from each 6-second vibration window (N = 60 samples at the 10 Hz host cadence). Features 1–5 are amplitude/energy descriptors; features 6–10 are dimensionless shape ratios invariant to amplitude scaling.

Statistical Feature	Associated Equation
Mean	$\mu = \frac{1}{N}\sum x_i$
Standard deviation	$\sigma = \sqrt{\frac{1}{N-1}\sum(x_i - \mu)^2}$
Root mean square	$x_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum x_i^2}$
Peak amplitude	$x_{\mathrm{pk}} = \max_i \lvert x_i \rvert$
Peak-to-peak	$x_{\mathrm{pp}} = \max_i x_i - \min_i x_i$
Skewness	$\gamma_1 = \frac{1}{N\sigma^3}\sum(x_i - \mu)^3$
Excess kurtosis	$\gamma_2 = \frac{1}{N\sigma^4}\sum(x_i - \mu)^4 - 3$
Crest factor	$\mathrm{CF} = \frac{x_{\mathrm{pk}}}{x_{\mathrm{RMS}}}$
Shape factor	$\mathrm{SF} = \frac{x_{\mathrm{RMS}}}{\frac{1}{N}\sum \lvert x_i \rvert}$
Impulse factor	$\mathrm{IF} = \frac{x_{\mathrm{pk}}}{\frac{1}{N}\sum \lvert x_i \rvert}$

4.3. Acoustic Modality and Log-Mel Features

Acoustic is an established source for early fault detection, particularly when vibration sensing is impractical or when normal operation produces apparently anomalous vibration [40–42]. Acoustic processing demands more computation than acceleration-based methods, so the acoustic node integrates a neural decision coprocessor (Syntiant NDP120) that combines a DSP and an in-sensor neural engine. The firmware pipeline is generated through Arduino Cloud and Edge Impulse: healthy and faulty acoustic is captured at 16 kHz mono with the on-board MEMS microphone, uploaded to the Edge Impulse cloud project, processed through the "log-bin" feature extractor specific to the NDP120/200 family [19,43], and the trained classifier is deployed back onto the application-

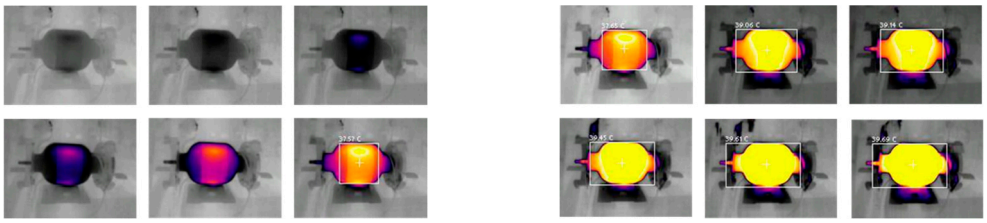
 doi:10.20944/preprints202606.1304.v1

specific integrated circuit (ASIC). The NDP120 is optimized to operate directly on log-Mel filterbank energies (the representation produced just before the DCT in a classical MFCC pipeline), and the "log-bin" block omits that final DCT step. The exact frame length, stride, Mel-filter count, FFT length and pre-emphasis coefficient used here are listed in Section 5.2 (Table 4).

4.4. Thermographic Modality and CNN Classifier

Vibration- and acoustic-emission-based PdM is sensitive to environmental noise: simultaneous operation of nearby machinery, operator activity, ambient conditions and EMI degrade the signal-to-noise ratio and may mask incipient faults [35,44]. Infrared thermography (IRT) is non-contact and non-invasive, making it a useful complement [45]. Traditional industrial thermal-vision systems stream massive radiometric flows to cloud servers, with latency, cybersecurity and bandwidth costs that the TinyML paradigm directly addresses [22,45].

MP4 videos of the laboratory test motor were captured with the embedded thermographic camera, one under nominal (balanced) operation and a second after the disc was unbalanced on the same day with the same setup; a Python script extracted individual frames. Each frame was resized to 96 × 96 px, three RGB channels, normalized to [0,1] and labeled 0 (normal) or 1 (fault). A lightweight CNN, consisting of convolution and max-pooling blocks followed by fully connected dense layers, produces a single fault-probability output. Training used the Adam optimizer with binary cross-entropy and early stopping on a validation subset; the model is then converted to TensorFlow Lite for embedded execution. Sample frames are shown in Figure 6. Because each class is represented by a single continuous video, the train/test split is enforced at the frame level rather than at the video or session level; the implications for the reported metrics are analyzed in Section 6.4.



(**a**) (**b**)

Figure 6. Sample 96 × 96 px thermographic frames captured by the OpenMV Cam H7+ coupled to the FLIR Lepton microbolometer. (**a**) Frames acquired under nominal (balanced) operation, showing a stable thermal pattern. (**b**) Frames acquired under induced static unbalance, showing the increased surface temperature associated with friction at the bearings and the vibrating support.

5. Results

5.1. Vibration Anomaly-Detection Benchmark

The five anomaly-detection algorithms were evaluated by F1-score, accuracy and inference time. Two complementary inference-time measurements are reported: a per-sample latency on a reference laptop (AMD Ryzen 7 260, 3.80 GHz, 32 GB RAM), and a single-inference measurement on the target Cortex-M4F host (nRF52832 of the Nicla Sense ME, not the BHI260AP Fuser2 core); the latter is the median of 16,605 single-shot calls instrumented with micros() in the Arduino runtime. Although the BHI260AP exposes an in-sensor compute fabric, its toolchain is restricted, whereas the nRF52832 runs standard Arduino-mbed firmware and supports the same C/C++ deployment path that would be used in production. Table 3 reports the full benchmark.

Table 3. Performance comparison of the five anomaly-detection algorithm variants on vibration data, with per-sample inference times on the reference laptop and the Cortex-M4F of the Nicla Sense ME, and Flash and stack footprints from the Arduino IDE build report and from -fstack-usage. The two FC autoencoder rows correspond to the same trained network exported in FP32 and INT8 (Q8INT) form.

 doi:10.20944/preprints202606.1304.v1

Model	F1	Accuracy	Laptop / sample (µs)	MCU / sample (µs)	MCU Flash (B)	MCU stack (B)
FC autoencoder Q8INT (TFLite Micro)	0.9976	0.9955	1.28	254	6,056	1,132
FC autoencoder FP32 (TFLite Micro)	0.9974	0.9953	2.40	293	6,640	1,224
One-Class SVM	0.9973	0.9950	32.37	7,057	14,584	40
Isolation Forest	0.9878	0.9781	0.31	32	14,308	28
PCA	0.0066	0.0910	0.18	13	132	64

PCA, despite being the lightest candidate (13 µs and 132 B on the MCU), achieves only F1 = 0.0066 with accuracy 0.091. This is the expected consequence of a linear-projection detector applied to a fault class whose feature-space signature partially overlaps with the normal regime: the moderate-unbalance case has a vibration magnitude actually lower than the balanced regime (see Section 5.4), so its reconstruction error under a PCA model trained only on healthy samples falls inside the 95th-percentile range and is mis-flagged as normal. PCA is therefore retained as a reference baseline but is not a viable deployment candidate. The four non-linear models all reach F1 above 0.98, with the autoencoder variants on top (F1 = 0.9974 FP32, 0.9976 Q8INT), narrowly followed by OC-SVM (0.9973) and Isolation Forest (0.9878). On the Cortex-M4F the ranking is PCA (13 µs) < Isolation Forest (32 µs) < FC autoencoder Q8INT (254 µs) < FC autoencoder FP32 (293 µs) ≪ OC-SVM (7,057 µs). The non-uniform slowdown factors have a clear architectural origin: Isolation Forest is a sequence of integer comparisons through a binary tree, natively supported by the M4F pipeline; the INT8 autoencoder exploits the CMSIS-NN integer kernels, which pack four 8-bit operations per 32-bit MAC where the FP32 variant has to issue one floating-point MAC per coefficient; the OC-SVM evaluates a kernel against 28 support vectors per inference, translating into ~280 dot products of floating-point arithmetic per call without SIMD acceleration. This is precisely the kind of cross-platform reordering that justifies characterizing inference on the deployment target rather than extrapolating from a laptop figure.

Post-training quantization yields a 13 % reduction in inference time, a 9 % reduction in Flash and an 8 % reduction in stack with no measurable F1 degradation, identifying the INT8-quantized autoencoder as the recommended deployment candidate: it lies at the Pareto front of the F1–footprint–latency trade-off, with a per-inference latency approximately 394× below the 100 ms inter-sample period of the I²C accelerometer stream, and generalizes correctly to faults whose feature-space distribution overlaps with normal. All four working candidates fall well within the 64 KB SRAM / 512 KB Flash budget of the nRF52832 and support future co-deployment of additional modalities. The F1 values obtained with the four working algorithms are competitive with those recently reported in [7] for TinyML-based PdM on motors.

5.2. Acoustic Classifier

 doi:10.20944/preprints202606.1304.v1

Acoustic samples of the test machine under healthy and faulty operation were recorded through the integrated Nicla Voice microphone (16 kHz mono, 16-bit) and uploaded to Edge Impulse. The "log-bin" feature extractor configured for the NDP120/200 family computes log-Mel filterbank energies over a frame length of 32 ms and a stride of 24 ms (8 ms overlap), with 40 Mel filters, an FFT length of 512 samples and a pre-emphasis coefficient of 0.96875. The features are fed to a small classifier deployed onto the NDP120 ASIC. In faulty samples, additional frequency bands carrying significant energy appear in the log-Mel spectrogram (Figure 7) that are imperceptible in the healthy baseline.

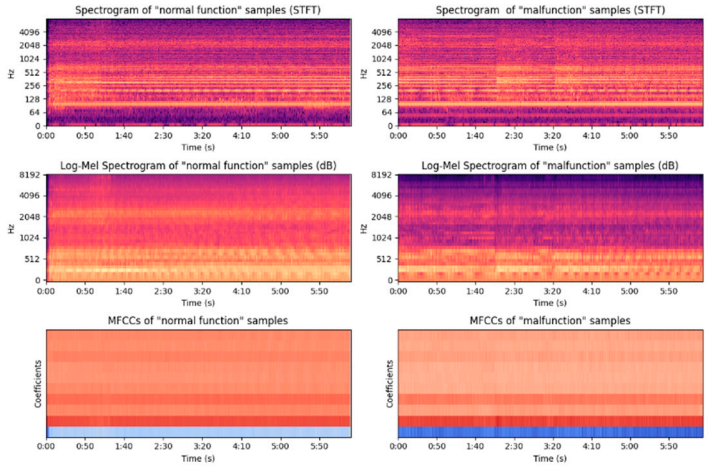


Figure 7. Acoustic feature representations for healthy and faulty motor samples. Top: short-time Fourier transform (STFT) spectrogram. Middle: log-Mel spectrogram with 40 Mel bands, used as the on-chip feature representation for the NDP120 deployment. Bottom: first 10 MFCC coefficients derived from the same log-Mel spectrum, shown as an analytical complement; the NDP120 "log-bin" extractor omits the DCT and consumes the log-Mel bins directly.

Table 4 reports the quantitative performance. The int8-quantized model deployed to the Syntiant Core 2 achieves perfect class separation on the validation set in this experimental configuration, with F1, accuracy, precision, recall and AUC equal to 1.00 and zero false positives or negatives. The dataset comprises 19 min 48 s of acoustic captured during a single laboratory session, with the Edge Impulse train/test split of 65 % / 35 % by duration. The model occupies 1.375 KB of parameter memory (0.2 % of the 640 KB budget) and consumes an estimated 5.55 µJ per inference at 0.9 V. The Core 2 is a streaming in-sensor accelerator: it produces one classification per frame and is latency-bounded by the 24 ms frame stride (~42 inferences/s), not by ASIC throughput. This single-session structure means that training and test partitions share the same ambient acoustic environment, mic placement and motor warm-up state; the F1 = 1.00 demonstrates separability in this recording session but does not establish cross-session generalization, a limitation revisited in Section 6.4. These metrics demonstrate intra-session separability of the recorded acoustic and must not be read as generalization performance: training and test segments are cut from a single continuous recording sharing the same acoustic environment, microphone placement and motor warm-up state. Cross-session validation is required to establish field performance and is left to future work.

Table 4. Configuration of the Edge Impulse "log-bin (NDP120/200)" feature extractor and validation-set performance of the int8-quantized acoustic classifier deployed on the Syntiant Core 2.

Parameter	Value
Frame length / stride	32 ms / 24 ms (8 ms overlap)
Mel filterbank size / FFT length	40 filters / 512 samples

 doi:10.20944/preprints202606.1304.v1

Parameter	Value
Pre-emphasis coefficient	0.96875
Output representation	40 log-Mel bin energies per frame (no DCT)
F1 / Accuracy / Precision / Recall / AUC (int8)	**1.00 / 1.00 / 1.00 / 1.00 / 1.00**
False positives / negatives	0 % / 0 %
Inference cadence on Syntiant Core 2	24 ms per decision (~42 inferences/s)
Model parameter memory (NDP120_B0)	1.375 KB of 640 KB (~0.2 %)
Estimated energy per inference (0.9 V)	5.55 µJ
Total acoustic collected (Train / Test split)	19 min 48 s (65 % / 35 % by duration)

5.3. Thermographic CNN

Figure 8 reports the training and validation curves of the binary thermographic CNN. Accuracy reaches values close to 100 % within the first few epochs on both partitions, and the binary cross-entropy loss falls close to 0 and remains there. As anticipated in Section 4.4 and discussed in Section 6.4, these curves should be interpreted as evidence that the CNN cleanly separates the frames of the two videos available for this study (one continuous video of the balanced motor and one of the unbalanced motor, both captured in the same session), rather than as evidence of cross-session generalization: with the train/test split applied at the frame level over two single videos, the temporal correlation between adjacent frames is the dominant signal the network can exploit. On the embedded camera the trained model is loaded from on-board memory; each captured frame is preprocessed to 96 × 96 px and classified in real time, with sliding statistics of fault rate displayed on-screen and periodic console reports. Moreover, because both videos were recorded in a single session, the temporal correlation between adjacent frames and the progressive motor warm-up are confounded with the fault signature and cannot be ruled out under the present single-session design; the near-100 % accuracy therefore reflects intra-session separability rather than cross-session generalization.

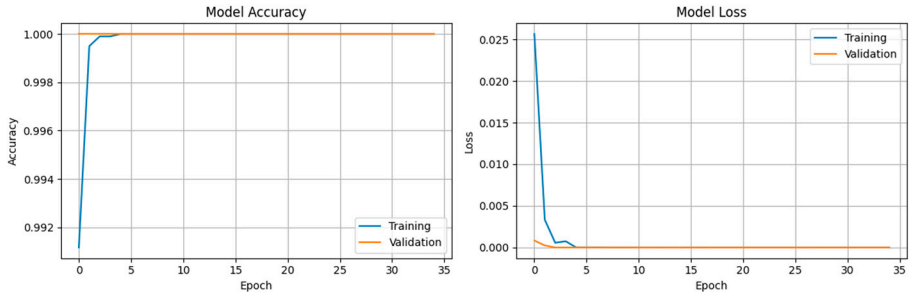


Figure 8. Training and validation curves of the binary thermographic CNN classifier (healthy vs. faulty motor) over 35 epochs. Left: classification accuracy. Right: binary cross-entropy loss. As discussed in Section 6.4, these curves reflect intra-session separability of frames sampled from two single-session videos with a frame-level train/test split, not cross-session generalization.

5.4. Multimodal Late Fusion

 doi:10.20944/preprints202606.1304.v1

Sections 5.1–5.3 evaluate each modality in isolation. To address whether the three modalities provide complementary or merely redundant evidence, a late-fusion experiment was carried out on three recorded sessions of the same motor: one balanced (N) and two unbalanced (moderate A and severe MA). The three modalities were captured under identical operating conditions but with each sensor operating on its own independent acquisition pipeline; strict event-level synchronization is therefore not available, and the experimental design accommodates this with a class-stratified bootstrap protocol.

Four late-fusion strategies are evaluated: logical OR (any modality flags fault), logical AND (all three modalities agree), majority vote (≥2 of 3), and a logistic-regression meta-learner trained on the three modality confidence scores via 5-fold stratified cross-validation. Because the three modalities are not strictly event-aligned in time, a class-stratified bootstrap was applied: at each of 1,000 iterations, 150 synthetic test events were assembled by drawing 50 normal-operation events and 100 faulty-operation events covering different fault severities, ranging from mild to severe malfunction conditions. The bootstrap preserves the class labels of each source while controlling the relative contribution of the different fault severities within the faulty class. Table 5 reports the resulting metrics as bootstrap means with 95 % confidence intervals. Note that, for the acoustic and thermographic modalities, the fusion is fed by offline scores re-derived from features (logistic regression on log-Mel and texture features, respectively) rather than by the in-sensor decisions of the NDP120 ASIC and the embedded CNN; the fusion is therefore an offline analysis and not the in-sensor decision pipeline.

Table 5. Late-fusion performance of the three modalities under a 1,000-iteration class-stratified bootstrap (150 events per iteration: 50 normal, 100 fault). Means are reported; the F1 column includes 95 % confidence intervals from the bootstrap distribution; the confusion column reports mean TN / FP / FN / TP per iteration. The vibration baseline is the INT8 FC autoencoder of Section 5.1.

Model	Accuracy	Precision	Recall	F1 [95 % CI]	TN/FP/FN/TP
FC autoencoder Q8INT (vibration)	0.789	0.981	0.697	0.814 [0.756, 0.870]	49 / 1 / 30 / 70
Acoustic only (log-Mel + LR)	0.947	0.927	1.000	0.962 [0.939, 0.985]	42 / 8 / 0 / 100
Thermographic only (textures + LR)	0.762	0.738	1.000	0.849 [0.830, 0.873]	14 / 36 / 0 / 100
Fusion — Logical OR	0.743	0.722	1.000	0.838 [0.820, 0.859]	11 / 39 / 0 / 100
Fusion — Logical AND	0.797	0.998	0.697	0.820 [0.758, 0.876]	50 / 0 / 30 / 70
Fusion — Majority vote (≥2 of 3)	0.956	0.939	1.000	0.968 [0.943, 0.990]	43 / 7 / 0 / 100
Fusion — Logistic regression (5-fold CV)	0.988	0.982	1.000	0.991 [0.980, 1.000]	48 / 2 / 0 / 100

 doi:10.20944/preprints202606.1304.v1

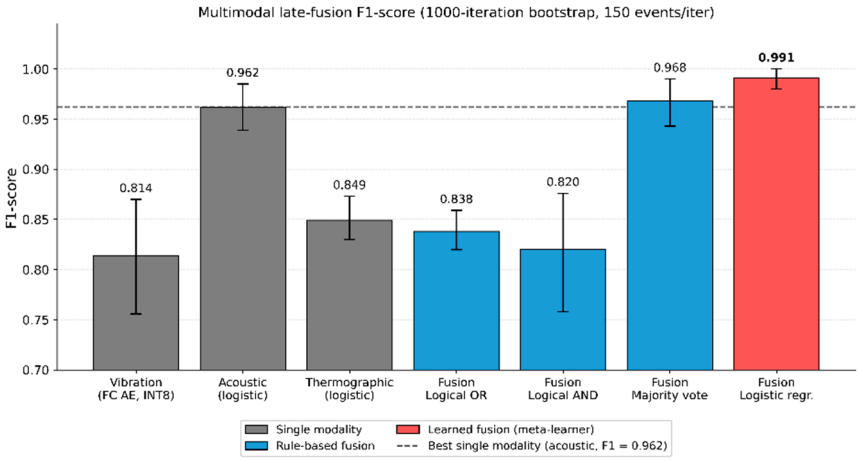


Figure 9. Multimodal late-fusion F1-score for the three single-modality baselines (vibration, acoustic, thermographic) and four late-fusion strategies (logical OR, logical AND, majority vote, and logistic regression on confidence scores trained via 5-fold stratified cross-validation). Bars show the mean F1 over 1,000 bootstrap iterations of 150 events per iteration (50 normal, 100 fault); error bars show 95 % confidence intervals. The dashed horizontal line marks the best single-modality baseline (acoustic, F1 = 0.962). The logistic-regression meta-learner (rightmost bar) is the only fusion strategy whose 95 % CI lies entirely above the best single-modality baseline.

Several observations follow. First, the vibration-only Q8INT autoencoder reaches F1 = 0.814 [0.756, 0.870] with high precision (0.98) but low recall (0.70); a per-source breakdown shows that the limitation is concentrated on mild-to-moderate fault conditions, of which only 47 % is correctly detected, while 92 % of severe fault events and 97 % of normal-operation events are correctly classified, the expected behavior of a one-class autoencoder on a fault class whose feature-space statistics partially overlap with the normal regime. Second, the acoustic baseline reaches F1 = 0.962 [0.939, 0.985] with perfect recall but 7–8 false positives per 50 normal events, reflecting the higher sensitivity of log-Mel features to background-noise variability. Third, the thermographic baseline reaches F1 = 0.849 [0.830, 0.873]; its lower precision reflects the difficulty of separating thermal patterns of the unbalanced motor from normal warm-up patterns when only frame-level texture features are available. Fourth, the rule-based fusion strategies show the expected pattern: OR amplifies the false-positive rate of the most permissive modality (F1 = 0.838); AND inherits the missed-detection rate of vibration on mild-to-moderate fault conditions (F1 = 0.820 with recall 0.70); majority vote averages out the precisions and lifts F1 to 0.968 [0.943, 0.990], close to the best single modality. Fifth, and central to the contribution of this work, the logistic-regression meta-learner reaches F1 = 0.991 [0.980, 1.000] with accuracy 0.988, precision 0.982 and a perfect recall, strictly improving on every single-modality baseline and on every rule-based fusion strategy in the mean and across the 95 % confidence interval. The confusion matrix averages 48.2 TN and 1.8 FP out of 50 normal-operation events, and 100 TP with no FN out of 100 faulty-operation events. Taken together, these results provide preliminary evidence of inter-modality complementarity under intra-session conditions; because the three modalities were not synchronized at the event level, the fusion operates over a class-stratified bootstrap of independently drawn modality scores rather than over genuinely co-occurring observations, so the complementarity holds in a marginal statistical sense. Confirmation under event-synchronized, cross-session data is left to future work. The key advantage of the multimodal design is twofold: the learned fusion surpasses every individual modality and every rule-based strategy in both mean F1 and its 95 % confidence interval, and, most relevant for deployment, it recovers the ~30 % of moderate-unbalance faults that no single sensor reliably detects, while keeping false alarms below two per fifty normal events (precision 0.982, perfect recall).

6.1. Architectural Implications

 doi:10.20944/preprints202606.1304.v1

The combination of vibration, acoustic and thermographic sensing produced complementary evidence of failure: vibration captures the mechanical signature, acoustic captures incipient acoustic patterns that may emerge before vibration-detectable damage, and thermography captures the steady-state thermal footprint, which is robust to ambient mechanical noise. The late-fusion experiment of Section 5.4 provides preliminary evidence of this complementarity (mean F1 = 0.991 above every single-modality baseline), specific to the induced static-unbalance scenario evaluated here, in which all three modalities ob-serve the same fault mode, and consistent in direction with results reported in [4] for vibration–acoustic data fusion in electric motors. Pushing inference to the edge implies that only high-level results are transmitted (anomaly flags, class labels, lightweight statistics), preserving bandwidth and privacy, which is particularly relevant in industrial environments with limited infrastructure or high data sensitivity.

6.2. Adaptation to 6G

Although the prototype uses Wi-Fi and a local MQTT broker, the architecture is designed to evolve toward a 6G substrate incrementally, without rewriting the application plane. Table 6 maps the six IMT-2030 usage scenarios [13] onto the architecture and the 3GPP Technical Reports that govern each enabler.

Beyond the scenario-by-scenario mapping, three transversal axes couple the architecture to 6G. First, cellular connectivity: nodes can migrate from Wi-Fi/MQTT to cellular UE in two steps, RedCap/eRedCap as a practical bridge over 5G-Advanced networks, and an A-IoT profile on those non-critical assets where the energy budget prohibits a conventional active radio. Second, private networking: for automation-intensive factory deployments, the combination of SNPN/PNI-NPN, 5G-LAN and IEEE TSN integration provides the deterministic guarantees that control links require. Third, security: the current MQTT broker and the x.509 certificates of IoT Core must evolve toward post-quantum profiles based on standardized key-encapsulation mechanisms (ML-KEM), following the trajectory that 3GPP SA3 is articulating from TR 33.841 [50–52]; this neutralizes the "harvest-now, decrypt-later" threat. A defining 6G contribution with respect to the current prototype is ISAC: the network itself contributes a fourth sensing modality, co-generated with data transmission, that adds evidence complementary to vibration, acoustic and thermography (reflection patterns on the casing of a rotating machine, presence of operators in a restricted area during an alarm, geometric degradation via micro-Doppler) [29,30]. Fusion with the three prototype modalities is a natural line of work once the 3GPP sensing-data exposure APIs are available. "6G readiness" is, however, not a binary property: the 6G air interface is being designed to be non-backward-compatible with NR [48,49], so migration will involve replacing the radio modems of the nodes rather than a firmware upgrade. What is preserved is the application plane, the TinyML logic, the message format, the MQTT gateway, the serverless back-end and the digital twin, with incremental investment concentrated at the radio edge.

Table 6. Mapping between the six IMT-2030 usage scenarios and the components of the proposed architecture.

IMT-2030 scenario	How the architecture addresses it	3GPP / IETF enablers	Refs.
HRLLC	PdM alarms and emergency-stop with latency <1 ms and reliability ≥1−10⁻⁶ between node, gateway and plant control plane.	URLLC → HRLLC slice; TS 23.501 (slicing); TS 22.104 (cyber-physical control).	[13,25,27]
Massive Communication	Hundreds to thousands of sensor nodes per plant without signaling	RedCap/eRedCap; Ambient IoT (TR	[46,47]

 doi:10.20944/preprints202606.1304.v1

IMT-2030 scenario	How the architecture addresses it	3GPP / IETF enablers	Refs.
	collapse; A-IoT on non-critical assets.	22.840, TR 38.848); NB-IoT/LTE-M evolution.	
Integrated AI & Communication	Inference distribution: node (TinyML), gateway (intermediate models), cloud/edge (heavy models, FL aggregation); federated and split FL.	AI-native air interface (TR 38.843); AI/ML model transfer; SA2/SA5 lifecycle frameworks.	[24,48,49]
ISAC	Network as a fourth sensing modality: operator presence detection, micro-Doppler of rotating parts and geometric changes without additional sensors.	TR 22.837 (ISAC requirements); TS 22.137 (normative); TR 38.901 (channel modeling).	[29,30]
Ubiquitous Connectivity	Remote PdM sites (wind farms, pipelines, high-altitude mining, solar) via terrestrial–satellite integration without changes in the application plane.	NTN integration (Rel-17/18/19 → 6G).	[14]
Immersive Communication	Digital-twin visualization and XR support for assisted maintenance (future work).	Mobile AI awareness, PDU set for mobile AI (Rel-20 5G-A).	[14]

6.3. End-to-End URLLC Latency Measurement

To complement the architectural discussion with a quantitative measurement, a controlled latency-injection experiment was carried out on the deployed prototype. A Python forwarder receives every event published to the local Mosquitto broker, applies a per-event Gaussian delay sampled from $N(\mu, \sigma^2)$ with μ = mean delay and σ = jitter, and republishes the message to AWS IoT Core over TLS-MQTT. An AWS IoT Core rule enriches each message with the server-side ingestion timestamp and forwards it to an AWS Lambda function, which timestamps the arrival and persists each event to a dedicated DynamoDB table. Five injection profiles are evaluated, each over 50 events at 10 Hz with an approximately 258-byte JSON payload: baseline (no injection), legacy LTE eMBB (μ = 50 ms, σ = 10 ms), 5G eMBB (μ = 20 ms, σ = 4 ms), 5G URLLC (μ = 1 ms, σ = 0.2 ms), and 6G URLLC (μ = 0.1 ms, σ = 0.05 ms). The URLLC profiles correspond to user-plane latency targets specified by 3GPP TS 22.261 [28] and Report ITU-R M.2410 [53]; the 6G targets are based on recent literature projections [14,48,49]. The methodology corresponds to a software shim that isolates the effect of one-way transport latency on the end-to-end pipeline while keeping the application plane unchanged across runs. The deployment site is Tucumán, Argentina, and the AWS region is us-east-2 (Ohio), separated by approximately 9,000 km. Table 7 reports the measured end-to-end latency.

Table 7. End-to-end latency from forwarder publish to AWS Lambda arrival under five injection profiles (50 events per profile, N = 250). All 250 events were matched at 100 % between local forwarder log and DynamoDB ingestion. Latencies are in milliseconds.

 doi:10.20944/preprints202606.1304.v1

Injection profile	n	mean	std	p50	p95	p99
Baseline (no injection)	50	562.63	59.35	561.24	649.59	662.78
Legacy LTE eMBB ($\mu = 50$, $\sigma = 10$)	50	637.97	63.61	638.61	732.41	744.23
5G eMBB ($\mu = 20$, $\sigma = 4$)	50	591.23	65.42	596.07	686.00	720.84
5G URLLC ($\mu = 1$, $\sigma = 0.2$)	50	583.09	69.46	580.75	669.78	762.13
6G URLLC ($\mu = 0.1$, $\sigma = 0.05$)	50	584.67	61.45	584.07	673.28	695.26

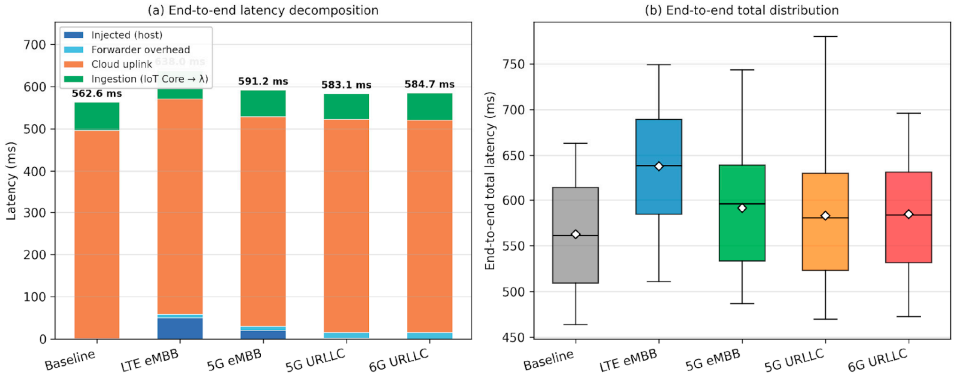


Figure 10. End-to-end latency decomposition of the forwarder → AWS IoT Core → Lambda pipeline under five injection profiles (N = 50 events per profile, 10 Hz, ≈258-byte payload, route Tucumán → AWS us-east-2). (a) Stacked-bar decomposition per leg: synthetic injection on the host, forwarder overhead, cloud uplink (forwarder → IoT Core via TLS) and ingestion (IoT Core rule → Lambda dispatch). (b) Full distribution of total end-to-end latency. The cloud-uplink leg (~500 ms) dominates the budget across all profiles, neutralizing the differences between LTE, eMBB and URLLC radio-access regimes.

Three observations follow. First, the injection mechanism is faithful for eMBB-class profiles (<1 % error on the mean, <6 % on the standard deviation). Second, the host scheduler saturates the sub-millisecond regime: for the 5G and 6G URLLC profiles, the forwarder overhead (~15 ms) saturates regardless of the configured target, attributable to the Windows scheduler timer resolution (~15.625 ms per tick). Third, and most relevant, the cloud-uplink leg dominates the latency budget: the forwarder → AWS IoT Core segment contributes consistently between 495 and 513 ms across all profiles, between 79 % and 88 % of the total. This is the combined effect of the ~9,000 km geodesic distance, mutually authenticated TLS session establishment, and propagation through intermediate networks not optimized for industrial traffic. Total end-to-end latencies converge to the 563–638 ms interval, and the difference between the means of 6G URLLC (584.67 ms) and baseline (562.63 ms) is statistically indistinguishable from the cloud-uplink noise. A hypothetical three-order-of-magnitude improvement in radio-access latency (from 50 ms to 0.1 ms) translates to an end-to-end improvement below 9 % when processing occurs in a distant cloud region. This has a direct architectural implication: even with a 0.1 ms URLLC access link, the only viable route to meet sub-100 ms end-to-end budgets at industrial scale is to run inference locally on the device. The TinyML paradigm adopted here is therefore not an optional optimization but a necessary condition for the URLLC capabilities promised by 6G to manifest as observable application-level improvements. The injection mechanism is faithful only for eMBB-class profiles; for the 5G and 6G URLLC profiles the host timer resolution (~15.6 ms tick) saturates the configured sub-millisecond targets, so these profiles are not independently resolved. The URLLC conclusion therefore follows from the magnitude of the cloud-uplink leg (~500 ms) rather than from sub-millisecond injection fidelity, and holds a fortiori for any

 doi:10.20944/preprints202606.1304.v1

radio-access regime. The us-east-2 region was used for deployment; because the conclusion is dominated by the cloud-uplink leg, it would also hold for a geographically nearer region such as sa-east-1.

6.4. Limitations

The current evaluation is restricted to a single motor under controlled conditions with a relatively limited dataset; generalization to other rotating machines will require additional data or transfer-learning strategies. Two modalities reach near-perfect scores on the present validation sets: the thermographic CNN reaches values close to 100 % on accuracy and the acoustic classifier achieves F1, precision, recall and AUC all equal to 1.00. These results reflect the strong intra-modality separability of the present dataset more than fully established generalization. A more fundamental concern affects both modalities: the acoustic dataset was captured in a single session (the motor was first run balanced, then the disc was unbalanced and the recording continued; the Edge Impulse 65 % / 35 % split was obtained by cutting each of these long recordings into a training and a test segment) and the thermographic dataset has an analogous structure (one video balanced, one video unbalanced, both on the same day, frame-level train/test split). In both modalities, training and test partitions therefore share the same recording session, ambient environment, sensor placement and motor warm-up state, with adjacent frames highly correlated. The F1 = 1.00 and the near-100 % CNN accuracy demonstrates separability of the present recordings but do not establish session-to-session generalization. A cross-session validation protocol with independently captured sessions will be incorporated in future work. The vibration benchmark of Section 5.1 is less exposed because the ten statistical features are intrinsically frequency-independent, but it is also derived from a single test bench and the cross-machine caveat applies. An additional physical constraint affected the severe-unbalance case MA: the thermographic recording could only be sustained for 43 s because operating the bench at MA for longer intervals risked structural damage. This is not a methodological choice but a physical limitation that strengthens the motivation for the proposed PdM architecture: the cases in which intervention is most urgent are precisely those for which extended data collection is unsafe, favoring detectors calibrated on the more abundant normal and moderate-unbalance regimes.

7. Conclusions and Future Work

This article presented and experimentally evaluated a novel multi-sensor edge architecture for industrial predictive maintenance based on TinyML, with an explicit forward-looking discussion of its evolution path toward 6G. Three sensing modalities (vibration, acoustic and thermography) were deployed on commercial off-the-shelf hardware connected through an I²C-based serial link to an Arduino Portenta H7 gateway and a serverless AWS IoT back-end. On the vibration modality, five anomaly-detection algorithm variants were trained and compared on real signals from a controlled unbalance test bench; four of the five reached an F1-score above 0.98, PCA failed at F1 = 0.007 because a linear-projection one-class detector cannot separate a moderate-unbalance regime that overlaps with normal, and the INT8-quantized FC autoencoder (Q8INT) emerged as the recommended deployment default with F1 = 0.9976, 254 µs of inference latency and 6,056 B of Flash on the target Cortex-M4F. The acoustic modality was addressed with a classifier on log-Mel filterbank energies (the native feature representation of the Syntiant NDP120 in-sensor neural decision coprocessor) and the thermographic modality with a lightweight binary CNN. A preliminary intra-session late-fusion experiment provided initial evidence of inter-modality complementarity: a logistic-regression meta-learner on the three confidence scores reached mean F1 = 0.991 [0.980, 1.000] on a class-stratified bootstrap, above every single-modality baseline and every rule-based fusion strategy, with cross-session confirmation left to future work. A controlled URLLC latency-injection experiment showed that the cloud-uplink leg dominates the end-to-end budget by 79–88 %, demonstrating that edge inference is a necessary condition for the URLLC capabilities promised by 6G to manifest at the application level. The findings should be interpreted within the limitations of Section 6.4 (single motor, intra-session leakage in the acoustic and thermographic datasets) and several research

 doi:10.20944/preprints202606.1304.v1

directions emerge: split federated learning across nodes, gateway and cloud, leveraging communication-efficient variants such as PipeSFL and CSE-FSL for heterogeneous clients [54,55]; neuromorphic acceleration of sparse, event-driven inference on coprocessors such as BrainChip AKD1000 or Innatera Pulsar integrated with a Jetson-class host; post-quantum cryptography on the gateway and selected nodes (e.g., ML-KEM-768) to secure 6G-era industrial IoT links against "harvest-now, decrypt-later" attacks; and selective state-space models (e.g., Mamba) for long vibration and acoustic sequences.

Author Contributions: Conceptualization, C.E.G., G.N.M. and M.K.; methodology, C.E.G., F.A.M.B. and M.F.; software, C.E.G. and G.N.M.; validation, S.G.G., P.A.R. and F.I.P.; investigation, C.E.G., G.N.M., S.G.G., P.A.R. and F.I.P.; data curation, G.N.M.; writing—original draft preparation, C.E.G. and G.N.M.; writing—review and editing, all authors; supervision, F.A.M.B. and M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CIASUR and by the Laboratorio Área IV – Termología, Departamento de Mecánica, UTN-FRT.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets generated and analyzed during the study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, T.; Ran, Y.; Zhou, X.; Wen, Y.; Deng, R. A survey of predictive maintenance: Systems, purposes and approaches. arXiv 2019, arXiv:1912.07383.
2. Ismail, E.A.I.; Ahmad, M.R. Anomaly detection in the temperature of an AC motor using embedded machine learning. Jurnal Teknologi 2023, 85, 67–73.
3. Aung, K.H.H.; Kok, C.L.; Koh, Y.Y.; Teo, T.H. An embedded machine learning fault detection system for electric fan drive. Electronics 2024, 13, 493.
4. Suawa, P.; Meisel, T.; Jongmanns, M.; Huebner, M.; Reichenbach, M. Modeling and fault detection of brushless direct current motor by deep learning sensor data fusion. Sensors 2022, 22, 3516.
5. Givnan, S.; Chalmers, C.; Fergus, P.; Ortega-Martorell, S.; Whalley, T. Anomaly detection using autoencoder reconstruction upon industrial motors. Sensors 2022, 22, 3166.
6. Njor, E.; Hasanpour, M.A.; Madsen, J.; Fafoutis, X. A holistic review of the TinyML stack for predictive maintenance. IEEE Access 2024, 12, 184861–184882.
7. Reis, M.J.C.S. Lightweight signal processing and edge AI for real-time anomaly detection in IoT sensor networks. Sensors 2025, 25, 6629.
8. Hamdan, S.; Ayyash, M.; Almajali, S. Edge-computing architectures for Internet of Things applications: A survey. Sensors 2020, 20, 6441.
9. Warden, P.; Stewart, M.; Plancher, B.; Katti, S.; Reddi, V.J. Machine learning sensors. Commun. ACM 2023, 66, 25–28.
10. Guo, S.; Zhou, Q. Machine Learning on Commodity Tiny Devices: Theory and Practice; CRC Press: Boca Raton, FL, USA, 2022.
11. Shayea, I.; Hadri Azmi, M.; Abd Rahman, T.; et al. Spectrum gap analysis with practical solutions for future mobile data traffic growth in Malaysia. IEEE Access 2019, 7, 24910–24933.
12. Saad, W.; Bennis, M.; Chen, M. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. IEEE Netw. 2020, 34, 134–142.
13. ITU-R. Recommendation ITU-R M.2160-0: Framework and Overall Objectives of the Future Development of IMT for 2030 and Beyond; ITU: Geneva, Switzerland, 2023.

 doi:10.20944/preprints202606.1304.v1

14. 3GPP. Release 20 Description; 5G-Advanced and 6G Studies. https://www.3gpp.org/specifications-technologies/releases/release-20 (accessed on 1 May 2026).
15. Situnayake, D.; Plunkett, J. AI at the Edge: Solving Real-World Problems with Embedded Machine Learning; O'Reilly Media: Sebastopol, CA, USA, 2023.
16. Warden, P.; Situnayake, D. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers; O'Reilly Media: Sebastopol, CA, USA, 2019.
17. Kallimani, R.; Pai, K.; Raghuwanshi, P.; Iyer, S.; López, O.L.A. TinyML: Tools, applications, challenges, and future research directions. Multimed. Tools Appl. 2024, 83, 29015–29045.
18. TensorFlow Lite for Microcontrollers. https://www.tensorflow.org/lite/microcontrollers (accessed on 1 April 2026).
19. Edge Impulse. https://edgeimpulse.com/ (accessed on 1 April 2026).
20. Arciniegas, S.; Rivero, D.; Piñan, J.; Diaz, E.; Rivas, F. IoT device for detecting abnormal vibrations in motors using TinyML. Discov. Internet Things 2025, 5.
21. Tanuska, P.; Spendla, L.; Kebisek, M.; Duris, R.; Stremy, M. Smart anomaly detection and prediction for assembly process maintenance in compliance with Industry 4.0. Sensors 2021, 21, 2376.
22. Wang, J.; Wang, Y.; Li, Y. Edge-to-cloud IIoT for condition monitoring in manufacturing systems with ubiquitous smart sensors. Sensors 2022, 22, 5901.
23. Kolok, P.; Hodoň, M.; Ševčík, P.; et al. Low-cost IoT-based predictive maintenance using vibration and acoustic signals on MEMS sensors. Sensors 2025, 25, 6610.
24. 3GPP TR 38.843. Study on AI/ML for NR Air Interface (Release 18); 3GPP, 2024.
25. 3GPP TS 23.501. 5G System Architecture; 3GPP, latest release.
26. 3GPP TR 22.821. 5G LAN Support; 3GPP, latest release.
27. 3GPP TS 22.104. Service Requirements for Cyber-Physical Control in Vertical Domains; 3GPP, latest release.
28. 3GPP TS 22.261. Service Requirements for the 5G System; 3GPP, latest release.
29. 3GPP TR 22.837. Study on Integrated Sensing and Communication (Release 19); 3GPP, 2024.
30. 3GPP TS 22.137. Service Requirements for Integrated Sensing and Communication; 3GPP, 2024.
31. Amazon Web Services IoT Core. https://aws.amazon.com/iot-core/ (accessed on 1 April 2026).
32. Building Data Lakes on AWS. https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/amazon-s3-data-lake-storage-platform.html (accessed on 1 April 2026).
33. Real-Time Inference with Amazon SageMaker. https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-single-model.html (accessed on 1 April 2026).
34. AWS Lambda Quotas. https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html (accessed on 1 April 2026).
35. Mohanty, A.R. Machinery Condition Monitoring: Principles and Practices; CRC Press: Boca Raton, FL, USA, 2014.
36. Randall, R.B. Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications; Wiley: Chichester, UK, 2011.
37. Taylor, J.I. The Vibration Analysis Handbook; Vibration Consultants: Tampa, FL, USA, 1994.
38. Perera, P.; Oza, P.; Patel, V.M. One-class classification: A survey. arXiv 2021, arXiv:2101.03064.
39. Mehrotra, K.G.; Mohan, C.K.; Huang, H. Anomaly Detection: Principles and Algorithms; Springer: Cham, Switzerland, 2017.
40. Chang, S.H.; Purnomo, A.T.; et al. Machine fault detection through acoustic analysis using MFCC and machine learning. Jurnal POLIMESIN 2025, 23, 270.
41. Akbalık, F.; Yıldız, A.; Ertuğrul, Ö.F.; Zan, H. Engine fault detection by acoustic analysis and machine learning. Appl. Sci. 2024, 14, 6532.
42. Henriquez, P.; Alonso, J.B.; Ferrer, M.A.; Travieso, C.M. Review of automatic fault diagnosis systems using audio and vibration signals. IEEE Trans. Syst. Man Cybern. Syst. 2014, 44, 642–652.
43. Arduino Nicla Voice User Manual. https://docs.arduino.cc/tutorials/nicla-voice/user-manual/ (accessed on 1 April 2026).
44. Antoni, J.; Randall, R. Unsupervised noise cancellation for vibration signals. Mech. Syst. Signal Process. 2004, 18, 89–101.

 doi:10.20944/preprints202606.1304.v1

45. Alvarado-Hernandez, A.I.; Zamudio-Ramirez, I.; Jaen-Cuellar, A.Y.; et al. Infrared thermography smart sensor for the condition monitoring of gearbox and bearings faults in induction motors. Sensors 2022, 22, 6075.
46. 3GPP TR 22.840. Study on Ambient IoT Use Cases; 3GPP, 2023.
47. 3GPP TR 38.848. Study on Ambient IoT in RAN; 3GPP, 2023.
48. 3GPP TR 38.914. Study on 6G Scenarios and Requirements; 3GPP, work in progress (2026).
49. 3GPP TR 22.870. Study on 6G Use Cases and Service Requirements (Stage 1); 3GPP, 2025.
50. 3GPP TR 33.841. Study on the Support of 256-bit Algorithms for 5G; 3GPP, latest release.
51. 3GPP TS 33.501. 5G Security Architecture and Procedures; 3GPP, latest release.
52. NIST. FIPS 203 / FIPS 204. Module-Lattice-Based Key-Encapsulation and Signature Standards (ML-KEM, ML-DSA); NIST, 2024.
53. Report ITU-R M.2410-0. Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s); ITU-R: Geneva, Switzerland, 2017.
54. Bolat, Y.; Murray, I.; Ren, Y.; Ferdosian, N. Decentralized distributed sequential neural networks inference on low-power microcontrollers: a predictive maintenance case study. Sensors 2025, 25, 4595.
55. Liu, X.; Dong, X.; Jia, N.; Zhao, W. Federated learning-oriented edge computing framework for the IIoT. Sensors 2024, 24, 4182.