



Developing a Lightweight Edge-AIoT Model for Real-Time Prediction of Mechanical Failures to Improve Maintenance

Prepared By

Name: Abdulaziz Saleh Mohammed Alhasil

ID No.: 261002308

Name: Majed Humud Alhumuda

ID No.: 253002679

*Submitted in fulfillment of the requirements for the degree Master of Science in
Artificial Intelligence*

At

Collage of Informatics

Midocean University

Supervisor Name: Dr Sherif Mostafa

Collage of Informatics, Midocean University, Comoros

Email:.....

Acknowledgements

Special thanks to Dr. Sherif Mostafa

Abstract

Unplanned mechanical breakdowns are still one of the most expensive problems in industry, and most of the intelligent maintenance systems proposed so far continue to depend on sending raw sensor streams to a remote server before any decision can be taken. That dependency consumes bandwidth, exposes operational data, and adds a delay that is hard to justify when a valve or a bearing is only seconds away from failing.

This thesis develops a lightweight Edge-AIoT model that performs the whole diagnostic cycle inside a low-cost microcontroller instead. Vibration and temperature signals are windowed and reduced to a small set of spectral and statistical descriptors, and an INT8-quantised neural network executed under TensorFlow Lite Micro classifies the condition of the machine into several fault classes rather than raising a single anomaly flag, so that a harmless transient is not treated in the same way as a developing leak. Two further mechanisms are proposed: a fault-tolerant fusion layer that keeps the prediction usable when one of the sensors degrades or drops out, and an asynchronous, lightweight explainability step that reports the features behind each alert without breaking the real-time budget of the device.

The review of fifteen recent studies presented in this work shows that these three requirements — compression, tolerance to imperfect sensing, and on-device explanation — have so far been addressed separately rather than combined in one deployable node. The proposed framework is intended to close that gap, and it will be evaluated on real industrial datasets in terms of classification accuracy, inference latency, memory footprint, and robustness when a sensor fails.

Keywords: *Edge AI; AIoT; TinyML; Predictive Maintenance; Multi-Class Fault Classification; Model Quantisation; Explainable AI (XAI); Industrial IoT.*

List of Publications

Table of Contents

Acknowledgements.....	2
Abstract.....	3
List of Publications.....	4
List of Abbreviations.....	6
List of Figures.....	8
List of Tables.....	9
Chapter 1: Introduction.....	2
1.1 Overview.....	2
1.2 Problem Definition.....	2
1.3 Thesis Objectives.....	3
1.4 Thesis Contribution.....	3
1.5 Thesis Organization.....	4
Chapter 2: Background and Related Work.....	6
2.1 Research Background.....	6
2.1.1 Industrial Maintenance Paradigms.....	6
2.1.2 IoT Evolution and the Bandwidth Bottleneck.....	8
2.1.3 Edge Intelligence (AIoT) and Hardware Optimization.....	9
2.1.4 Multi-Class Fault Analytics and Localized Anomaly Detection.....	12
2.2 Related Work and Literature Review.....	15
Chapter 3 Proposed Approach (Methodology).....	22
Chapter 4 :Experimental Results and Discussion.....	23
Chapter 5 Conclusion and Future Work.....	24
References.....	25

List of Abbreviations

4IR:	Fourth Industrial Revolution
Acc.:	Accuracy
AE:	Autoencoder
AI:	Artificial Intelligence
AIoT:	Artificial Intelligence of Things
ALIF:	Adaptive Leaky Integrate-and-Fire
ANN:	Artificial Neural Network
ANOVA:	Analysis of Variance
ARM:	Advanced RISC Machine
ASIC:	Application-Specific Integrated Circuit
CNN:	Convolutional Neural Network
CWRU:	Case Western Reserve University (Bearing Dataset)
DNN:	Deep Neural Network
DSP:	Digital Signal Processor
EI:	Edge Intelligence
FFT:	Fast Fourier Transform
Float32:	32-Bit Floating-Point Precision
FPGA:	Field-Programmable Gate Array
GAN:	Generative Adversarial Network
GDPR:	General Data Protection Regulation
GPU:	Graphics Processing Unit
GRU:	Gated Recurrent Unit
HMAC-SHA256:	Hash-Based Message Authentication Code with SHA-256
IDPS:	Intrusion Detection and Prevention System
IIoT:	Industrial Internet of Things
INT8:	8-Bit Integer Precision
IoT:	Internet of Things

ISO:	International Organization for Standardization
LIF:	Leaky Integrate-and-Fire
LIME:	Local Interpretable Model-Agnostic Explanations
LoRaWAN:	Long-Range Wide-Area Network
LSTM:	Long Short-Term Memory
MCU:	Microcontroller Unit
ML:	Machine Learning
MQTT:	Message Queuing Telemetry Transport
MSE:	Mean Squared Error
OpEx:	Operational Expenditure
PdM:	Predictive Maintenance
PRISMA:	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RAM:	Random-Access Memory
RF:	Random Forest
RMS:	Root Mean Square
RNN:	Recurrent Neural Network
ROC:	Receiver Operating Characteristic
RUL:	Remaining Useful Life
SHAP:	SHapley Additive exPlanations
SME:	Small and Medium-Sized Enterprise
SNN:	Spiking Neural Network
SRM:	Spike Response Model
SVM:	Support Vector Machine
TF-Lite:	TensorFlow Lite (Micro)
TinyML:	Tiny Machine Learning
TTF:	Time-to-Failure
WAN:	Wide Area Network
XAI:	Explainable Artificial Intelligence

List of Figures

Figure 1: Evolution of industrial maintenance paradigms from run-to-failure to condition-based prediction, with the associated trade-offs in downtime risk, maintenance cost, and dependency on data and artificial intelligence.....	7
Figure 2: Cloud-centric IoT data path compared with the edge AII data path, showing the bandwidth, latency, and privacy constraints that are removed once inference is executed locally.....	8
Figure 3 : Four-tier AII reference architecture for edge-based predictive maintenance, in which the edge intelligence layer constitutes the core scope of this thesis.....	10
Figure 4: TinyML optimisation pipeline leading from a Float32 trained model to an INT8 model executed by TensorFlow Lite Micro on a resource-constrained microcontroller.....	11
Figure 5: On-device multi-class fault classification chain: sliding-window acquisition of the raw signal, spectral and statistical feature extraction, quantised inference, per-class confidence, and millisecond-level local response.....	13
Figure 6: Multi-class fault taxonomy and localized override: raw sensory telemetry is classified on the edge node into four operational classes, and only the critical class triggers an autonomous safety action.....	14
Figure 7: a local device integrating compressed AI algorithms with sensors, successfully reducing response time by 40%.....	15
Figure 8: Explainable AI (xAI): How SHAP and LIME demystify AI decisions.....	17
Figure 9: Vibration-based Predictive Maintenance.....	18

List of Tables

Table 2.2 — Related Work and Literature Review.....20

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Overview

There is no doubt that today's industrial outputs are an essential part of human life and the development of nations across all aspects of life (such as food, health, sports, and education). Without industry, many resources would not have been available. Since the First Industrial Revolution, and passing through the Second, industry has become a top priority for countries, completely reshaping the human lifestyle.

Driven by the significant impact of these factories—where any downtime directly affects the availability of resources and leads to substantial profit losses—factories have been placing a paramount importance on maintenance since the Fourth Industrial Revolution (4IR). Maintenance has shifted from traditional periodic maintenance or reactive maintenance (after a breakdown occurs) to predictive maintenance. These factories have begun integrating Artificial Intelligence (AI) with the Internet of Things (IoT) to provide a powerful tool that collects and analyzes data in real-time, predicting the lifespan of machinery before they stall or fail.

Therefore, the project will be: "**Developing an AIoT model to predict mechanical failures and improve maintenance.**" This project aims to help factories ensure business continuity without disruptions or incurring heavy losses, while incorporating the latest research and experiments in this field to deliver a project that enhances factory efficiency and seamless operations without downtime

1.2 Problem Definition

First: The Mechanical and Economic Problem

Sudden breakdowns of mechanical machinery cause massive financial losses and disrupt production lines. This, in turn, directly impacts both manufacturers and end-consumers regarding the availability of goods and services.

Second: The Current Software Problem

Current solutions rely either on connecting Artificial Intelligence (AI) models to the cloud—which introduces data transmission delays (latency)—or on using compressed models that compromise accuracy, as well as dealing with noisy or distorted data.

1.3 Thesis Objectives

Objective 1: Design a lightweight AIoT model optimized for edge computing devices to ensure real-time mechanical failure prediction.

Objective 2: Develop a fault-tolerant data fusion algorithm capable of handling noisy or missing sensor data (vibration, temperature).

Objective 3: Implement an event-driven mechanism to balance between IoT device energy efficiency and high-fidelity transient signal capture.

Objective 4: Evaluate the proposed framework using real-world industrial datasets to benchmark accuracy and processing latency

1.4 Thesis Contribution

- o **Contribution 1:** Introducing an **Adaptive Edge-AI framework** that self-updates locally on the IoT gateway without relying on cloud retraining.
- o **Contribution 2:** A novel **Fault-Tolerant multi-sensor fusion architecture** that maintains predictive accuracy even during individual sensor failure.
- o **Contribution 3:** An optimized **Asynchronous XAI mechanism** (like lightweight TreeSHAP) that explains failure root-causes to mechanical engineers without violating real-time execution constraints.

1.5 Thesis Organization

The remainder of this thesis is organized as follows:

- o **Chapter 2 (Background and Related Work)** provides the theoretical background of AIoT and mechanical fault analysis, alongside a critical review of related work.
- o **Chapter 3 Proposed Approach (Methodology)** details the proposed adaptive AIoT model architecture, sensor fusion, and edge deployment strategies.
- o **Chapter 4 (Experimental Results and Discussion)** presents the software implementation setup, dataset description, and performance evaluation metrics.
- o **Chapter 5 (Conclusion & Future Work)** summarizes the core findings of this thesis and outlines potential directions for future research."

Chapter 2

Background and Related Work

Chapter 2

Background and Related Work

2.1 Research Background

2.1.1 Industrial Maintenance Paradigms

In modern industrial engineering and infrastructure management, maintaining the structural and operational integrity of mechanical assets is paramount. Historically, facilities relied heavily on Reactive Maintenance (Run-to-Failure), which entails repairing equipment only after a breakdown occurs. While structurally simple, this paradigm incurs severe financial deficits due to unexpected downtime, safety hazards, and catastrophic secondary damage to adjacent machinery.

To mitigate these risks, industries transitioned to Preventive Maintenance, which implements schedule-based servicing. Although more structured, preventive schedules often result in “over-maintenance,” causing premature replacement of functional components and unnecessary operational overhead. This economic and logistical imbalance has accelerated the global adoption of Predictive Maintenance (PdM)—a proactive approach that evaluates the actual real-time condition of an asset to forecast anomalies before failure occurs. Figure 1 summarises this progression together with the downtime, cost, and data-dependency trade-offs that characterise each paradigm.

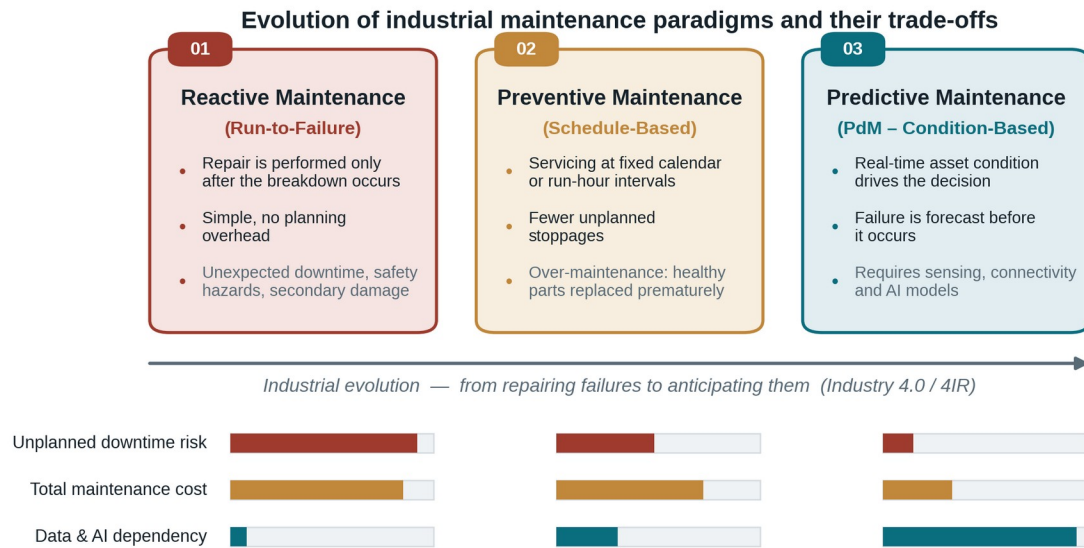


Figure 1: Evolution of industrial maintenance paradigms from run-to-failure to condition-based prediction, with the associated trade-offs in downtime risk, maintenance cost, and dependency on data and artificial intelligence.

As shown in Figure 1, the three paradigms are arranged along the industrial evolution axis, and each panel states the triggering condition of the maintenance action, its principal advantage, and its main drawback. The three horizontal indicators beneath the panels compare the paradigms qualitatively on three dimensions. The risk of unplanned downtime declines sharply from reactive to predictive maintenance, because failures are anticipated rather than discovered. The total maintenance cost peaks under the reactive scheme, remains high under preventive servicing because healthy components are discarded prematurely, and falls once interventions are triggered by the measured condition of the asset. Conversely, the dependency on sensing and artificial intelligence rises steeply in the opposite direction, which explains why predictive maintenance became practical only after industrial sensing and machine learning matured. The figure therefore establishes the central premise of this thesis: the benefits of the predictive paradigm are obtained at the price of a demanding data and inference infrastructure, and it is the cost and latency of that infrastructure that the remainder of the chapter examines.

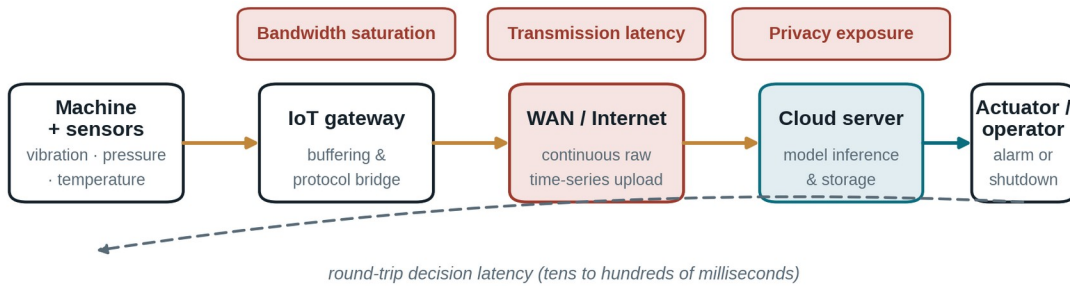
2.1.2 IoT Evolution and the Bandwidth Bottleneck

The realization of Predictive Maintenance was significantly advanced by the Internet of Things (IoT). Deploying specialized wireless sensory networks allowed engineers to stream physical machinery telemetry—such as fluid pressure oscillations, mechanical vibration amplitudes, and heat gradients—to centralized storage frameworks.

However, classic cloud-centric IoT architectures suffer from critical constraints when scaled across extensive facilities. Streaming continuous, high-frequency time-series data to remote cloud servers demands immense network bandwidth, exposes data to privacy vulnerabilities, and introduces transmission latencies. In high-pressure hydraulic setups or high-speed rotational systems, a delay of milliseconds in anomaly detection can mean the difference between a minor leak and a catastrophic rupture. Figure 2 contrasts the two data paths and identifies the points at which these constraints arise.

(a) Cloud-centric IoT architecture

Raw high-frequency telemetry is streamed to a remote server, and the decision returns over the WAN



(b) Edge AIoT architecture adopted in this thesis

Inference runs on the microcontroller at the point of measurement, and the cloud becomes optional

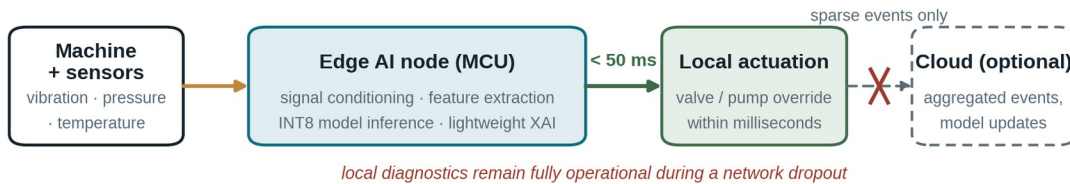


Figure 2: Cloud-centric IoT data path compared with the edge AIoT data path, showing the bandwidth, latency, and privacy constraints that are removed once inference is executed locally.

Figure 2 is divided into two parts. Part (a) traces the conventional cloud-centric path, in which the machine and its sensors deliver raw telemetry to a gateway, the gateway uploads a continuous time-series stream over the wide area network, and the decision is produced by a remote server before it travels back to the actuator or the operator. Three constraints are annotated along that path, namely bandwidth saturation caused by the volume of high-frequency data, transmission latency introduced by the round trip, and the privacy exposure that follows from storing raw operational data outside the facility. The dashed return arrow represents the complete round-trip decision latency, which for the reviewed deployments ranges from tens to hundreds of milliseconds. Part (b) shows the architecture adopted in this thesis, in which conditioning, feature extraction, quantised inference, and explanation are executed inside the microcontroller itself, so that the actuation command is issued locally within milliseconds. The cloud is retained only as an optional destination for aggregated events and model updates, which is why the diagram marks the link to it as severed without interrupting the diagnostic function: local protection continues to operate during bandwidth saturation or a complete network outage.

2.1.3 Edge Intelligence (AIoT) and Hardware Optimization

To resolve cloud over-dependence, the modern engineering paradigm introduces AIoT (Artificial Intelligence of Things), which embeds cognitive machine learning capabilities directly into localized hardware layers, establishing Edge Intelligence. By migrating model inference to the edge, sensory nodes can interpret raw physical signals right at the point of measurement. The resulting layered organisation of such a system is illustrated in Figure 3.

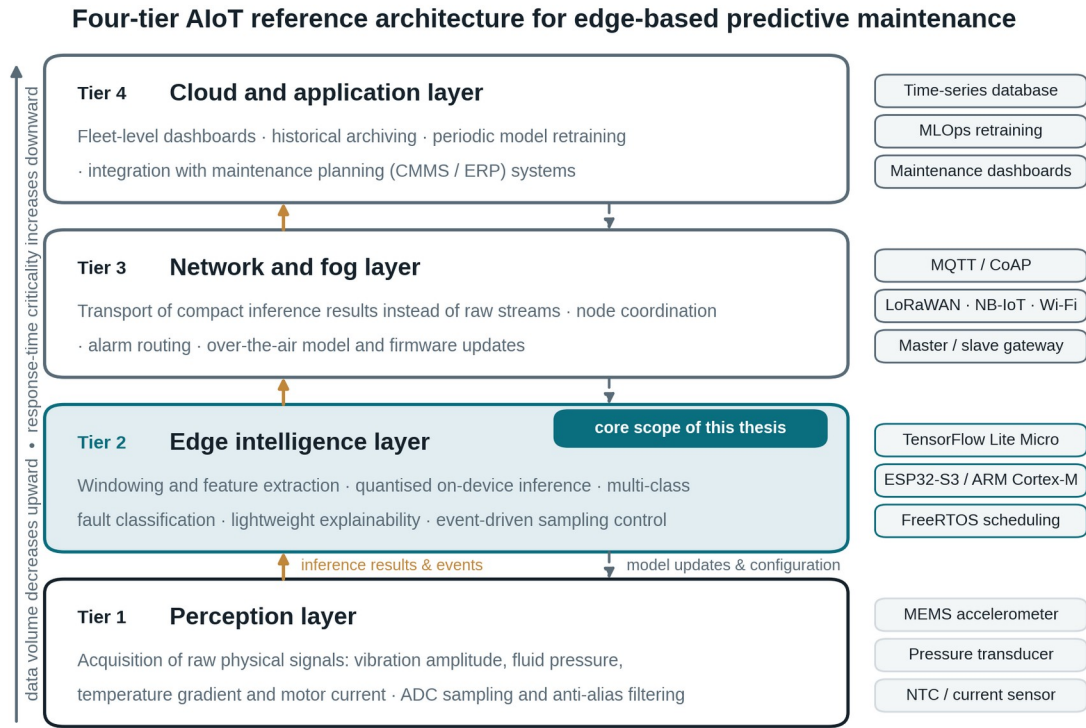


Figure 3 : Four-tier AIoT reference architecture for edge-based predictive maintenance, in which the edge intelligence layer constitutes the core scope of this thesis.

Figure 3 decomposes the system into four tiers. The perception tier acquires the raw physical signals and performs sampling and anti-alias filtering. The edge intelligence tier, highlighted as the core scope of this thesis, executes windowing, feature extraction, quantised inference, multi-class classification, lightweight explainability, and event-driven sampling control. The network and fog tier transports compact inference results rather than raw streams, coordinates the nodes, routes alarms, and delivers over-the-air model and firmware updates. The cloud and application tier finally provides fleet-level dashboards, historical archiving, periodic retraining, and integration with maintenance planning systems. The right-hand column lists the enabling technology associated with each tier, while the two vertical arrows between the tiers distinguish the upward flow of inference results and events from the downward flow of model updates and configuration. The vertical annotation on the left states the governing design principle of the architecture: data volume decreases as information ascends the stack, whereas the criticality of the response time increases as it descends toward the machine.

Recent breakthroughs in deep learning optimization—such as Model Quantization (converting weights from Float32 to INT8 precision) and embedded runtimes like TensorFlow Lite Micro—have made it possible to deploy complex mathematical models onto low-cost, resource-constrained microcontrollers (e.g., the ESP32 platform). This ensures that heavy computational diagnostics no longer require expensive industrial computers or active internet dependencies. Figure 4 outlines the corresponding optimisation pipeline and its effect on the deployability of the model.

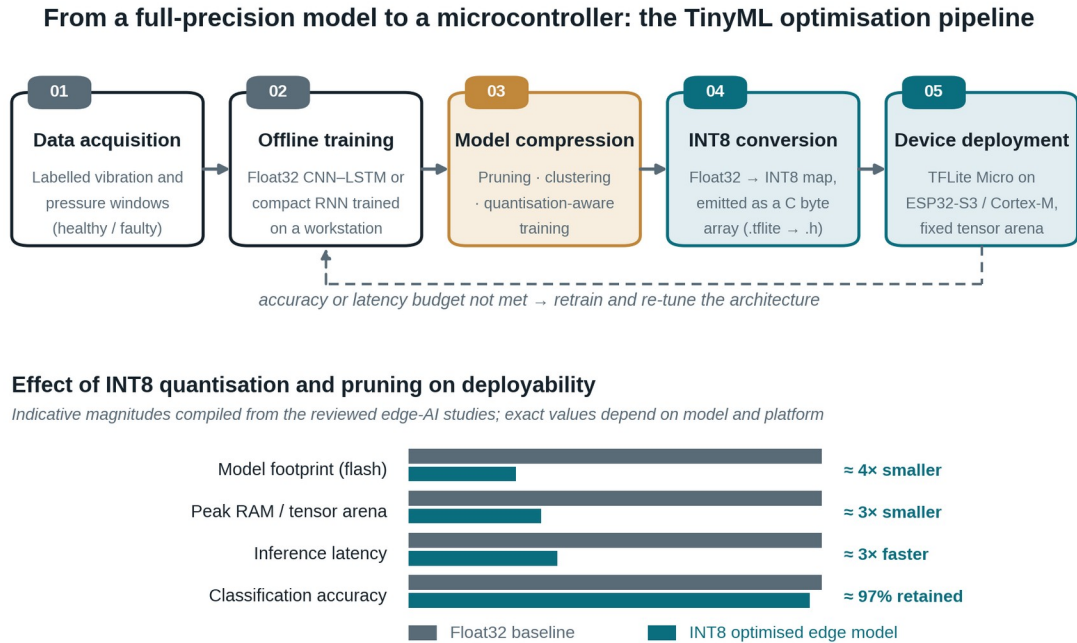


Figure 4: TinyML optimisation pipeline leading from a Float32 trained model to an INT8 model executed by TensorFlow Lite Micro on a resource-constrained microcontroller.

The upper part of Figure 4 presents the five sequential stages of the pipeline: acquisition of labelled healthy and faulty windows, offline training of a Float32 network on a workstation, compression through pruning and quantisation-aware training, conversion of the weights to INT8 and emission of the model as a C byte array, and finally deployment under the TensorFlow Lite Micro interpreter with a fixed tensor arena. The dashed return path expresses the iterative nature of the process, since a model that violates the accuracy or latency budget of the target device must be retrained or re-architected rather than merely re-converted. The lower part quantifies why this pipeline is indispensable on constrained hardware. Relative to the Float32 baseline, the compiled evidence from the reviewed

studies indicates roughly a fourfold reduction in flash footprint, a threefold reduction in peak RAM, and a threefold reduction in inference latency, while approximately ninety-seven per cent of the original classification accuracy is retained. This asymmetry between a large resource saving and a small accuracy penalty is the technical justification for adopting a quantised edge model in the proposed framework.

2.1.4 Multi-Class Fault Analytics and Localized Anomaly Detection

The ultimate efficacy of an AIoT predictive framework relies on its algorithmic capacity to not only flag general anomalies but to accurately map and classify specific degradation behaviors concurrently (Multi-Class Fault Classification). In mechanical and hydraulic domains, raw sensory signatures (such as fluid pressure telemetry or vibration frequencies) must be classified at the edge node to distinguish between a “benign transient pressure drop,” a “structural line leak,” and a “critical valve failure.”

By executing lightweight, highly optimized on-device deep learning algorithms, a standalone microcontroller can analyze streaming time-series data locally and isolate specific failure roots in real-time. This localized processing enables the system to issue instant diagnostic responses or initiate safety overrides—such as shutting off local valves or pumps—within milliseconds. Crucially, this advanced diagnostic protection remains fully operational even during complete communication network dropouts, ensuring robust, low-power, and continuous hardware asset reliability. The signal-processing chain that produces this behaviour is depicted in Figure 5, and the resulting fault taxonomy and its associated override are summarised in Figure 6.

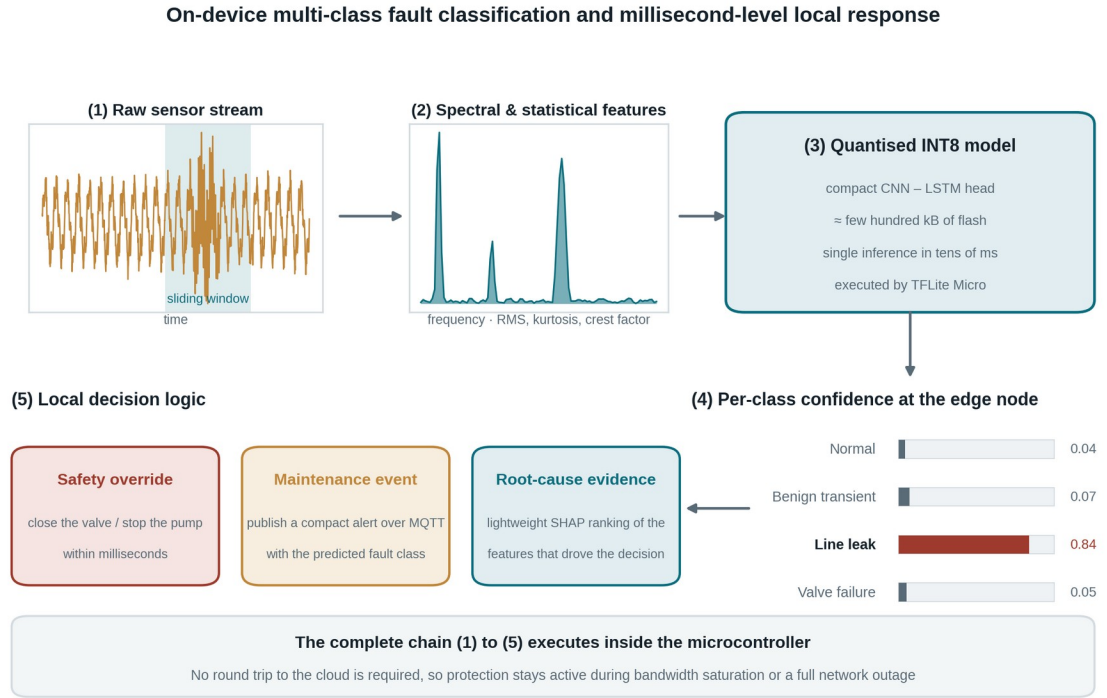


Figure 5: On-device multi-class fault classification chain: sliding-window acquisition of the raw signal, spectral and statistical feature extraction, quantised inference, per-class confidence, and millisecond-level local response.

Figure 5 follows a single measurement through five stages. In the first stage the node samples a continuous stream and isolates a sliding window, shaded in the time-domain plot, in which a transient event is visible. In the second stage that window is transformed into a compact descriptor combining spectral content with statistical indicators such as root mean square, kurtosis, and crest factor, so that the classifier operates on a few informative values instead of thousands of raw samples. The third stage submits the descriptor to the quantised INT8 model executed by TensorFlow Lite Micro, whose footprint of a few hundred kilobytes allows it to reside entirely in the flash memory of the microcontroller. The fourth stage yields a confidence value for every fault class rather than a single binary anomaly flag; in the illustrated case the structural line leak dominates the distribution. The fifth stage converts that distribution into three parallel outcomes, namely an immediate safety override acting on the valve or the pump, a compact maintenance event published over MQTT, and a lightweight ranking of the features that drove the decision, which supplies the maintenance engineer with root-cause evidence. The band at the foot of

the figure states the property that distinguishes this design: the entire chain executes inside the microcontroller, so protection is preserved when connectivity is degraded or absent.

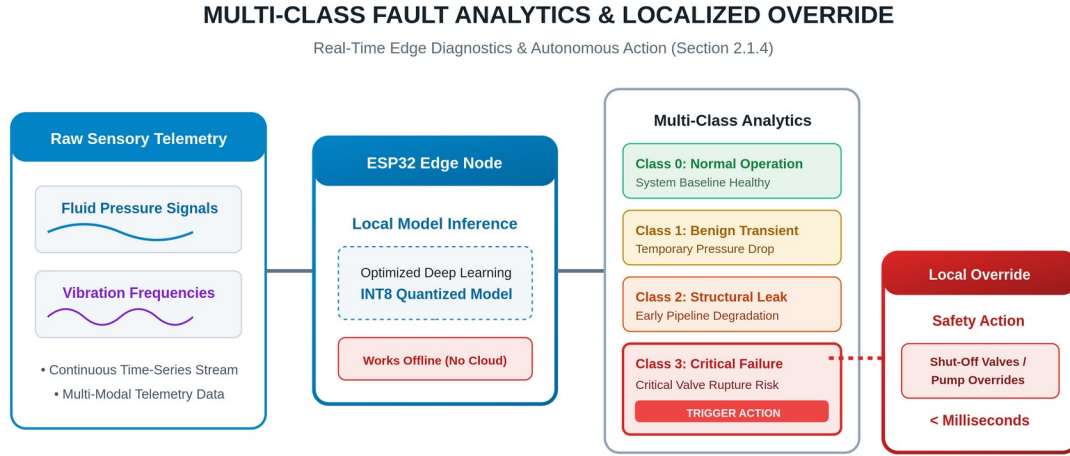


Figure 6: Multi-class fault taxonomy and localized override: raw sensory telemetry is classified on the edge node into four operational classes, and only the critical class triggers an autonomous safety action.

Figure 6 presents the same mechanism from the perspective of the fault taxonomy. The leftmost panel identifies the multi-modal telemetry that feeds the node, namely fluid pressure signals and vibration frequencies arriving as a continuous time series. The second panel represents the ESP32 edge node executing the optimised INT8 model, and the badge within it emphasises that the node remains functional without any cloud connection. The third panel enumerates the four output classes that the model must separate concurrently. Class zero corresponds to healthy baseline operation, class one to a benign transient pressure drop that must deliberately not raise an alarm, class two to a structural leak indicating early pipeline degradation, and class three to a critical valve failure. The distinction between class one and class two is the practical value of multi-class analytics, because a binary anomaly detector would treat both as a single deviation and would therefore generate false alarms that erode the confidence of the maintenance team. Only class three propagates to the rightmost panel, where the local override issues a shut-off command to the valves or the pumps in less than a millisecond-scale interval. The figure thus expresses the governing policy of the proposed framework: diagnosis is graduated

across several severities, whereas autonomous physical intervention is reserved for the single class that carries an immediate safety consequence.

2.2 Related Work and Literature Review

A major challenge in early predictive-maintenance systems involved relying on cloud computing for both data storage and analysis, where data-transmission latency severely impacted the accuracy of mechanical-fault predictions. To address this, J. S. Kang, M. J. Kim, and J. W. Kim, deployed a local device integrating compressed AI algorithms with sensors, successfully reducing response time by 40% (Figure 7), although this approach decreased fault-prediction accuracy [1].

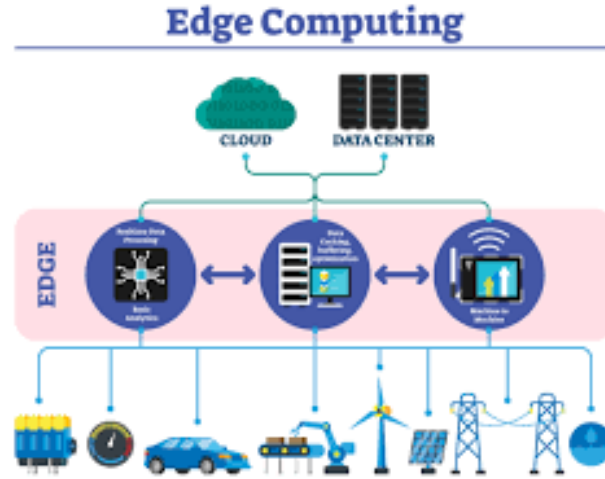


Figure 7: a local device integrating compressed AI algorithms with sensors, successfully reducing response time by 40%.

Extending this line of work to a specific mechanical asset, Montes-Sánchez et al. evaluated 939 hybrid RNN architectures (LSTM and GRU variants) across four embedded MCUs to identify the lightest model for early tube-wear detection in peristaltic pumps, using six multi-modal sensors downsampled to 12 Hz, and achieved near-perfect F1-scores of 0.99–1.00 with ultra-low latency; the work, however, remained confined to static operating conditions and did not address dynamic pressure or flow variation [2]. Further expanding on sensor configuration, Zhao and Wang addressed the limitation of relying on a single sensor by incorporating additional sensors and a Long Short-Term Memory (LSTM) recurrent neural network to track equipment degradation, which improved prediction

accuracy; nevertheless, a challenge arose from the assumption that all sensors operate at full efficiency, leading to unstable or incorrect results when a sensor degrades [3].

Data imbalance poses a further obstacle, since fault data typically constitutes a mere 1% of the total dataset. To resolve this, Berghout et al. utilized Generative Adversarial Networks (GANs) to generate synthetic fault data, which effectively enhanced the system's ability to detect rare faults and boosted recall from 60% to 92%; yet a research gap remained, as the generated data failed to simulate the continuous and gradual degradation of mechanical equipment over time [4]. In a related effort to improve robustness under imperfect sensing conditions, Desikan et al. combined statistical anomaly detection with HMAC-SHA256 cryptographic validation and a fault-adjusted Dempster-Shafer fusion of five sensor-specific models (MobileNet, EfficientNet, Random Forest, SVM, and Decision Tree) for fire-risk detection in oil-and-gas IIoT environments, sustaining 67.8% accuracy even with four of five sensors failed; the evaluation, however, relied mainly on simulated rather than live field data [5]. Additionally, to tackle the problem of high energy consumption in sensors resulting from high-frequency sampling, Zhang et al. developed an intelligent scheduling algorithm based on adaptive sampling combined with lightweight machine-learning models embedded within the sensors, connected via LoRaWAN networks. While this successfully reduced energy consumption by 35%, it led to lower sampling rates, causing the loss of transient signals that could trigger major mechanical failures [6].

A further significant challenge lies in engineers' trust in AI models, where the lack of clarity regarding the reasons behind generated alerts has hindered confidence in these systems. To tackle this, Silva et al. integrated Explainable AI (XAI) techniques, specifically the SHAP and LIME methods (Figure 8), which enhanced user trust and improved maintenance procedures; nonetheless, calculating SHAP values requires high computational power, resulting in prolonged execution times that pose a critical risk to industrial plants and time-sensitive faults [7].

Explainable AI: Methods and Applications

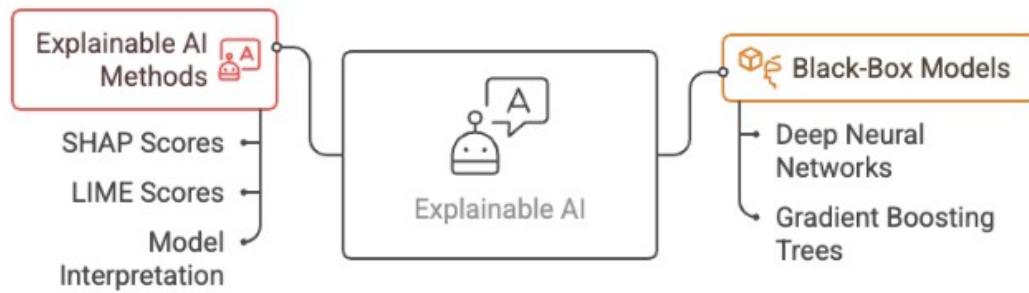


Figure 8: Explainable AI (xAI): How SHAP and LIME demystify AI decisions

Building on this concern for real-time constraints, Kumar et al. later embedded a lighter, SHAP-based XAI mechanism directly into an INT8-quantized edge classifier for parallel multi-fault prediction and Time-to-Failure (TTF) tracking, deployed via TensorFlow Lite Micro on an ESP32-S3 with a confidence-based Edge-Cloud fallback mechanism; the system achieved 42.3 ms average inference latency while retaining 97.3% of cloud-model precision, though it was tested only on a single, isolated MCU in a static environment [8].

Model compression for resource-constrained hardware has also been explored across different mechanical domains. Razaq et al. proposed a hybrid CNN-LSTM classifier paired with a Health Index Prognostic Model for Remaining Useful Life (RUL) estimation in grid inverters, compressed via TensorFlow Lite pruning and quantization and deployed on ARM-based nodes over MQTT; the approach cut processing delay from 85 ms to 18 ms while reaching 96.8% diagnostic accuracy, though it relied partly on synthetic fault data due to sparse real-world records [9]. In a comparable effort targeting cyber-physical smart environments, Garg et al. designed lightweight CNN and RNN models within a 4-tier architecture using MQTT communication and TensorFlow Lite deployment across Raspberry Pi and NVIDIA Jetson Nano boards, achieving 97.1% anomaly-detection precision and cutting latency from 180 ms to 45 ms, though the design targets relatively heavy compute nodes rather than bare-metal microcontrollers [10]. At a smaller hardware scale, Arciniegas et al. combined FFT-based spectral feature extraction with a compact neural network (3-20-10-3) and K-Means clustering to detect abnormal motor-bearing

vibrations on a Saeed XIAO ESP32S3 with an MPU6050 accelerometer, achieving 96.5% accuracy and 25 ms inference time while sustaining 98% accuracy over two months of real-world operation; the system, however, remained sensitive to environmental noise and was validated on a single motor only [11].

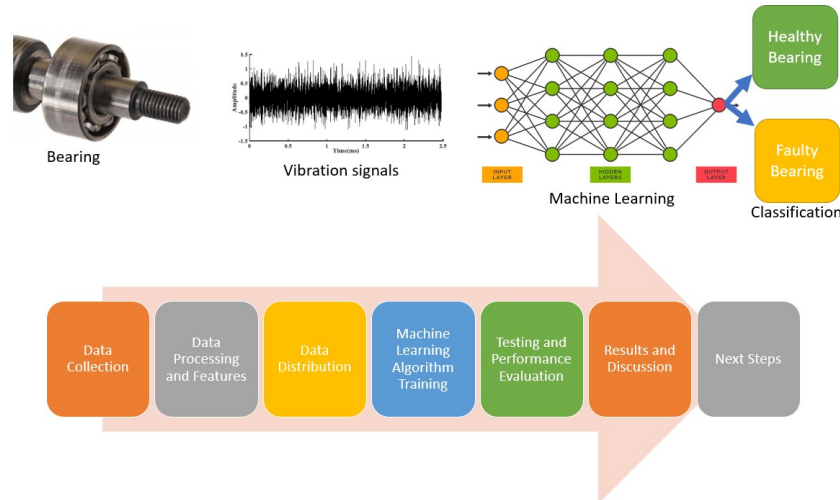


Figure 9: Vibration-based Predictive Maintenance

Several recent surveys situate these individual contributions within a broader landscape. Vasilache et al. reviewed 12 Spiking Neural Network (SNN) and 8 Artificial Neural Network (ANN) approaches for low-power, vibration-based predictive maintenance, finding the CWRU bearing dataset to be the most common benchmark and noting that feed-forward SNNs favored shallow architectures (1–3 layers) while ANNs used deeper ones (4+ layers) (Figure 9), with only a handful of the reviewed works validated on real hardware and no standardized benchmark yet available for the domain [12]. Njor et al. similarly conducted a holistic review of the TinyML stack for predictive maintenance, covering hardware, toolchains (TensorFlow Lite Micro, Edge Impulse, STM32Cube.AI), and nine model-optimization techniques, and found that only three usable predictive-maintenance datasets exist, with research disproportionately focused on model optimization while data-centric methods and toolchain benchmarking remain underexplored [13]. Reis extended the edge-intelligence discussion beyond mechanical assets into last-mile delivery logistics through a PRISMA 2020 systematic review combined with a two-round Delphi expert validation, formalizing the deployment pillars of Edge Intelligence (EI) and finding measurable gains in route optimization and latency; the

resulting framework, however, remains exploratory, lacks concrete edge-algorithm development, and faces adoption hesitancy among European SMEs due to cost and skill barriers [14]. Finally, Khurshid et al. broadened the lens further to security and privacy, systematically reviewing roughly 2,500 papers down to 250 to map the security and privacy landscape of AIoT-based surveillance systems through an ISO 27001 lens, supported by real-world breach case studies (Verkada, Snowflake, Ring, and an Amazon France GDPR fine); the review found that only about 30% of AIoT devices currently meet interoperability standards and that this qualitative synthesis lacks an original prototype or empirical validation [15] — underscoring that a lightweight, fault-tolerant, and explainable Edge-AIoT framework, of the kind pursued in this thesis, must still be developed and empirically validated rather than assumed from theory alone.

Table 2.2 — Related Work and Literature Review

Ref.	Author & Year	Objective	Method	Key Results	Gap / Opportunity
[1]	J. S. Kang, M. J. Kim, and J. W. Kim. (2021)	Reduce cloud reliance and data-transmission lag.	Lightweight quantized CNN; IIoT wireless vibration sensors on a Raspberry Pi edge gateway.	Latency reduced 40%; 89% fault-classification accuracy.	Quantization lost accuracy on overlapping faults; static model — opportunity for online learning.
[2]	Montes-Sánchez et al. (2025)	Early pump tube wear detection; prevent sudden fluid runtime failures.	Evaluated 939 RNN variants (LSTM/GRU, 2–64 hidden units); 6 multi-modal sensors on 4 embedded MCUs @ 12 Hz.	F1-score 0.99–1.00; ultra-low latency edge inference.	Confined to static settings — opportunity for dynamic pressure/flow handling.
[3]	Zhao, M., et al. (2020)	Predict machinery RUL via multiple data streams.	Multi-sensor data-fusion algorithm + LSTM over a heterogeneous IIoT network (vibration, temperature, current).	High RUL estimation accuracy; 15% MSE reduction.	Assumes 100% sensor reliability — opportunity for fault-tolerant fusion mechanisms.
[4]	Berghout, T., et al. (2021)	Address severe class imbalance (<1% failure data).	GANs for synthetic fault-data generation; IoT gear/motor data streamed to a local training server.	Recall for rare failures boosted from 60% to 92%.	GAN snapshots fail to mimic continuous degradation — move to sequential time-series simulation.
[5]	Desikan et al. (2025)	Handle faulty/noisy multi-sensor fire data in oil-and-gas IIoT; reduce false alarms.	Dynamic-threshold anomaly detection + HMAC-SHA256 validation; Dempster-Shafer fusion of 5 sensor-specific models on a Raspberry Pi.	98.9% accuracy (no anomaly); 75.8% at 40% anomaly rate; 67.8% with 4 of 5 sensors failed.	Evaluated mainly on simulated data — opportunity for live field validation.
[6]	Zhang, L., et al. (2022)	Reduce battery consumption of wireless IoT sensors.	Adaptive-sampling algorithm + lightweight embedded ML on battery-powered LoRaWAN sensors.	IoT node energy reduced 35%; extended battery life.	Lower sampling rates missed transient signals — opportunity for event-driven sampling.
[7]	Silva, R., et al. (2023)	Resolve engineers' distrust of “black-box” deep-learning alerts.	Integrated XAI (SHAP and LIME) on an AIoT dashboard mapping sensors to alerts.	Enhanced user trust; pinpointed anomalous sensor attributes.	Heavy SHAP computation causes real-time latency — compress/optimize XAI for edge execution.
[8]	Kumar et al. (2026)	Eliminate cloud latency/downtime for machinery continuity.	INT8 edge classifier with TTF tracking + SHAP-based XAI; TF-Lite Micro on ESP32-S3 with Edge-Cloud fallback.	42.3 ms inference latency; retained 97.3% of cloud-model precision.	Single isolated MCU — opportunity for a coordinated multi-point sensory grid.
[9]	Razaq et al. (2026)	Fix grid-inverter fault diagnosis and OpEx; bridge cloud-reliant data silos.	Hybrid CNN-LSTM classifier + Health Index RUL model; TF-Lite pruning/quantization on ARM via MQTT.	96.8% diagnostic accuracy; latency cut from 85 ms to 18 ms.	Relied on synthetic fault data — opportunity for Master/Slave distributed deployment.
[10]	Garg et al. (2026)	Resolve cloud telemetry lag/cost; enable low-latency local decisions.	Lightweight CNN/RNN in a 4-tier architecture; TF-Lite on Raspberry Pi & Jetson Nano via MQTT.	97.1% anomaly-detection precision; latency dropped from 180 ms to 45 ms.	Heavy compute nodes — opportunity to adapt the 4-tier design to bare-metal ESP32.
[11]	Arciniegas et al. (2025)	Real-time motor-bearing fault detection at the edge, avoiding cloud latency/privacy risk.	Lightweight DNN (3-20-10-3) on FFT features + K-Means; 8-bit quantization on Sseed XIAO ESP32S3 + MPU6050.	96.5% accuracy, 25 ms inference; outperformed RF/SVM; 98% accuracy sustained over 2 months.	Noise sensitivity; single-motor validation — opportunity for sensor fusion and federated learning.
[12]	Vasilache et al. (2024)	Survey low-power on-device neural computation for vibration-based PdM.	Comparative review of 12 SNN and 8 ANN approaches (LIF/ALIF/SRM, CNN,	CWRU dataset most common; SNNs shallow (1–3 layers), ANNs	No standardized benchmark for low-power vibration-based PdM.

Ref.	Author & Year	Objective	Method	Key Results	Gap / Opportunity
			LSTM, AE); hardware on DYNAP-SE, STM32F407, FPGAs.	deeper (4+ layers); little real-hardware validation.	
[13]	Njor et al. (2024)	Holistic review of the TinyML stack for predictive maintenance.	Reviewed 9 model-optimization techniques and major TinyML toolchains across MCUs, DSPs, FPGAs, ASICs, GPUs.	Only 3 usable PdM datasets found; optimization over-studied relative to data-centric methods.	No standardized PdM benchmarks/datasets — opportunity in energy harvesting, XAI, standardization.
[14]	Reis (2025)	Reduce last-mile logistics overhead; formalize Edge Intelligence (EI) pillars.	PRISMA 2020 literature review; 2-round Delphi expert validation.	Optimized routing and latency; SME cost/skill adoption barriers noted.	Purely theoretical, no code/silicon tests — embed strategies into a low-cost Edge AI node.
[15]	Khurshid et al. (2025)	Map the security/privacy landscape of AIoT-based surveillance systems.	Systematic review (~2,500 → 250 papers) via IEEE/ACM/Scopus/WoS; ISO 27001 lens + real breach case studies.	Adversarial attacks cause major accuracy drops; proposed IDPS/zero-trust solutions report >97% accuracy.	Qualitative synthesis only, no prototype — opportunity for a lightweight zero-trust/IDPS framework for constrained AIoT nodes.

Chapter 3 Proposed Approach (Methodology)

Chapter 4 :Experimental Results and Discussion

Chapter 5 Conclusion and Future Work

References

- [1] J. S. Kang, M. J. Kim, and J. W. Kim, "Fast and Autonomous Tuning of Ultra-Lightweight 1-D CNN for Real-Time Motor Fault Diagnosis," *Sensors*, vol. 68, no. 11, pp. 11452-11461, 2021, <https://doi.org/10.1109/s21144676>.
- [2] M. Montes-Sánchez et al., "Predictive maintenance edge AI for peristaltic pumps," *IEEE Trans. Reliability*, 2025, <https://doi.org/10.1109/TR.2024.3488963>.
- [3] Y. Zhao and Y. Wang, "Remaining useful life prediction for multi-sensor systems using a novel end-to-end deep-learning method," *Measurement*, vol. 182, p. 109685, 2021, <https://doi.org/10.1016/j.measurement.2021.109685>.
- [4] T. Berghout, L. H. Mouss, et al., "A deep generative adversarial network for remaining useful life prediction of rotating machinery with class imbalance," vol. 101, p. 107031, 2021, <https://doi.org/10.1016/j.asoc.2020.107031>.
- [5] J. Desikan, S. K. Singh, A. Jayanthiladevi, S. Bhushan, V. Rishiwal, and M. Kumar, "Hybrid machine learning-based fault-tolerant sensor data fusion and anomaly detection for fire risk mitigation in IIoT environment," *Sensors*, vol. 25, no. 7, p. 2146, 2025, <https://doi.org/10.3390/s25072146>.
- [6] L. Zhang et al., "Energy-Efficient Adaptive Sampling and Lightweight Machine Learning for LoRaWAN-Based Wireless Sensor Nodes in Industrial IoT," *Comput. Electr. Eng.*, vol. 9, p. 12345, 2022, <https://doi.org/10.1016/10.1109/JIOT.2022.3141201>.
- [7] R. Silva et al., "Explainable Artificial Intelligence (XAI) for Predictive Maintenance in Industrial IoT", vol. 19, no. 6, p. 7845, 2023, <https://doi.org/10.1109/TII.2023.3214567>.
- [8] S. Kumar et al., "Design of a lightweight edge-AI system for predictive maintenance on ESP32-S3," *Appl. Sci.*, vol. 16, no. 11, p. 5287, 2026, <https://doi.org/10.3390/app16115287>.
- [9] M. Razaq, S. Zaman, and F. Karim, "Predictive maintenance framework for solar inverters and smart grid assets," *Saudi J. Eng. Technol.*, vol. 11, no. 4, 2026, <https://doi.org/10.36348/sjet.2026.v11i04.013>.
- [10] R. Garg et al., "Edge AI-enabled IoT architecture for real-time data processing in cyber-physical smart environments," *Int. J. Adv. AI Mach. Learn.*, vol. 6, no. 2s, pp. 700–711, 2026, <https://doi.org/10.51483/IJAIML.6.2s.2026.700-711>.
- [11] S. Arciniegas, D. Rivero, and J. Piñán et al., "IoT device for detecting abnormal vibrations in motors using TinyML," *Discov. Internet Things*, vol. 5, no. 41, 2025, <https://doi.org/10.1007/s43926-025-00142-4>.
- [12] A. Vasilache et al., "Low-power vibration-based predictive maintenance for Industry 4.0 using neural networks: A survey," <https://arxiv.org/abs/2408.00516>, 2024.

- [13] E. Njor, M. A. Hasanpour, J. Madsen, and X. Fafoutis, “A holistic review of the TinyML stack for predictive maintenance,” IEEE Access, vol. 12, pp. 184861–184882, 2024, <https://doi.org/10.1109/ACCESS.2024.3512860>.
- [14] C. Reis, “Edge intelligence in enhancing last-mile delivery logistics,” IEEE Access, 2025, <https://doi.org/10.1109/ACCESS.2025.3570894>.
- [15] K. Khurshid, K. Khurshid, M. U. Hadi, M. Al Bataineh, and N. Saeed, “Securing AIoT surveillance: Techniques, challenges, and solutions,” IEEE Open J. Commun. Soc., vol. 6, pp. 6517–6550, 2025, <https://doi.org/10.1109/OJCOMS.2025.3593311>.