



**Politecnico
di Torino**

ScuDo

Scuola di Dottorato - Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Computer and Control Engineering (38th cycle)

A Data-Centric Edge Framework for Predictive Maintenance in Industrial IoT

By

Pietro D'Agostino

Supervisor:

Prof. Massimo Violante

Industrial Tutor:

Ing. Gianpaolo Macario

Politecnico di Torino

2026

Declaration



The project has been funded by the PNRR, Mission 4, Component 2 “From Research to Business” – Investment 3.3 “Introduction of innovative PhD programs that meet the innovation needs of companies and promote the hiring of researchers by businesses,” through Ministerial Decree No. 352 of April 9, 2022;

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Pietro D’Agostino
2026

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

*To the patience and love of Mirka and Mostrina, and to the hands that solve what
Tacchino cannot.*

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Massimo Violante, for his invaluable guidance and professional expertise throughout these years. His insight and encouragement have been fundamental to my growth as a researcher and the completion of this work.

I am equally grateful to my industrial tutor, Gianpaolo Macario, for the opportunity to bridge the gap between academic research and industrial application. His mentorship and the trust he placed in my work have been essential to this project.

Special thanks go to my colleagues and friends at AROL and the B-AROL-O Team for the shared challenges, technical discussions, and for creating such a stimulating environment. I also thank the team at AROL HEXAGONE: Virginie, for her leadership and warm welcome, and Emilie, whose constant support and presence have been a true cornerstone of my time there.

To my friends, thank you for the necessary distractions and for your unwavering encouragement, reminding me of the world that exists beyond these pages.

Finally, my deepest thanks go to my family. To my mother, my father, and my sister: thank you for your unconditional love and for being my constant point of reference. Your belief in me, even when I doubted myself, has been the true foundation of this journey. Everything I have achieved is thanks to the strength and the home you have provided.

I could not have done this without you.

Abstract

The Industrial Internet of Things (IIoT) generates vast streams of heterogeneous industrial data, yet extracting actionable insights in real time remains challenging due to the limitations of proprietary platforms and non-scalable academic prototypes. This dissertation introduces a data-centric approach for industrial edge computing, focusing on feature extraction, preprocessing, and customizable analytics directly on raw data streams.

The primary contribution is a framework that enables third-party developers to define and deploy custom transformations and predictive models on industrial data, decoupling analytics from original equipment manufacturer (OEM) constraints: the Concentrator. By studying both slow-varying signals (e.g., temperature) and high-frequency signals (e.g., vibrations), meaningful features are generated, facilitating predictive maintenance, anomaly detection, and operational optimization.

Experimental validation on an AROL capping machine demonstrated that containerized, edge-deployed analytics can achieve forecasting accuracies above 99% for slow signals and F1-scores over 98% for complex vibration anomalies. The methodology highlights how raw industrial data can be systematically understood, transformed, and personalized to support flexible, open, and collaborative IIoT ecosystems.

Overall, this research establishes a practical foundation for intelligent, data-driven industrial operations, emphasizing feature engineering, edge-level customization, and actionable insight generation, rather than mere data collection.

Contents

List of Figures	x
List of Tables	xiii
Nomenclature	xv
1 Introduction	1
1.1 The Industrial Internet of Things (IIoT): A Paradigm Shift in Manufacturing	1
1.1.1 The Digital Transformation of Industry	1
1.1.2 Evolution of Maintenance Strategies	2
1.1.3 Fundamental Computing Architectures: Cloud, Edge, and Fog	3
1.2 Problem Definition and Research Gaps	4
1.2.1 Limitations of Current Data-Centric Infrastructures	4
1.2.2 From Data Collection to Data Value	4
1.3 Original Contributions of the Dissertation	5
1.4 Application to Predictive Maintenance	6
1.4.1 Justification for Signal Selection	7
1.5 Industrial Impact	7
1.6 Scope and Limitations	8
1.7 Structure of the Thesis	8

2	Related Works and Background	9
2.1	Industrial Internet of Things (IIoT)	10
2.2	Computing Architectures in IIoT	13
2.2.1	Cloud-Centric Architectures	13
2.2.2	Edge Computing	15
2.2.3	Fog Computing	16
2.2.4	Comparative Overview	18
2.3	Fog and Edge Implementations in Industrial Practice and Research .	18
2.3.1	Industrial Implementations	19
2.3.2	Academic and Experimental Prototypes	20
2.3.3	Synthesis and Positioning of the Proposed Work	21
2.4	Data-Centric Perspective on Implementations	22
2.5	Predictive Maintenance Techniques in IIoT	22
2.5.1	Data Sources and Signal Types	23
2.5.2	Traditional Signal Processing Approaches	23
2.5.3	Machine Learning and Data-Driven Models	24
2.5.4	Integration with IIoT Architectures	25
2.6	Research Gaps and Synthesis	25
3	The Data Concentrator	29
3.1	Introduction	29
3.2	The Production Line	31
3.2.1	AROL Capping Machines	31
3.3	Project Motivation and Requirements Elicitation	32
3.4	Candidate Contribution	33
3.4.1	Sensor Nodes	33
3.5	System Requirements	35

3.5.1	Functional Requirements	35
3.5.2	Non-Functional Requirements	37
3.6	High Level Architecture	39
3.6.1	Logical Architecture	39
3.6.2	Configuration (F20.1)	41
3.6.3	Operational Scenarios	44
3.6.4	Physical Architecture	49
3.7	Hardware Selection	53
3.7.1	Hardware Requirements	53
3.7.2	Candidate Hardware Platforms	54
3.7.3	Final Selection Justification	59
3.8	Software Architecture	60
3.8.1	Overview	60
3.8.2	Key Architectural Components	61
3.8.3	Data-Centric Value of the Architecture	62
3.8.4	Illustration of Containerized Customization	63
3.8.5	Summary	64
4	Predictive Maintenance through Slow Signals	66
4.1	Long Short-Term Memory Networks for Predictive Maintenance . .	67
4.2	Dataset and Preprocessing	69
4.3	Design of the LSTM model	72
4.4	Training and Validation Strategy	73
4.5	Results and Comparative Analysis	74
4.5.1	Laboratory Results	75
4.5.2	Real-World Industrial Results	77
4.5.3	Comparison with Existing Approaches	78

4.6	Remarks on the results	79
4.7	Limitations and Scope of the Approach	80
5	Predictive Maintenance through Fast Signals	83
5.1	Related Work on Vibration-Based Anomaly Detection	84
5.2	Vibration Analysis for Predictive Maintenance	85
5.3	Dataset and Preprocessing	87
5.4	Laboratory Experiment	88
5.4.1	Data Acquisition	88
5.4.2	Preprocessing and Calibration	89
5.4.3	Design of the Vibration Container and Models	90
5.4.4	Diagnostic Methods	90
5.4.5	Method Combinations	95
5.4.6	Training and Validation Strategy	96
5.4.7	Deployment and Resource Profiling	98
5.4.8	Results	100
5.5	Concluding Remarks	115
6	Conclusions	117
6.1	Key Findings and Contributions	117
6.1.1	From Raw Data to Actionable Features	117
6.1.2	Decoupled and Customizable Intelligence	118
6.1.3	Hybrid Predictive Maintenance Paradigm	118
6.2	Industrial and Scientific Impact	119
6.3	Limitations and Future Work	119
6.4	Final Remarks	120
	References	121

List of Figures

2.1	Vertical and Horizontal Integration	11
2.2	Cloud Architecture	14
2.3	Edge Architecture	15
2.4	Fog Architecture	17
3.1	AROL Capping Machine	31
3.2	Nicla Sense ME	33
3.3	The Sensor Node (NS2)	34
3.4	Operational Entities	40
3.5	High-level Functions	40
3.6	Logical Functions	41
3.7	Logical Architecture	42
3.8	Interfaces between IDS-NS2 and the Data Concentrator	44
3.9	Data exchange architecture	44
3.10	Configuration of the Data Concentrator, part 1	45
3.11	Configuration of the Data Concentrator, part 2	45
3.12	Successful connection, part 1	46
3.13	Successful connection, part 2	47
3.15	Anomaly detected scenario, part 2	47
3.14	Anomaly detected scenario, part 1	48

3.17 No connection scenario, part 2	49
3.16 No connection scenario, part 1	50
3.18 Hardware physical architecture	51
3.19 Hardware interface scenario	51
3.20 Software physical architecture	52
3.21 Interface architecture of physical components	53
3.22 HP EliteBook Laptop used for prototyping	54
3.23 Raspberry Pi 4 Model B	55
3.24 Arduino Portenta X8	56
3.25 EdgeBox RPI-200	57
3.26 Software Architecture of the Data Concentrator	61
3.27 Containerized Microservices for Custom Feature Generation	64
4.1 General structure of an LSTM unit.	68
4.2 Fog Computing layers	69
4.3 Flowchart of the experiment	70
4.4 Flowchart of the code	72
4.5 MSE Results from Laboratory setting	76
4.6 Image of the sensor NODE_13 mounted on the machine, head 13	77
4.7 MSE Results from Industrial setting	78
4.8 Temperature prediction results from the Industrial setting	78
5.1 Capping Machine sensorized with ISM330DHCX IMU	88
5.2 Slightly worn bush	89
5.3 Flowchart of the experiment	99
5.4 First experiment, drift anomaly, band anomalies	100
5.5 First experiment, drift anomaly, point anomalies	101
5.6 First experiment, drift anomaly, confusion matrix	102

5.7	Second experiment, drift anomaly, band anomalies	103
5.8	Second experiment, drift anomaly, point anomalies	103
5.9	Second experiment, drift anomaly, confusion matrix	104
5.10	First experiment, transient anomaly, band anomalies	105
5.11	First experiment, transient anomaly, point anomalies	105
5.12	First experiment, transient anomaly, confusion matrix	107
5.13	Second experiment, transient anomaly, band anomalies	108
5.14	Second experiment, transient anomaly, point anomalies	108
5.15	Second experiment, transient anomaly, confusion matrix	109
5.16	First experiment, mixed anomaly, band anomalies	110
5.17	First experiment, mixed anomaly, point anomalies	110
5.18	First experiment, mixed anomaly, confusion matrix	112
5.19	Second experiment, mixed anomaly, band anomalies	113
5.20	Second experiment, mixed anomaly, point anomalies	113
5.21	Second experiment, mixed anomaly, confusion matrix	114

List of Tables

1.1	Key Contributions of the Dissertation	6
2.1	Comparative overview of cloud, edge, and fog computing in IIoT . .	18
2.2	Positioning of the proposed work compared to State-of-the-Art Edge/Fog Platforms	21
3.1	Comparison of Hardware Platforms for the Data Concentrator . . .	58
4.1	LSTM experiment resource consumption.	75
4.2	Comparison of predictive performance with existing approaches. . .	79
5.1	Methods and Data Requirements	86
5.2	Architecture and Deployment Characteristics	86
5.3	Experimental test conditions and acquisition parameters.	89
5.4	Summary of vibration analysis diagnostic methods	94
5.5	Suitability of diagnostic methods for different anomaly types	95
5.6	Anomaly Detection Timing and Performance Metrics — First Ex- periment	101
5.7	Anomaly Detection Timing and Performance Metrics — Second Experiment	104
5.8	Anomaly Detection Timing and Performance Metrics — First Tran- sient Experiment	106

5.9	Anomaly Detection Timing and Performance Metrics — Second Transient Experiment	109
5.10	Anomaly Detection Timing and Performance Metrics — Mixed Anomalies, First Experiment	111
5.11	Anomaly Detection Timing and Performance Metrics — Mixed Anomalies, Second Experiment	114

Nomenclature

Acronyms / Abbreviations

<i>1D</i>	One-Dimensional
<i>5G</i>	Fifth Generation Mobile Network
<i>ADF</i>	Augmented Dickey–Fuller
<i>AE</i>	Autoencoder
<i>AGV</i>	Automated Guided Vehicle
<i>AI</i>	Artificial Intelligence
<i>API</i>	Application Programming Interface
<i>ARIMA</i>	AutoRegressive Integrated Moving Average
<i>AWS</i>	Amazon Web Services
<i>BLE</i>	Bluetooth Low Energy
<i>BPFI</i>	Ball Pass Frequency Inner race
<i>BPFO</i>	Ball Pass Frequency Outer race
<i>BPTT</i>	Backpropagation Through Time
<i>CAE</i>	Convolutional Autoencoder
<i>CAN</i>	Controller Area Network
<i>CFCN</i>	Fog Processor Node

<i>CNN</i>	Convolutional Neural Network
<i>CoAP</i>	Constrained Application Protocol
<i>Cortex-A53</i>	ARM Cortex-A53 Processor
<i>Cortex-M4</i>	ARM Cortex-M4 Processor
<i>CPS</i>	Cyber-Physical Systems
<i>CPU</i>	Central Processing Unit
<i>CRI – O</i>	Container Runtime Interface for OCI
<i>CWRU</i>	Case Western Reserve University
<i>DC</i>	Direct Current
<i>DC-MLP</i>	Deep Cascade Multi-Layer Perceptron
<i>DL</i>	Deep Learning
<i>DTW</i>	Dynamic Time Warping
<i>DWT</i>	Discrete Wavelet Transform
<i>EMC</i>	Electromagnetic Compatibility
<i>EMI</i>	Electromagnetic Interference
<i>eMMC</i>	Embedded MultiMediaCard
<i>ERP</i>	Enterprise Resource Planning
<i>F1</i>	F1-Score (Harmonic Mean of Precision and Recall)
<i>FAQMP</i>	Forecasting Air Quality with Multi-Processing
<i>FDPR</i>	Fault Detection and Prognosis using Regression
<i>FFT</i>	Fast Fourier Transform
<i>FPGA</i>	Field-Programmable Gate Array
<i>FS</i>	Fog Server

<i>GB</i>	Gigabyte
<i>GPIO</i>	General Purpose Input/Output
<i>GPU</i>	Graphics Processing Unit
<i>GRU</i>	Gated Recurrent Unit
<i>HDD</i>	Hard Disk Drive
<i>HMI</i>	Human–Machine Interface
<i>HTTP</i>	HyperText Transfer Protocol
<i>I²C</i>	Inter-Integrated Circuit
<i>IDS-NS2</i>	Intelligent Distributed Sensor Node – NS2
<i>IEA</i>	International Energy Agency
<i>IF</i>	Isolation Forest
<i>IIoT</i>	Industrial Internet of Things
<i>IMU</i>	Inertial Measurement Unit
<i>IoT</i>	Internet of Things
<i>IoT-Edge</i>	Internet of Things Edge Computing
<i>IT</i>	Information Technology
<i>k-NN</i>	k-Nearest Neighbors
<i>KDE</i>	Kernel Density Estimation
<i>KNN</i>	K-Nearest Neighbor (classifier)
<i>LAN</i>	Local Area Network
<i>LoRa</i>	Long Range Radio
<i>LoRaWAN</i>	Long Range Wide Area Network
<i>LSTM</i>	Long Short-Term Memory

LTS Long-Term Support

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MES Manufacturing Execution System

ML Machine Learning

Modbus/TCP Modbus Transmission Control Protocol

MQTT Message Queuing Telemetry Transport

MSE Mean Squared Error

NFV Network Function Virtualization

NS2 Sensor Node 2 (experimental platform)

NSA Non-Stationary Analysis (contesto: dataset reale non stazionario)

NXP Next eXPerience (Semiconductors)

OEM Original Equipment Manufacturer

OPC-UA Open Platform Communications – Unified Architecture

OT Operational Technology

OTA Over-The-Air

PAN Personal Area Network

PC Personal Computer

PCIe Peripheral Component Interconnect Express

PCs Personal Computers

PdM Predictive Maintenance

PLC Programmable Logic Controller

PoE Power over Ethernet

PROFINET Process Field Network

PSD Power Spectral Density

PSO Particle Swarm Optimization

R^2 Coefficient of Determination

RAM Random Access Memory

ReLU Rectified Linear Unit

REST Representational State Transfer

RF Random Forest

RMS Root Mean Square

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

RUL Remaining Useful Life

SAMD21 Microchip SAM D21 Microcontroller

SCADA Supervisory Control And Data Acquisition

SDN Software-Defined Networking

Seq2Seq Sequence-to-Sequence

SoC System on Chip

SoM System on Module

SPI Serial Peripheral Interface

SSD Solid State Drive

STFT Short-Time Fourier Transform

SVDD Support Vector Data Description

SVM Support Vector Machine

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol / Internet Protocol

TFLite TensorFlow Lite

TinyML Tiny Machine Learning

UART Universal Asynchronous Receiver–Transmitter

UWB Ultra Wide Band

WAN Wide Area Network

Wi-Fi Wireless Fidelity

Chapter 1

Introduction

1.1 The Industrial Internet of Things (IIoT): A Paradigm Shift in Manufacturing

While the general IIoT landscape encompasses a wide range of technologies and applications, this dissertation focuses specifically on the design of a data-centric edge framework for predictive maintenance. The objective is not to introduce new communication protocols or specialized hardware platforms, but to enable the practical transformation of heterogeneous industrial data into actionable knowledge through modular analytics deployed close to the data source.

1.1.1 The Digital Transformation of Industry

The Industrial Internet of Things (IIoT) is fundamentally reshaping modern manufacturing by enabling pervasive connectivity among machines, sensors, and enterprise information systems [1]. Unlike consumer-oriented IoT scenarios, industrial environments impose stringent requirements in terms of latency, determinism, reliability, and security [2]. The transformation enabled by IIoT is therefore not limited to the digitization of existing processes, but extends to the creation of fully data-driven ecosystems in which value is generated primarily from the intelligent exploitation of data streams.

At the core of this evolution are cyber-physical systems (CPS), where computational and communication capabilities are tightly integrated with physical assets [3]. CPS enable real-time monitoring, feedback control loops, and adaptive manufacturing strategies. In this context, data becomes the central resource for decision-making: it is collected from distributed sensors, preprocessed near the source, transmitted through resilient networks, and ultimately transformed into actionable insights. Without robust and reliable data pipelines, however, IIoT infrastructures remain fragmented and unable to deliver their full potential [4].

The long-term objective of digital transformation in industry is the transition from reactive practices to predictive, prescriptive, and adaptive operational models [5]. Achieving this vision requires effective data integration strategies. Vertical integration ensures the alignment of information across hierarchical layers, from shop-floor controllers to enterprise resource planning systems, while horizontal integration supports interoperability among partners in global supply chains [6]. These forms of data-driven collaboration enable flexible production, just-in-time maintenance, and rapid reconfiguration of processes in response to evolving market demands [7].

1.1.2 Evolution of Maintenance Strategies

Industrial maintenance represents a particularly clear example of how data-centric innovation can generate concrete economic value [8]. Maintenance strategies have gradually evolved from purely reactive approaches, in which repairs are performed only after failures occur, to preventive models based on scheduled interventions. More recently, predictive maintenance (PdM) has emerged as a data-driven alternative capable of anticipating equipment degradation before faults manifest [9].

The economic impact of unplanned downtime is substantial, with global losses estimated in the order of tens of billions of dollars per year across multiple industrial sectors [10]. Predictive maintenance frameworks address this issue by enabling early detection of anomalies, improved asset utilization, and reduction of waste and unnecessary interventions [11]. The key enabler of this transformation is not merely the presence of sensors, but the ability to convert raw signals into meaningful and reliable knowledge. Vibration measurements can reveal early mechanical faults,

temperature trends can indicate progressive wear, and electrical signatures can expose hidden operational stresses.

Early generations of PdM systems relied primarily on classical signal processing techniques such as Fast Fourier Transform (FFT), envelope detection, and spectral analysis to identify basic fault patterns [12]. Contemporary industrial environments, however, generate increasingly complex and heterogeneous data characterized by nonlinear interactions. As a result, modern PdM solutions combine physics-based models with machine learning (ML) and deep learning (DL) techniques [13], enabling the discovery of latent patterns, long-term temporal dependencies, and rare but critical events.

The maintenance value chain exemplifies the central thesis of this dissertation: competitive advantage in industrial environments arises from the effective exploitation of data. Sensors, networks, and computing platforms are important only insofar as they support richer data, better features, and more reliable predictive models.

1.1.3 Fundamental Computing Architectures: Cloud, Edge, and Fog

The capacity to exploit industrial data is strongly influenced by the underlying computing infrastructure. Cloud computing initially emerged as the dominant paradigm, offering virtually unlimited resources for storage and large-scale analytics [14]. Nevertheless, the physical separation between data sources and centralized cloud servers introduces latency that is often incompatible with real-time industrial requirements. Moreover, the continuous transmission of raw sensor streams is bandwidth-intensive and raises concerns regarding data ownership and security [15].

To address these limitations, decentralized paradigms such as edge and fog computing have been introduced. Edge computing brings computation closer to the data source, reducing latency and enabling immediate local responses. Fog computing complements this approach by creating an intermediate layer between edge devices and the cloud, allowing sensitive or time-critical processing to occur locally while delegating computationally intensive tasks to centralized infrastructures [16]. From a data-centric perspective, these paradigms are primarily concerned with determining where in the processing pipeline data should be aggregated, enriched, and interpreted.

The future of IIoT will therefore be shaped not by a strict choice between cloud or edge, but by the ability to orchestrate hybrid ecosystems that optimize data flows. Different tasks require different trade-offs between responsiveness, scalability, and global visibility [17]. Effective orchestration ensures that data is transformed into useful information at the appropriate location and at the appropriate time.

1.2 Problem Definition and Research Gaps

1.2.1 Limitations of Current Data-Centric Infrastructures

Despite rapid technological advances, contemporary industrial infrastructures remain highly fragmented. Proprietary platforms dominate the market, creating isolated silos in which data is locked within vendor-specific ecosystems [18]. This lack of interoperability limits the possibility of cross-domain analytics and significantly slows the adoption of predictive and prescriptive applications.

Academic research has proposed numerous flexible and lightweight solutions, but these are often validated only on small-scale laboratory testbeds [19]. Real industrial environments require robustness against noise, harsh operating conditions, and high data throughput, constraints that many experimental platforms fail to address adequately [20]. Consequently, a clear gap exists between closed but reliable commercial solutions and innovative yet fragile academic prototypes.

The fundamental challenge is therefore not the creation of new devices, but the design of scalable and interoperable data-centric frameworks capable of unlocking the full value of IIoT. This requires a holistic view of the data lifecycle, encompassing acquisition from heterogeneous sources, preprocessing, feature generation, secure communication, large-scale analytics, and integration into operational decision-making processes.

1.2.2 From Data Collection to Data Value

Transforming raw industrial signals into valuable insights involves a structured multi-stage process. Data must first be acquired from diverse physical assets and legacy systems, then cleaned and aligned to ensure consistency. Meaningful features

must be engineered to capture degradation patterns and operational anomalies, after which machine learning models can be applied to generate predictions or diagnostics. Finally, the resulting insights must be integrated into maintenance workflows and business processes.

When any stage of this pipeline is weak or poorly implemented, data remains merely a cost factor, consuming storage and bandwidth without producing tangible benefits. Conversely, an effective pipeline enables new service-oriented business models in which manufacturers deliver reliability and uptime rather than simply physical products.

1.3 Original Contributions of the Dissertation

While IIoT, edge computing, and predictive maintenance are mature research areas, their practical integration into a coherent and industrially deployable framework remains a significant challenge. The originality of this dissertation lies primarily at the architectural and system level, focusing on a data-centric methodology that decouples analytics from original equipment manufacturer (OEM) constraints. In this context, it is crucial to clarify that the Data Concentrator presented in this work is not the sole object of study, but rather a specific functional instance of the proposed framework—a physical validation of a broader architectural paradigm.

The primary contribution of this research is the design and implementation of an abstraction layer capable of aggregating heterogeneous industrial sources and exposing them through standardized interfaces. This is achieved through a containerized analytics framework that supports customizable feature extraction pipelines deployable directly at the edge, allowing third-party developers to personalize data transformations. Furthermore, this work proposes a unified methodological approach for combining multi-scale data—integrating slow-varying signals, such as temperature, with fast-varying signals, such as vibration—within a single predictive maintenance ecosystem. By merging classical signal processing with machine learning and deep learning models into an industrial-grade workflow, the entire framework has been validated in a real production environment on AROL capping machinery. This validation demonstrates not only the feasibility and robustness of the system but also the practical utility of treating raw industrial data as a flexible, open, and collaborative asset rather than a static collection of points.

It is important to emphasize that the machine learning models adopted—such as LSTM networks, autoencoders, and isolation forests—are not claimed as novel in their mathematical formulation. Instead, the originality resides in their systematic integration and deployment within this coherent, data-centric architecture, proving that the value of modern IIoT lies in the ability to transform and personalize data streams at the edge.

Table 1.1 Key Contributions of the Dissertation

Contribution	Description	Significance
Data-centric architecture	Modular framework for IIoT streams and containerized analytics.	Interoperability, scalability, and vendor-agnostic integration.
Feature generation platform	Custom, user-defined feature extraction from raw signals.	Transforms raw data into actionable insights for PdM.
Hybrid PdM framework	Integration of slow/fast signal analysis (Classical + ML/DL).	Improved robustness in capturing gradual and sudden failures.
Industrial validation	Real-world demonstration with energy-intensive machinery.	Proves feasibility and industrial-grade applicability.

1.4 Application to Predictive Maintenance

The proposed data-centric approach is demonstrated through its application to predictive maintenance. Two complementary classes of signals are considered. Slow-varying signals, such as temperature, reflect gradual wear and long-term degradation. These signals are characterized by low sampling rates and are analyzed using recurrent neural networks, particularly Long Short-Term Memory (LSTM) architectures, which are well suited for modeling temporal dependencies. Fast-varying signals, such as vibrations, capture high-frequency phenomena associated with sudden anomalies or incipient mechanical faults. These signals require hybrid processing pipelines that combine traditional signal processing techniques with machine learning methods, including autoencoders and isolation forests.

The integration of insights from both signal categories enables a holistic assessment of machine health, capturing both subtle long-term trends and abrupt failures.

The effectiveness of this approach derives primarily from data quality and feature generation rather than from any single technological component.

1.4.1 Justification for Signal Selection

The distinction between slow- and fast-varying signals is grounded in the physical characteristics of industrial equipment and in practical considerations related to data acquisition. Slow signals, such as temperature or energy consumption, typically represent gradual degradation processes and can be effectively analyzed through forecasting models. Fast signals, particularly vibration measurements, contain high-frequency information directly linked to mechanical defects and therefore require specialized processing techniques and higher sampling rates.

In the industrial scenario addressed in this dissertation, temperature and vibration were selected because they are directly correlated with common failure modes, can be acquired non-invasively, provide complementary information, and are widely recognized in the predictive maintenance literature. Other types of signals, including electrical current or torque, could also be incorporated into the framework, but their inclusion depends mainly on sensor availability and operational constraints rather than on architectural limitations.

1.5 Industrial Impact

The framework proposed in this dissertation has been conceived with practical deployment requirements in mind. Its adoption enables earlier detection of anomalies, reduction of unplanned downtime, improved utilization of existing machinery, and simplified integration of third-party analytics services. Moreover, the creation of reusable datasets and features facilitates the development of additional data-driven applications.

Although a detailed economic assessment is beyond the scope of this work, the real-world deployment described in later chapters demonstrates that the framework can be integrated into existing production environments with minimal disruption and tangible operational benefits.

1.6 Scope and Limitations

This dissertation focuses on the data-centric pipeline and the technical feasibility of edge analytics. Consequently, the development of novel machine learning algorithms, formal real-time deterministic scheduling, and prescriptive models for economic maintenance optimization are outside its scope. Similarly, while the framework is designed for multi-node scalability, the research does not aim to introduce new cloud-to-edge orchestration protocols.

1.7 Structure of the Thesis

The remainder of the dissertation is organized as follows. Chapter 2 reviews relevant literature on IIoT technologies and predictive maintenance. Chapter 3 presents the design principles and implementation of the proposed data-centric framework. Chapter 4 describes predictive maintenance approaches based on slow-varying signals, while Chapter 5 focuses on fast-varying signal analysis. Chapter 6 summarizes the contributions, discusses industrial implications, and outlines future research directions.

Chapter 2

Related Works and Background

A deep understanding of IIoT technologies, architectures, and applications is essential to grasp how industrial data can be effectively collected, processed, and transformed into value. The ability to leverage data intelligently is what enables new services and business models, while the enabling infrastructures remain functional components [21]. This chapter surveys the state of the art in IIoT with particular attention to three domains where data plays a decisive role:

- Computing architectures that allow distributed, low-latency data analytics, including cloud, edge, and fog computing;
- Fog and edge solutions tailored for industrial contexts, with a focus on embedded, containerized, and extensible approaches that facilitate data collection and processing;
- Predictive maintenance systems, where data-driven strategies—ranging from classical models to machine learning—demonstrate the potential of exploiting operational information.

By reviewing these domains, the chapter aims to highlight the current research gaps in data acquisition and exploitation within IIoT systems, thereby providing the foundation for the subsequent proposal of a fog-based infrastructure used in this thesis. Such an infrastructure should be understood not as an end in itself, but as an enabling mechanism for extracting, managing, and monetizing industrial data.

The remainder of the chapter is structured as follows. Section 2.1 outlines the main concepts of IIoT, its enabling technologies, benefits, and challenges, with particular emphasis on the role of data. Section 2.2 compares cloud, edge, and fog paradigms in supporting efficient data processing. Section 2.3 discusses fog and edge implementations in both research and industry, highlighting their effectiveness in managing data flows. Section 2.4 reviews predictive maintenance approaches, illustrating how data analytics drives value creation. Section 2.5 synthesizes the main research gaps and identifies opportunities for improving the way industrial data is leveraged. Finally, Section 2.6 concludes with a summary that frames the fog infrastructure adopted in Chapter 3 as a functional enabler within a broader data-centric vision.

2.1 Industrial Internet of Things (IIoT)

The Industrial Internet of Things (IIoT) represents a paradigm shift in industrial operations, where the central element is no longer the machinery itself but the data continuously produced by connected assets [22]. By transforming raw sensor readings into actionable insights, IIoT enables a new generation of smart factories where processes are guided by information flows rather than static control rules. In this context, the value of IIoT lies in its ability to support real-time monitoring, predictive analytics, and data-driven decision-making, ultimately leading to improved efficiency, flexibility, and reliability [23].

Unlike the consumer-oriented Internet of Things (IoT), which emphasizes convenience and user experience through wearable devices, smart home systems, or personal electronics, IIoT is deeply rooted in industrial contexts where downtime translates into significant financial and operational risks. This difference elevates the importance of deterministic communication, secure data handling, and system scalability, as industrial data must be collected, transported, and processed under strict reliability and latency constraints [24].

At the heart of IIoT are cyber-physical systems (CPS), which unify physical assets with digital representations, allowing the continuous flow of data between machines and computational intelligence. CPS function as engines of data transformation: signals from heterogeneous sensors are collected, preprocessed, and translated into higher-level indicators that can support both autonomous decision-

making and human-in-the-loop control. This shift positions data as the fundamental raw material of industrial innovation [25].

IIoT extends the scope of traditional automation by incorporating advanced analytics, machine learning, and AI-driven optimization. These technologies enable the evolution from reactive to predictive and prescriptive maintenance strategies, leveraging data to anticipate failures before they occur and to recommend optimal interventions [26]. By closing the loop between data acquisition, feature generation, and predictive modeling, IIoT maximizes equipment availability, reduces waste, and improves production quality [27]. The data-centric perspective also broadens integration: vertical integration ensures that sensor-level information propagates upward to inform enterprise-level planning, while horizontal integration allows data to be shared across supply chains, facilitating collaborative and adaptive production ecosystems [28]. This dual integration, summarized in Fig. 2.1, underscores the role of data as a unifying medium for industrial collaboration.

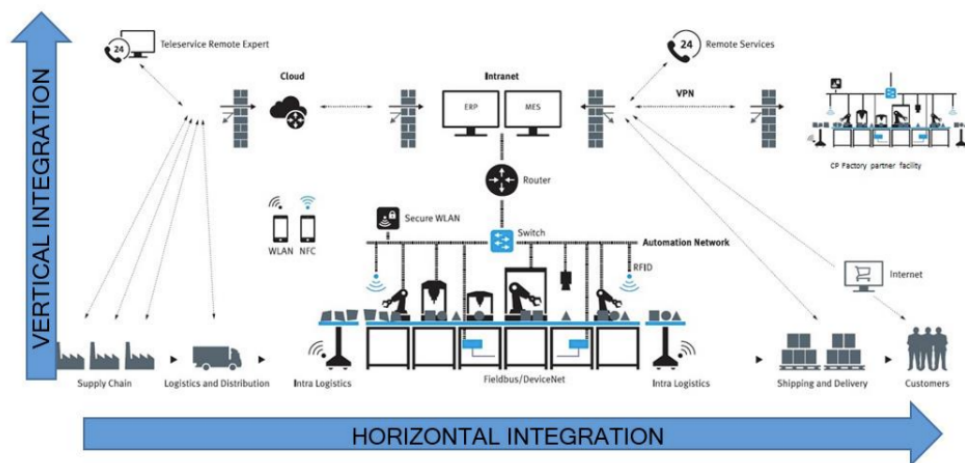


Fig. 2.1 Vertical and Horizontal Integration [28]

The technological foundations of IIoT—smart sensors, communication protocols, and interoperable data models—are best understood as stages in the data lifecycle. Smart sensors perform the first transformation step, converting physical phenomena such as vibration, temperature, or pressure into digital signals. Many modern sensors already incorporate local intelligence, enabling on-device filtering, anomaly detection, or feature extraction before transmission [29]. Communication protocols such

as OPC-UA, MQTT, Modbus, and PROFINET provide the channels through which data flows, ensuring that information retains integrity and interoperability as it moves across heterogeneous environments. At a higher level, ontologies and standardized data models ensure semantic consistency, making it possible to aggregate and analyze information from diverse sources in a unified framework [30]. These layers collectively form a pipeline in which data becomes progressively more structured, meaningful, and valuable.

The benefits of adopting IIoT are therefore inseparable from the value of the data it produces. Continuous monitoring enables the extraction of temporal patterns that reduce unplanned downtime and optimize maintenance scheduling [31]. Adaptive manufacturing becomes possible when production data can be fed back into planning processes in near real-time, allowing mass customization without sacrificing throughput or quality [32]. At the economic level, the capacity to monetize data—through performance-based contracts, predictive maintenance services, or digital twins—opens entirely new business models that extend beyond the physical product.

Yet the transition to data-centric operations is not without challenges. Legacy equipment often generates data in proprietary or non-standard formats, requiring costly retrofitting or integration [33]. Ensuring the security and sovereignty of industrial data is a persistent concern, especially when it leaves the boundaries of the factory floor. Equally, the human workforce must adapt to new roles where data literacy, analytics, and cybersecurity are essential skills [34]. Finally, the massive scale of IIoT-generated data imposes unprecedented demands on computing infrastructures, highlighting the importance of distributed paradigms such as cloud, edge, and fog computing.

Within the broader context of Industry 4.0, IIoT emerges as a data-centric infrastructure for digital manufacturing. By transforming raw signals into structured knowledge and actionable predictions, IIoT acts as the foundation of smart, connected, and adaptive industrial systems. Addressing interoperability, security, and computational scalability is therefore crucial to realize the full promise of Industry 4.0 and to support the next generation of intelligent, data-driven industrial applications [35].

In summary, IIoT is not merely a network of interconnected devices but a distributed system for the continuous generation, transformation, and exploitation of

data. This reorientation toward data as the core resource enables predictive maintenance, adaptive production, and innovative business models, while also posing technical and organizational challenges that shape the ongoing evolution of industrial digitalization.

2.2 Computing Architectures in IIoT

The effectiveness of IIoT systems depends not only on the ability to acquire data, but also on the ability to process, store, and transform it into actionable knowledge. Computing architectures define where along the network this processing occurs, and thereby shape the latency, scalability, and value of the resulting data [36]. From a data-centric perspective, the choice of architecture is not a matter of hardware deployment alone, but a strategic decision about how information flows are governed, how quickly they can be transformed into insights, and how securely they can be shared across industrial ecosystems [37].

Traditionally, cloud computing has dominated IIoT data management. Cloud platforms provide virtually unlimited storage and computational resources, enabling large-scale analytics and enterprise-level integration. However, the cloud model also centralizes data processing, introducing latency, bandwidth consumption, and privacy risks that are often incompatible with time-critical industrial processes [38]. To address these limitations, new paradigms such as edge and fog computing have emerged. These architectures distribute computation closer to the data source, reducing the time between data generation and actionable decision-making, while also alleviating the pressure on communication networks [39].

In the following subsections, we examine cloud, edge, and fog computing from the perspective of data-centric IIoT. Each paradigm is assessed not only in terms of its computational properties but also in its ability to support the data lifecycle—from raw signal acquisition to feature generation, predictive modeling, and decision support.

2.2.1 Cloud-Centric Architectures

In a cloud-centric model, industrial data flows from machines and sensors to centralized data centers, where it is stored, aggregated, and analyzed at scale (Fig. 2.2).

Cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform offer elastic infrastructures that decouple industrial applications from physical hardware, enabling organizations to scale resources dynamically [40, 41].

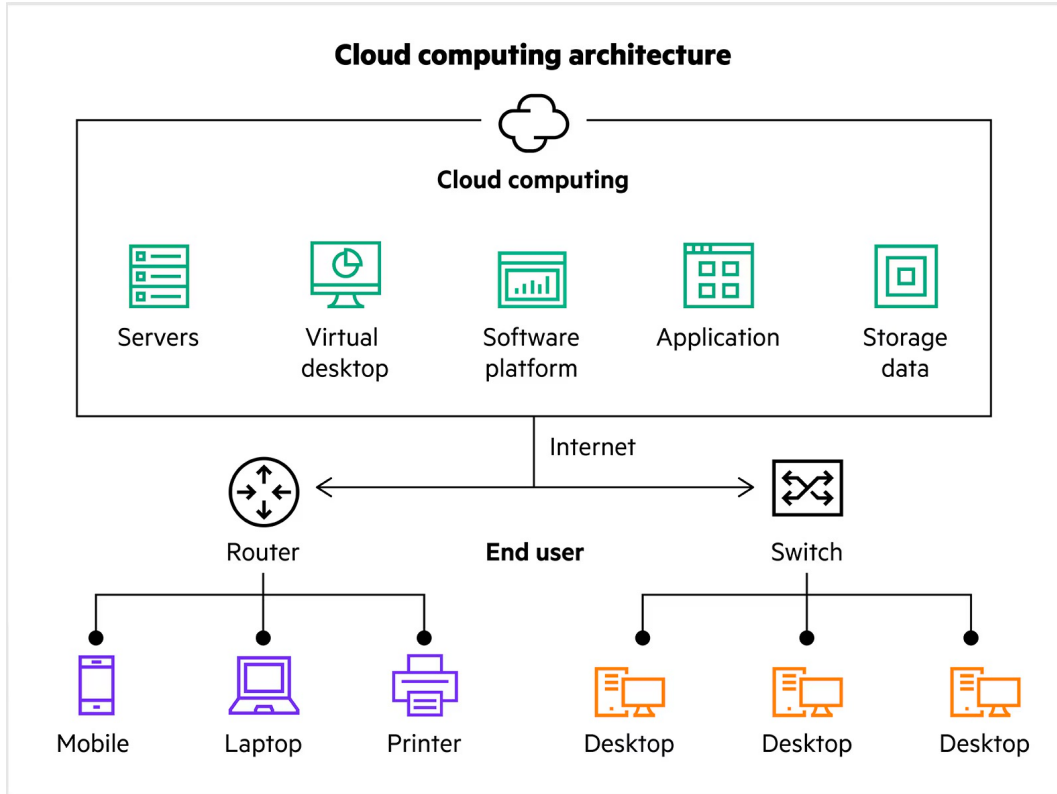


Fig. 2.2 Cloud Architecture

From a data-centric standpoint, the key advantage of the cloud lies in its capacity to support advanced analytics and long-term data exploitation. Machine learning pipelines, digital twins, and business intelligence platforms rely on vast, historical datasets that only cloud infrastructures can realistically maintain [42]. Moreover, cloud platforms enable cross-site integration: by consolidating data from multiple plants, fleets, or supply chains, they generate a global perspective that can uncover patterns invisible at the local scale [43].

However, the centralization of data processing introduces significant drawbacks. Transmitting raw high-frequency signals such as vibration, acoustic, or video streams to remote servers consumes substantial bandwidth and incurs non-negligible delays [44][45]. These latencies are incompatible with applications that demand real-time responsiveness, such as closed-loop process control or safety-critical monitoring

[46]. In addition, outsourcing data to third-party infrastructures raises concerns about security, sovereignty, and compliance with local regulations [47]. From a data governance perspective, the cloud excels at large-scale, long-term exploitation of information, but it struggles to meet the immediacy and locality requirements of operational analytics [48].

2.2.2 Edge Computing

Edge computing relocates processing power from centralized data centers to the periphery of the network, i.e., near the devices that generate data (Fig. 2.3). In this paradigm, embedded controllers, industrial PCs, or programmable logic controllers (PLCs) perform computations locally, often transforming raw signals into aggregated features or anomaly indicators before transmission [49].

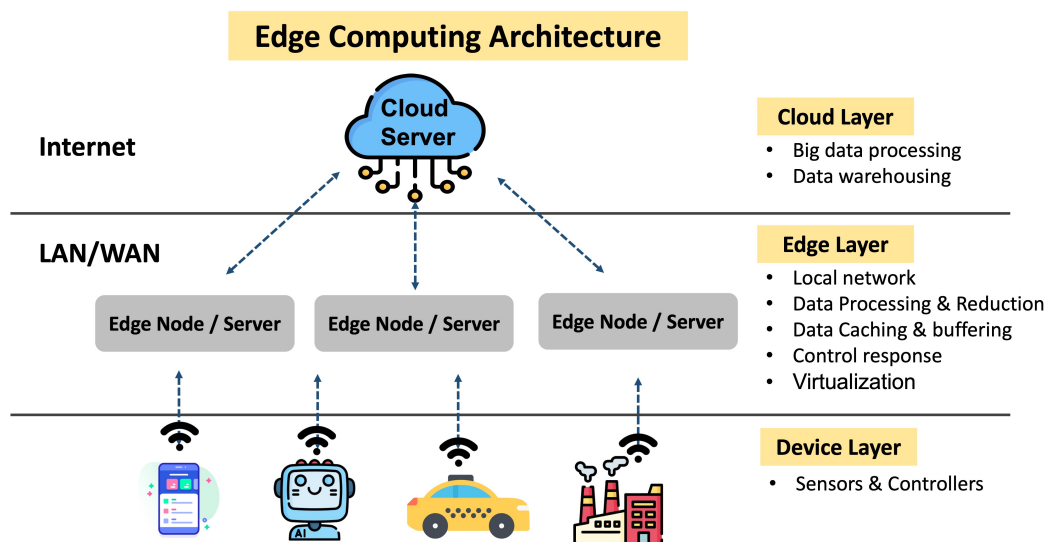


Figure : Edge computing architecture overview
Source : The research team

Fig. 2.3 Edge Architecture

This shift has profound implications for the data lifecycle. By extracting features directly at the edge, IIoT systems reduce the volume of data that must traverse the network, thereby lowering bandwidth consumption and operational costs [50]. More importantly, edge computing drastically reduces latency, enabling sub-second decision-making for tasks such as robotic motion control, adaptive process regulation,

or real-time safety monitoring [51, 52]. These capabilities make edge architectures particularly effective for predictive maintenance applications, where anomalies must be detected early enough to prevent cascading failures.

In addition to efficiency, edge computing enhances data protection. Sensitive operational information can remain within the industrial site, reducing exposure to third parties and mitigating risks of data exfiltration or regulatory non-compliance [53]. However, the benefits of locality are tempered by resource constraints: edge devices often lack the computational capacity to run complex machine learning models or to store large historical datasets [54–56]. As such, edge computing is best suited for tasks that benefit from immediacy, while still relying on higher layers for long-term analysis and cross-site optimization [57].

2.2.3 Fog Computing

Fog computing extends the benefits of edge computing by introducing an intermediate layer between local devices and remote clouds (Fig. 2.4). Fog nodes—such as gateways or micro-servers—aggregate and process data across multiple machines, production lines, or sites, enabling context-aware analytics without relying exclusively on centralized resources [58].

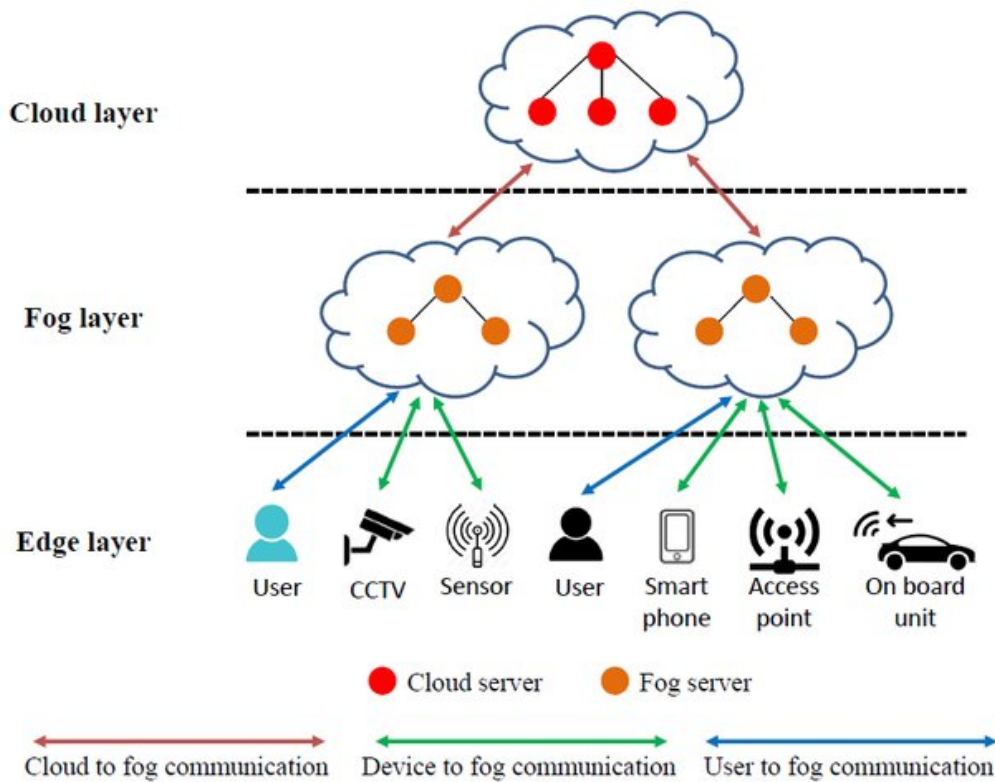


Fig. 2.4 Fog Architecture

From a data perspective, fog computing plays a crucial bridging role. It allows time-sensitive tasks such as anomaly detection or predictive feature extraction to occur close to the data source, while also preparing and aggregating datasets for higher-level cloud analytics [59]. By distributing intelligence hierarchically, fog computing reduces network congestion, supports scalable monitoring, and ensures that data is exploited at the most appropriate level of granularity [58].

Industrial implementations increasingly adopt fog computing to orchestrate fleets of robots, manage collaborative production systems, or coordinate predictive maintenance across distributed assets [60–62]. Platforms such as Cisco IOx and Eclipse Fog05 demonstrate how containerized applications can be deployed dynamically on fog nodes, allowing the data pipeline to remain flexible and extensible [63, 64]. The challenge, however, lies in the complexity of managing heterogeneous and distributed infrastructures. Standardization, orchestration, and security mechanisms remain active areas of research [65].

2.2.4 Comparative Overview

A comparative analysis of cloud, edge, and fog architectures highlights their complementary contributions to the data lifecycle (Table 2.1). Cloud excels at historical analysis, global integration, and advanced modeling. Edge ensures immediacy, locality, and privacy. Fog bridges the two, enabling near-real-time, context-aware analytics across distributed environments.

The interplay between these paradigms reflects a broader shift: computing architectures are no longer evaluated solely by their technical performance, but by their ability to transform industrial data into timely, reliable, and valuable knowledge.

Table 2.1 Comparative overview of cloud, edge, and fog computing in IIoT

Characteristic	Cloud Computing	Edge Computing	Fog Computing
Location	Remote data centers	On-device / local site	Intermediate layer
Latency	High	Very low	Low to moderate
Scalability	Virtually unlimited	Limited	Moderate
Data privacy	Lower	High	Moderate
Analytics capabilities	Advanced, large-scale	Real-time, local	Context-aware, distributed
Bandwidth usage	High	Very low	Moderate
Operational complexity	Low	High	Moderate to high
Best suited for	Historical analytics, Digital twins, Fleet optimization	Real-time control, Robotics, On-site anomaly detection	Distributed monitoring, Predictive maintenance, Cross-site coordination

2.3 Fog and Edge Implementations in Industrial Practice and Research

The transition from conceptual architectures to operational IIoT systems requires concrete implementations capable of handling real-world data. Fog and edge computing play a pivotal role in bridging the gap between raw data acquisition and

actionable intelligence, enabling industrial organizations to exploit the full potential of the data generated by their assets [66].

2.3.1 Industrial Implementations

In industrial practice, fog and edge nodes are typically deployed as gateways, micro-servers, or embedded controllers that aggregate and preprocess data from machines and sensors. Their primary function is to reduce the latency of data processing, minimize bandwidth consumption, and enforce data governance policies locally [67–69]. By transforming raw sensor streams into higher-level indicators, these devices ensure that only relevant information is forwarded to enterprise systems or cloud platforms for further analysis.

Modern industrial landscapes are characterized by robust, albeit often closed, edge platforms. Solutions such as Siemens Industrial Edge provide a secure ecosystem for deploying containerized applications directly on industrial controllers, focusing on high availability and seamless integration with automation hardware. Similarly, Microsoft Azure IoT Edge and AWS IoT Greengrass extend cloud capabilities to the field, offering managed orchestration, security, and digital twin synchronization [70, 71]. While these platforms excel in reliability, they often impose a degree of vendor lock-in and may provide limited flexibility for low-level, vendor-agnostic feature extraction pipelines that need to operate independently of a specific provider's cloud ecosystem.

Industrial deployments highlight several practical benefits of data-centric fog and edge computing:

- **Predictive Maintenance:** By locally processing vibration, temperature, and acoustic signals, edge nodes can detect early anomalies and generate predictive features, reducing downtime [72, 73].
- **Real-Time Control:** Manufacturing processes rely on local computation to implement closed-loop control, using data streams that require millisecond-level responsiveness [74, 75].
- **Data Standardization and Interoperability:** Edge and fog nodes can enforce consistent data formats (e.g., via OPC-UA), converting heterogeneous signals into standardized datasets [76, 77].

- **Security and Data Sovereignty:** Processing sensitive operational data on-site ensures that privacy and regulatory compliance requirements are met [78, 79].

2.3.2 Academic and Experimental Prototypes

Academic research often explores novel edge and fog solutions focused on the optimization and transformation of industrial data. Prototypes typically leverage platforms such as Raspberry Pis, NVIDIA Jetson devices, or custom micro-servers to validate concepts in laboratory or pilot-scale testbeds [80–82].

Beyond hardware selection, research efforts have focused on the orchestration of distributed resources. Frameworks like Eclipse Fog05 aim to provide a unified infrastructure for computing, storage, and networking across heterogeneous nodes, emphasizing the virtualization of resources [64]. Furthermore, recent literature has explored the challenges of real-time guarantees within containerized environments. Systems such as RT-Kube and research on mixed-criticality container scheduling attempt to introduce deterministic execution and worst-case execution time (WCET) modeling into orchestrators like Kubernetes [83]. However, these academic efforts often focus on the underlying computational theory—such as scheduling and resource isolation—rather than on the application-level challenges of multi-frequency industrial data fusion and modular feature engineering for predictive maintenance.

Key contributions of research prototypes include:

- **Flexible Data Pipelines:** Modular software architectures enable experimentation with different preprocessing and machine learning workflows [84, 85].
- **Interoperability Studies:** Academic prototypes often explore cross-vendor integration, bridging proprietary data formats to create unified, vendor-agnostic data streams [86–88].
- **Resource-Constrained Analytics:** Experimental systems demonstrate how computationally efficient algorithms can be deployed on resource-limited nodes [89–91].

Despite their innovative approaches, research prototypes face challenges that limit industrial applicability. Low-cost hardware may lack robustness in harsh

factory environments, and experimental deployments often do not scale beyond small testbeds [92].

2.3.3 Synthesis and Positioning of the Proposed Work

To better understand the positioning of the research presented in this dissertation, Table 2.2 compares the proposed framework with existing industrial platforms and academic research focused on real-time orchestration.

Table 2.2 Positioning of the proposed work compared to State-of-the-Art Edge/Fog Platforms

Feature	Commercial (e.g., Azure, Siemens)	Academic RT (e.g., RT-Kube, Fog05)	This Dissertation
Primary Focus	Reliability, security, and ecosystem integration.	Computational theory, scheduling, and determinism.	Data-centric analytics and PdM pipelines.
Flexibility	Limited by proprietary ecosystems and billing models.	High flexibility in resource orchestration.	High; vendor-agnostic and modular analytics.
Data Handling	Generic telemetry and cloud-to-edge sync.	Focus on resource efficiency and isolation.	Specialized Slow/Fast signal fusion for PdM.
Industrial Goal	Operations management and fleet monitoring.	Optimization of distributed computing resources.	Predictive maintenance and anomaly detection.

As shown in the comparison, while commercial systems provide robustness and academic prototypes explore formal real-time guarantees, a gap remains for a framework that combines industrial-grade modularity (via containerization) with a specific focus on the heterogeneous data requirements of predictive maintenance. The framework proposed in Chapter 3 addresses this gap by prioritizing the data lifecycle and feature engineering over new scheduling or communication theories.

2.4 Data-Centric Perspective on Implementations

Regardless of the deployment context—industrial or experimental—the ultimate goal of edge and fog implementations is the generation of valuable data. Devices act as nodes in a broader data ecosystem, where raw sensor readings are transformed into structured, analyzable information [93, 87, 84]. Key functions include:

1. **Data Acquisition:** Capturing high-frequency and heterogeneous signals from machines, sensors, and actuators [82, 88, 94].
2. **Preprocessing and Feature Extraction:** Filtering noise, normalizing signals, and computing features relevant for predictive analytics or process control [95, 96].
3. **Local Decision-Making:** Applying rule-based logic or lightweight machine learning models to generate immediate insights and trigger alerts [80, 81, 84].
4. **Data Aggregation and Transmission:** Summarizing or compressing data for downstream analytics in the cloud or across multiple production sites [87, 85, 97].

This data-centric view emphasizes that edge and fog nodes are not valuable as standalone hardware components. Their importance derives entirely from their ability to enable the continuous transformation of raw signals into actionable knowledge, supporting predictive maintenance, process optimization, and higher-level decision-making across the industrial enterprise [93, 84, 98].

2.5 Predictive Maintenance Techniques in IIoT

Predictive maintenance (PdM) represents a paradigm shift in industrial asset management, moving from reactive or scheduled interventions to data-driven strategies that anticipate failures and optimize operational decisions. From a data-centric perspective, PdM is fundamentally about transforming raw sensor streams into actionable knowledge that informs maintenance planning, reduces downtime, and extends equipment lifespan [99, 100].

2.5.1 Data Sources and Signal Types

Effective PdM relies on diverse data sources, each providing unique insights into equipment health:

- **Slow-varying signals:** Temperature, pressure, and lubricant characteristics evolve gradually, reflecting long-term degradation or wear [101]. These signals are essential for identifying progressive failures and scheduling routine interventions.
- **Fast-varying signals:** Vibration, acoustic emissions, and electrical current fluctuations capture transient events indicative of sudden anomalies or early-stage faults [94, 96]. High-frequency data enables the detection of subtle patterns that precede catastrophic failures.
- **Contextual and operational data:** Production parameters, environmental conditions, and operator actions provide context that enhances the interpretability and predictive power of sensor data [95, 102].

The integration of heterogeneous signals into a unified data model is a prerequisite for accurate predictive analytics. Edge and fog nodes often perform initial preprocessing and feature extraction to reduce data volume, highlight relevant patterns, and improve downstream model performance [97, 103].

2.5.2 Traditional Signal Processing Approaches

Historically, PdM relied on classical signal processing techniques to extract features from raw sensor data:

- **Frequency-domain analysis:** Fast Fourier Transform (FFT), Short-Time Fourier Transform (STFT), and spectral envelope analysis decompose vibration or acoustic signals into frequency components, revealing mechanical anomalies such as imbalance, misalignment, or bearing defects [96, 102].
- **Time-domain metrics:** Root Mean Square (RMS), peak-to-peak, skewness, and kurtosis quantify signal amplitude variations, enabling detection of abnormal operating conditions [94, 95].

- Wavelet-based methods: Discrete Wavelet Transform (DWT) and multiresolution analysis capture transient events and non-stationary patterns, improving fault localization and early warning capabilities [98, 101].

These techniques remain foundational in PdM pipelines, particularly for edge or fog implementations where computational resources are constrained. They convert high-frequency signals into structured features suitable for threshold-based alarms, statistical monitoring, or lightweight models [97, 103].

2.5.3 Machine Learning and Data-Driven Models

The rise of machine learning (ML) and artificial intelligence (AI) has expanded PdM capabilities, enabling the exploitation of complex, nonlinear relationships in large datasets:

- Supervised learning: Regression and classification models, such as Support Vector Machines (SVM), Random Forests (RF), and Gradient Boosted Trees, map extracted features to health indicators, failure probabilities, or Remaining Useful Life (RUL) estimates [99, 104, 105].
- Deep learning: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks process raw or minimally processed signals to automatically learn hierarchical features, capturing temporal dependencies and subtle anomalies [100, 106, 107].
- Unsupervised and anomaly detection: Autoencoders, Isolation Forests, and Support Vector Data Description (SVDD) identify deviations from normal operational patterns without requiring labeled failure data [95, 108, 109].

From a data-centric standpoint, these models illustrate the key principle of PdM: the value of sensor data increases as it is transformed into features and predictive indicators. High-quality data preprocessing, feature engineering, and integration of multiple signal sources directly influence model accuracy and the reliability of maintenance recommendations [97, 102, 103].

2.5.4 Integration with IIoT Architectures

PdM systems are closely tied to the underlying IIoT architecture. Edge and fog nodes play a dual role: they reduce latency by processing data locally for immediate anomaly detection, and they prepare aggregated datasets for cloud-based analytics, model retraining, or cross-site optimization [93, 87, 84]. By enabling hierarchical data processing, IIoT architectures allow predictive models to operate effectively across multiple scales:

- Local scale: Rapid detection of anomalies, feature extraction, and immediate alerts [82, 89].
- Site scale: Aggregation of equipment-level features, trend analysis, and contextual correlation across production lines [110, 88, 104].
- Enterprise scale: Global analytics, cross-plant benchmarking, and strategic maintenance planning [80, 81, 105].

This hierarchical processing ensures that data is exploited where it is most valuable, balancing immediacy, computational load, and insight generation [85, 106].

2.6 Research Gaps and Synthesis

Despite the rapid evolution of IIoT, edge and fog computing paradigms, and predictive maintenance methodologies, the full exploitation of industrial data remains constrained by structural and methodological limitations. The literature extensively addresses enabling technologies; however, fewer works provide an integrated, deployable, and experimentally validated framework capable of bridging heterogeneous data acquisition, edge-level processing, and advanced analytics within real industrial environments.

A first major limitation concerns data fragmentation. Industrial plants are typically composed of heterogeneous devices, proprietary communication protocols, and vendor-specific data formats, resulting in isolated data silos [86, 87]. Although middleware platforms partially mitigate interoperability issues, they often require substantial customization and remain tightly coupled to specific ecosystems [104, 108].

As a consequence, scalability across production lines and multi-vendor contexts is limited. What remains insufficiently addressed in the literature is the design of open, vendor-agnostic, and container-ready infrastructures that allow third parties to deploy custom data processing logic directly at the edge while preserving interoperability and data ownership.

A second research gap concerns feature extraction and data valorization. While predictive maintenance algorithms have received significant attention, considerably less focus has been devoted to the systematic transformation of raw industrial signals into high-quality, contextualized features. In practice, feature engineering remains a bottleneck, requiring domain expertise and ad hoc preprocessing pipelines [97, 102]. Existing contributions often emphasize model performance without fully addressing how features are generated, validated, and maintained in dynamic industrial environments. Furthermore, limited attention is given to the operational constraints of embedded platforms, where computational resources, memory, and latency requirements impose strict boundaries on deployable analytics solutions.

Closely related to this issue is the challenge of scalable and hierarchical analytics. Cloud-based architectures provide powerful computational capabilities but introduce latency, bandwidth consumption, and dependency on external infrastructures [104, 108]. Edge and fog paradigms mitigate these issues; however, the literature frequently presents conceptual architectures or small-scale prototypes, with limited discussion of orchestration mechanisms, containerization strategies, and real-time deployment of machine learning models on resource-constrained devices [91, 106]. In particular, empirical demonstrations of complex models—such as deep learning architectures—executed reliably within industrial edge hardware remain comparatively scarce.

Another critical aspect concerns the scope of predictive maintenance validation. Many studies evaluate algorithms in controlled laboratory conditions or on limited datasets, without fully assessing generalization across varying operating regimes. Moreover, single-variable approaches—such as temperature forecasting—are sometimes presented as end goals rather than as technological benchmarks. From a methodological standpoint, forecasting slowly varying signals can often be achieved through lightweight statistical models. The true research challenge lies not in the algorithmic novelty per se, but in demonstrating that an industrial-grade edge in-

infrastructure can host, orchestrate, and sustain advanced analytics pipelines under realistic operational constraints.

Within this context, the contribution of this thesis is positioned at the intersection of architecture and analytics. Rather than claiming the optimality of a specific predictive model, the work addresses the infrastructural gap that prevents industrial stakeholders from systematically deploying customized feature extraction and machine learning services at the edge. The Data Concentrator is conceived as a lightweight, data-centric, containerized fog node designed not only to execute analytics, but to preserve computational headroom for future personalization and third-party extensions.

In practical terms, this implies that advanced models must coexist with unused yet available resources, ensuring that additional services, transformations, or predictive pipelines can be deployed without architectural redesign. The system is therefore engineered to balance computational load, memory footprint, and latency, enabling:

- interoperability across heterogeneous sensor nodes and industrial equipment;
- deployment of user-defined preprocessing and feature engineering pipelines;
- execution of computationally demanding machine learning models on embedded hardware while maintaining resource margins;
- scalable multi-node configurations with controlled latency and predictable resource consumption.

The adoption of an LSTM-based forecasting pipeline in the subsequent chapter must therefore be interpreted as a technological validation exercise. While temperature prediction alone could be achieved with simpler statistical techniques, running a sequential neural architecture at one-second resolution within a containerized edge environment demonstrates that complex models can be supported without exhausting system resources. The relevance of the experiment lies not only in predictive accuracy, but in showing that such models can operate within a deliberately lightweight infrastructure designed to remain extensible.

It is also important to delineate the scope of the presented work. The predictive maintenance experiments focus on selected signal categories and operating conditions. Temperature signals represent only indirect indicators of machine health and

should be integrated with additional modalities—such as vibration, torque, or operational parameters—to build comprehensive health indices. Although both laboratory and industrial validations were conducted, broader multi-plant validation would be necessary to fully generalize the findings.

In summary, the literature highlights that, beyond algorithmic development, significant challenges remain in integrating heterogeneous industrial data sources and deploying analytics in a scalable and interoperable manner. This thesis addresses these challenges by experimentally validating a lightweight, containerized edge architecture within a real industrial context. The resulting framework provides a practical basis for extensible predictive maintenance and industrial analytics, where personalization and resource-aware deployment are central design principles.

Chapter 3

The Data Concentrator

3.1 Introduction

The transition toward data-driven industrial ecosystems requires more than pervasive sensing and reliable communication infrastructures. It demands the ability to transform heterogeneous raw data into structured, high-quality information that can directly support operational decisions and advanced analytics. In industrial practice, however, this transformation is rarely straightforward. Data sources are fragmented, acquisition systems are heterogeneous, and existing platforms are often tightly coupled with proprietary vendor ecosystems.

This chapter presents the design and implementation of the Data Concentrator, an edge-based platform developed within the scope of this PhD research to address these challenges. The Concentrator was conceived, designed, and implemented by the candidate in collaboration with AROL Closure Systems, starting from an initial set of industrial needs and progressively evolving into a fully functional prototype deployed on real production machinery.

Unlike commercial off-the-shelf solutions, the Data Concentrator is not a fixed product, but a flexible and customizable framework aimed at enabling data valorization at the edge. Its purpose is to systematically collect, preprocess, and transform heterogeneous industrial data streams into meaningful features that can be consumed by predictive maintenance applications and higher-level analytics.

The core idea underlying the proposed approach is that value in Industrial IoT does not primarily originate from data acquisition, but from data transformation. The Concentrator therefore acts as an industrial-grade fog node positioned between the operational technology (OT) layer—composed of machines, sensors, and controllers—and the information technology (IT) layer, where large-scale analytics and decision-making processes take place.

From an architectural perspective, the Concentrator has been designed around the following principles:

- Vendor-agnostic data integration,
- Modular and containerized analytics,
- Extensibility and configurability,
- Deployment at the edge close to data sources,
- Interoperability with existing industrial infrastructures.

The contribution of this chapter is not the introduction of a new computing paradigm, but the concrete realization of a data-centric architecture that enables practical and scalable predictive maintenance workflows in an industrial context.

The remainder of the chapter describes the industrial scenario, the requirements elicitation process, the architectural design, and the implementation of the platform, highlighting at each step the specific contributions of the candidate.

3.2 The Production Line

3.2.1 AROL Capping Machines



Fig. 3.1 AROL Capping Machine

AROL Closure Systems, founded in 1978 in Canelli (Italy), is a global leader in the design and manufacturing of industrial capping machines. The company delivers more than 700 machines per year and supports over 25,000 installations worldwide across several industrial sectors, including beverage, food, pharmaceutical, and cosmetics.

AROL machines perform the application of closures such as screw caps, press-on caps, crown corks, and capsules to containers. Depending on production requirements, machines can be rotary or linear and operate at speeds ranging from 1,000 to 100,000 bottles per hour.

A typical capping machine integrates multiple subsystems:

- Cap feeding and orientation mechanisms,

- Pick-and-place units,
- Torque-controlled capping heads,
- Sealing units,
- Inspection and quality control modules.

Despite the high level of automation, these machines remain subject to several operational challenges, including mechanical wear, misalignments, torque deviations, and feeding anomalies. Such issues can lead to unplanned downtime, reduced product quality, and increased maintenance costs.

The industrial objective motivating this research was therefore to enable early detection of anomalies and progressive degradation phenomena through a data-driven predictive maintenance approach.

3.3 Project Motivation and Requirements Elicitation

The motivation behind the Data Concentrator stems from two complementary needs:

- **Operational insight:** real-time detection of anomalies and progressive degradation phenomena on the shop floor to reduce downtime and maintenance costs.
- **Data valorization:** transforming raw measurements into structured features suitable for predictive maintenance and integration with higher-level analytics platforms.

A thorough requirements elicitation process was conducted in collaboration with industrial engineers and operators. Functional requirements addressed multi-protocol connectivity (BLE, Wi-Fi, OPC UA, Modbus), local preprocessing, feature extraction, anomaly detection, fault tolerance, and secure communication. Non-functional requirements emphasized industrial-grade robustness, environmental resilience, maintainability, and scalability for future expansions.

3.4 Candidate Contribution

The main contribution of this research lies in the design and implementation of the Data Concentrator as a flexible, edge-based platform that:

- Integrates heterogeneous OT and IoT devices, including the next-generation sensor nodes.
- Performs data preprocessing and feature generation locally, minimizing latency and dependency on centralized systems.
- Supports containerized microservices, allowing integrators to deploy custom processing pipelines and lightweight machine learning models at the edge.
- Provides secure, reliable, and extensible data export mechanisms for downstream analytics and predictive maintenance applications.

This contribution goes beyond the extension of already existing systems in AROL, introducing modularity, containerization, and industrial-grade deployment suitable for heterogeneous production environments.

3.4.1 Sensor Nodes

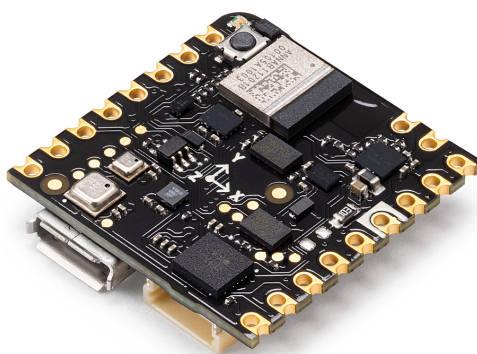


Fig. 3.2 Nicla Sense ME

The Arduino Nicla Sense ME is a compact, low-power sensor node for environmental and motion monitoring. It captures temperature, humidity, pressure, and acceleration

values for experimental setups.

Powered by a Cortex-M4 microcontroller, it performs data acquisition, buffering, preprocessing, and low-latency communication with the Concentrator. Integrated sensors include:

- Motion: 3-axis accelerometer and gyroscope,
- Environmental: temperature, humidity, pressure, air quality, CO₂.

Data is transmitted via Bluetooth Low Energy (BLE) 4.2, supporting multiple proximate nodes with minimal interference. Ultra-low power consumption allows continuous monitoring over extended periods. The Nicla Sense ME can be densely deployed for high-resolution spatial and temporal data collection.



Fig. 3.3 The Sensor Node (NS2)

3.5 System Requirements

3.5.1 Functional Requirements

The functional requirements of the Data Concentrator reflect its role as a pivotal subsystem for data acquisition, local processing, and secure communication within industrial environments. These requirements are articulated across several dimensions: connectivity, data handling, user interaction, fault management, security, and lifecycle support.

Data Acquisition and Connectivity

A primary requirement of the Concentrator is the ability to acquire data streams from a wide range of heterogeneous devices (RQ.F30, RQ.F40). These include AROL's proprietary sensor networks, third-party IoT devices, and traditional operational technology (OT) systems. To achieve this, the platform must support multiple communication standards. On the IoT side, wireless protocols such as Bluetooth Low Energy (BLE) (RQ.F40.1.1), Wi-Fi (with transport support for MQTT, HTTP, TCP/IP, and CoAP) (RQ.F40.1.2), and potentially Ultra Wide Band (UWB) (RQ.F40.1.3) are envisaged. For industrial connectivity, the Concentrator must implement Modbus/TCP (RQ.F40.2.1) and OPC-UA (RQ.F40.2.2), ensuring interoperability with programmable logic controllers (PLCs), SCADA systems, and other automation components.

In addition to protocol diversity, the system is required to sustain simultaneous connections with a large number of devices (RQ.F30.5, RQ.F30.6). For instance, the Concentrator must be capable of managing up to 64 concurrent IDS-NS2 sensor nodes while maintaining reliable synchronization with other industrial devices (RQ.F40.2.3). Synchronization signals (e.g., machine zero points and periodic 20 ms markers) must also be supported to enable time-critical coordination across production lines.

Local Data Processing

Beyond raw data collection, the Concentrator must perform local preprocessing (RQ.F50). This includes filtering invalid or out-of-range values (RQ.F50.2), ag-

gregating measurements from multiple devices (RQ.F50.3), and normalizing data into formats compatible with downstream systems (RQ.F50.4). Where necessary, calibration procedures should be applied to ensure accuracy and comparability.

Performance requirements emphasize both low latency and high sampling frequency (RQ.F50.5), enabling real-time responsiveness and alignment with fast-paced industrial operations. The platform must ensure that preprocessing does not introduce significant delays, thus preserving the integrity of time-sensitive data streams.

Human–Machine Interaction

A lightweight human–machine interface is required to facilitate device configuration, diagnostics, and data visualization (RQ.F20). The interface will be web-based, ensuring accessibility from standard browsers without the need for specialized software. In addition to system configuration, the HMI must allow operators to inspect diagnostic logs and, when necessary, consult historical datasets recorded locally on the Concentrator (RQ.F60.4, RQ.F60.5).

Fault Detection and Error Management

Robust error handling is a critical functional requirement (RQ.F60). The Concentrator must be able to detect and classify errors such as communication failures, transmission losses, and anomalous measurements (RQ.F60.1, RQ.F60.2). Detected faults must be logged and, where appropriate, signaled to external supervisory systems through warnings or alerts (RQ.F60.2, RQ.F60.5).

The system must also incorporate fault tolerance mechanisms. Temporary disconnections should trigger local buffering of data until communication is restored (RQ.F60.4), while critical failures should activate predefined emergency procedures such as controlled shutdown or safe fallback modes (RQ.F60.3). Furthermore, the architecture must distinguish between sacrificable data (e.g., certain temperature readings) and non-sacrificable data (e.g., essential process parameters) (RQ.F60.5, RQ.F60.6), thereby prioritizing information critical to system safety and performance.

Security and Data Protection

Given the physical exposure and potential vulnerability of fog nodes in industrial environments, security requirements are stringent (RQ.F70). The Concentrator must authenticate connected devices, enforce user access controls, and protect data both in transit and at rest. These mechanisms should ensure confidentiality, integrity, and resistance against unauthorized access or tampering.

Event-Driven and On-Demand Communication

The Concentrator must support dual modes of external communication (RQ.F5, RQ.F80). In event-driven scenarios, anomalies or critical events should trigger immediate notifications to supervisory systems such as the IoT Box or industrial PLCs (RQ.F5.5). In on-demand scenarios, operators must be able to query the Concentrator directly, either via the HMI or through an application, to obtain real-time or historical data (RQ.F6.6).

Updates and Lifecycle Management

Finally, the Concentrator must accommodate secure and remote lifecycle management (RQ.F80). This includes over-the-air (OTA) updates of both firmware and software components, support for rollback in case of deployment failures, and continuous monitoring of the system's operational health. These capabilities are essential to maintaining long-term reliability and adaptability in industrial contexts.

3.5.2 Non-Functional Requirements

The non-functional requirements of the Data Concentrator define the quality attributes and constraints that the system must satisfy in addition to its functional capabilities. These requirements are critical to ensure long-term reliability, operational continuity, and suitability for industrial deployment.

Reliability and Robustness

The Concentrator must be designed to operate reliably in industrial environments, which are often subject to mechanical vibrations, temperature variations, and electromagnetic interference (RQ.N20, RQ.G10). Hardware components must be industrial-grade, selected for long operational life and resistance to environmental stressors (RQ.N20.1). Comprehensive testing during production, including stress tests and quality control inspections, is mandatory to minimize hardware failure rates.

From a software perspective, the system must exhibit robustness against bugs, crashes, and unexpected conditions (RQ.N20.2). Fault-tolerant software design, rigorous testing procedures, and secure update mechanisms are necessary to maintain continuous operation. Error handling and recovery procedures must ensure that critical operations are preserved even in the event of component failures (RQ.F60).

Availability and Maintainability

High availability is a key requirement for the Concentrator, reflecting the need for minimal unplanned downtime in industrial processes (RQ.N30). The system should support automated recovery mechanisms and, when possible, hot-swappable components to reduce service interruptions (RQ.N30.2). In case of failure, restoration procedures must be straightforward, minimizing the need for manual intervention (RQ.N30.2, RQ.N30.3).

The platform should be supported by a comprehensive maintenance plan and technical support infrastructure. Remote monitoring capabilities are essential to enable proactive maintenance and timely detection of potential issues before they impact operations (RQ.F80).

Security and Data Integrity

Although primarily a functional concern, security also contributes to the non-functional integrity of the system (RQ.F70). The Concentrator must enforce strong access control, secure authentication of connected devices, and encryption of data both in transit and at rest. These measures not only protect sensitive information but also prevent system disruptions caused by unauthorized access.

Performance and Scalability

Performance requirements extend beyond low-latency data processing, encompassing system throughput, responsiveness under load, and efficient utilization of computational resources (RQ.F50.5, RQ.F30.6). The system must maintain stable operation under peak workloads and scale appropriately to accommodate the addition of new devices or sensors without degradation of service.

Cost and Resource Constraints

Economic feasibility is an additional non-functional constraint (RQ.N10). The target cost of the platform must be balanced with performance and reliability goals, ensuring that the system remains practical for industrial deployment at scale. Hardware and software choices must consider not only upfront costs but also lifecycle expenses, including maintenance, energy consumption, and potential future upgrades.

Environmental Compliance

Finally, the Concentrator must conform to relevant environmental and safety standards applicable to industrial electronic devices (RQ.G10, RQ.G20, RQ.G30). These may include electromagnetic compatibility (EMC), thermal limits, and power efficiency requirements, which collectively ensure safe and sustainable operation in the target deployment context.

3.6 High Level Architecture

3.6.1 Logical Architecture

The logical architecture was developed using the Capella modeling tool. The system is composed of several operational entities, namely: the Data Concentrator, IDS-NS2, Cyber Head, IoT Camera, torque sensors, and the IoT Box, shown in Fig. 3.4.

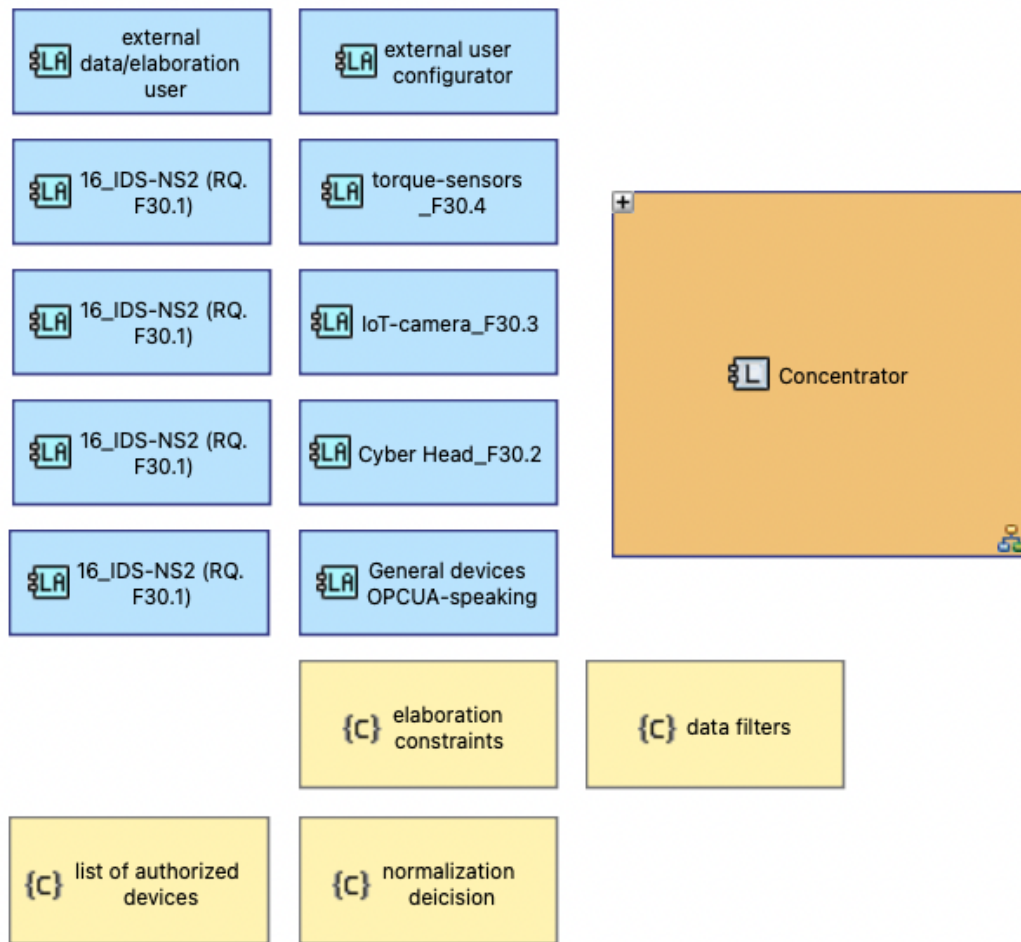


Fig. 3.4 Operational Entities

These actors perform specific logical functions, which together form the overall system architecture: Fig. 3.5 and 3.6.

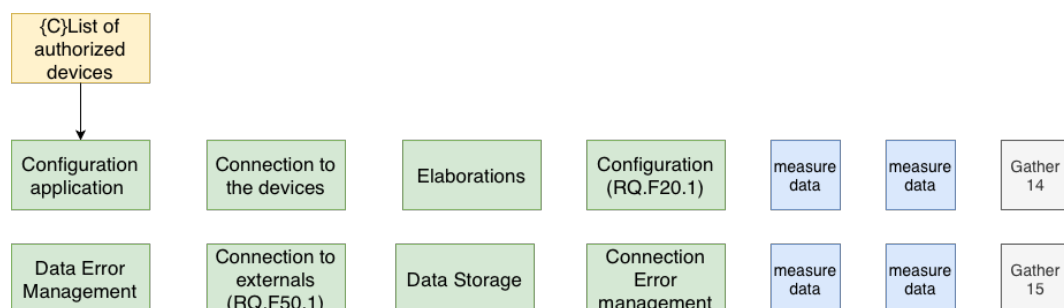


Fig. 3.5 High-level Functions

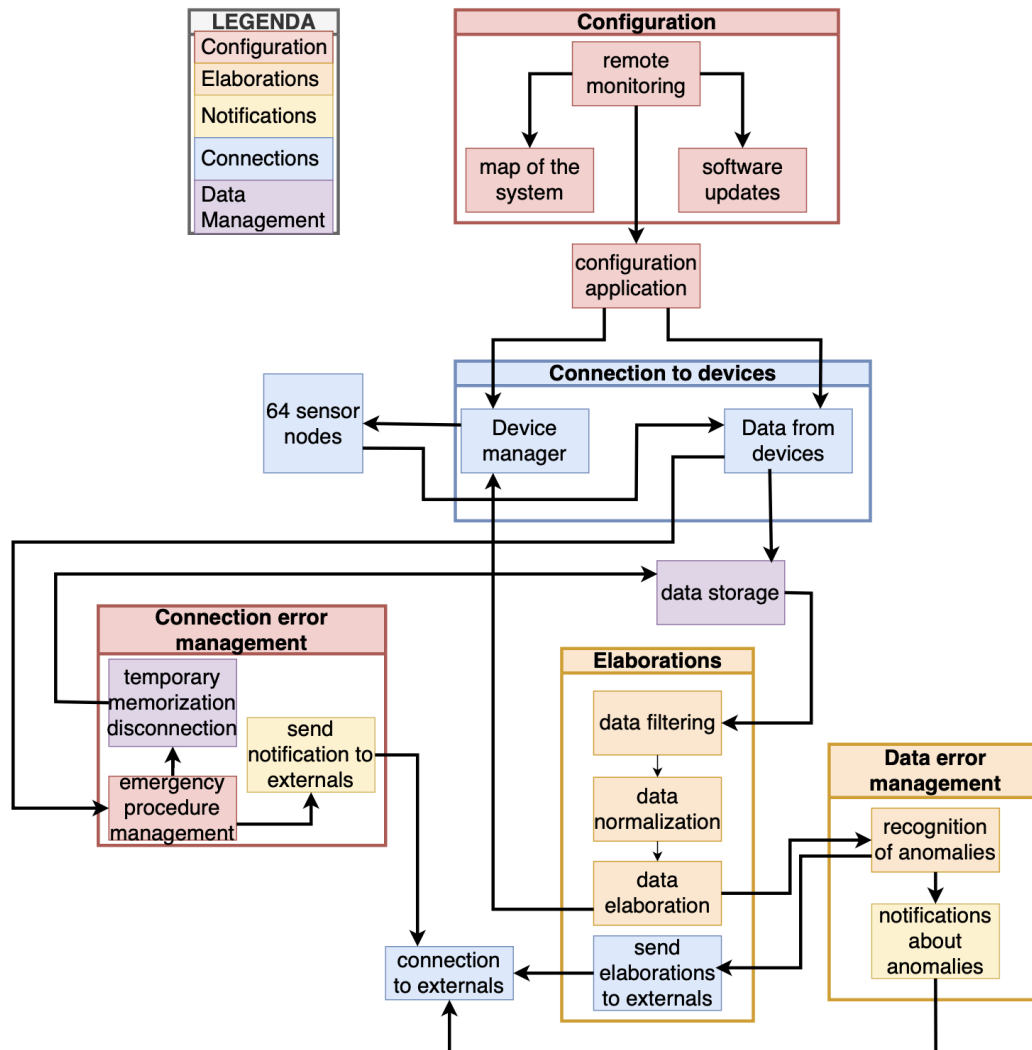


Fig. 3.6 Logical Functions

The general system architecture is presented in Fig. 3.7.

The Data Concentrator is the central component, supporting configuration, connectivity, data acquisition, processing, storage, and external communication.

3.6.2 Configuration (F20.1)

Remote configuration is supported via a web-based interface, which enables:

- System Mapping: identification of device positions and logical roles.

- Software Update (F80.1): remote updating of system firmware and application software.
- Remote Monitoring and Control (F20.2, F80.1): visualization of device status, operational history, and performance graphs.

Configuration constraints (authorized devices, communication parameters) are applied directly to the Data Concentrator.

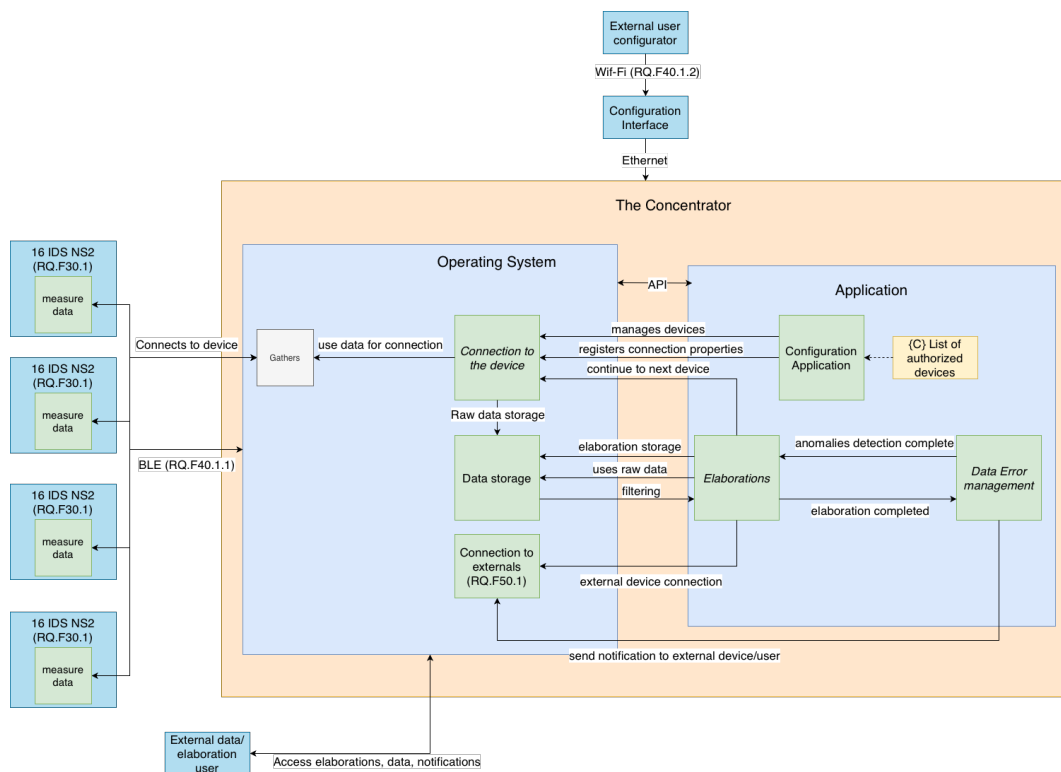


Fig. 3.7 Logical Architecture

Device Connectivity

The Data Concentrator integrates data from multiple devices via heterogeneous communication protocols:

- Device Management (F70.1): recognition of authorized devices.
- Data Collection (F50.3): acquisition of relevant data streams.

In case of communication failures, the system provides:

- Emergency Handling (F60.3): immediate system shutdown or disconnection.
- Temporary Data Storage (F60.4): preservation of recent measurements.
- External Notifications: transmission of alerts to users or external devices.

Both raw and processed data are stored locally, accessible through the configuration interface.

Data analysis functions include:

- Filtering (F50.2): focusing on relevant information.
- Normalization (F50.4): harmonization of heterogeneous sensor outputs.
- Processing (F50.1): computation of results, stored locally.
- External Transmission: processed data sent to external devices or users.

Anomaly Detection

When anomalies are identified:

- Recognition (F60.1): detection within collected or processed data.
- Notification (F60.2): reporting anomalies to external systems (configuration interface, PLC).

External Communication (F50.1)

Processed data and anomaly notifications are transmitted through both wireless (BLE) and wired (OPCUA) channels.

The IDS-NS2 acquires physical measurements (temperature, humidity, pressure, accelerations) and communicates with the Data Concentrator via BLE (F40.1.1), shown in Fig. 3.8 and Fig. 3.9.

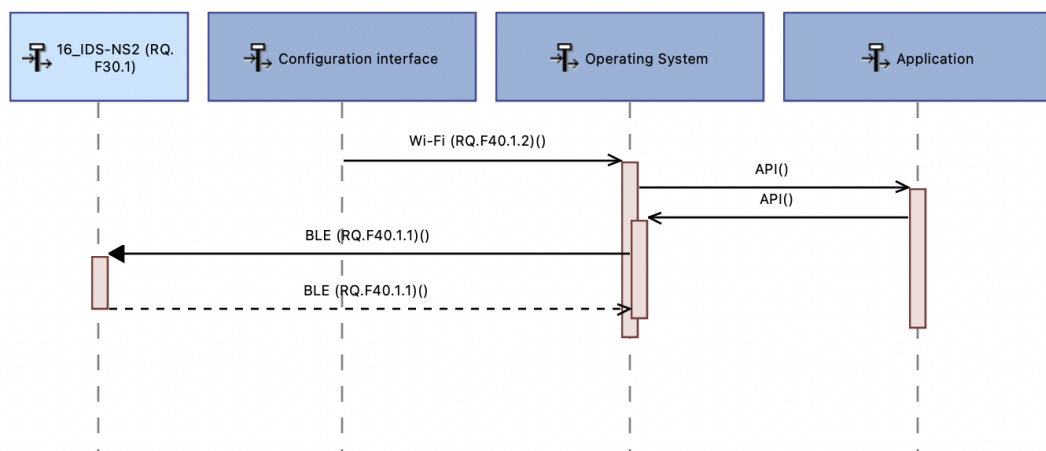


Fig. 3.8 Interfaces between IDS-NS2 and the Data Concentrator

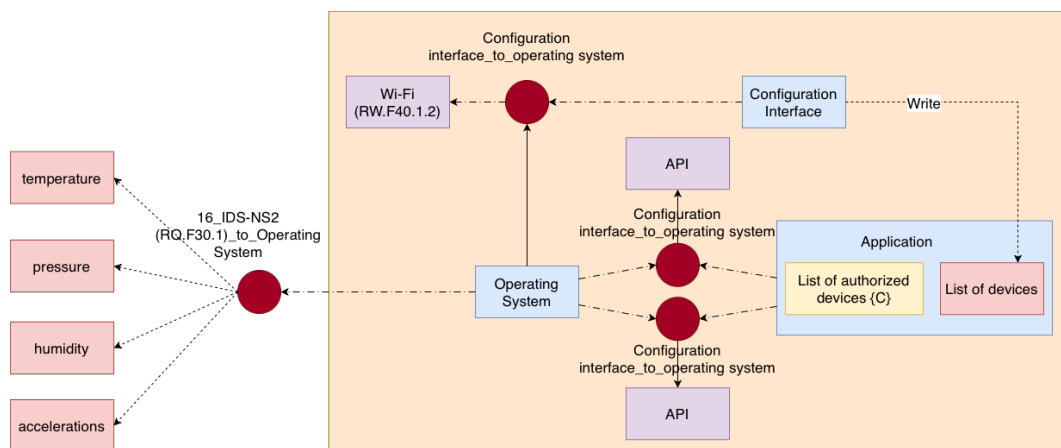


Fig. 3.9 Data exchange architecture

3.6.3 Operational Scenarios

Four scenarios have been modeled:

1. Configuration of the Data Concentrator, Fig. 3.10 and 3.11

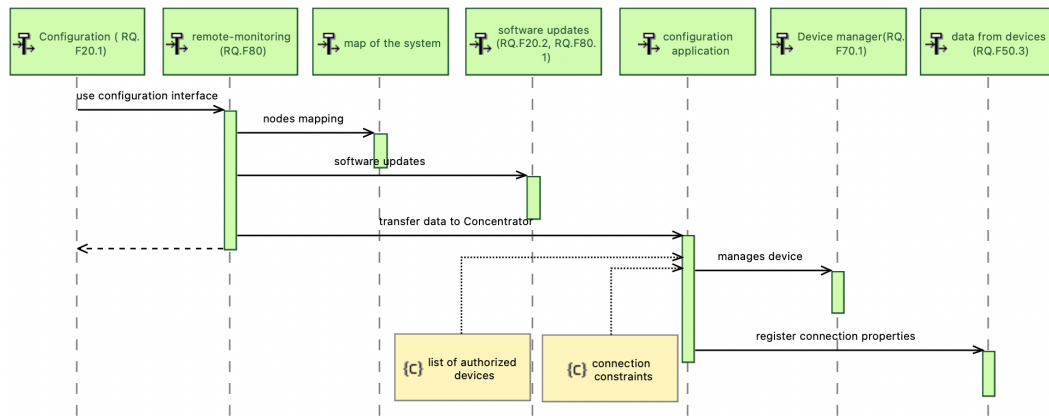


Fig. 3.10 Configuration of the Data Concentrator, part 1

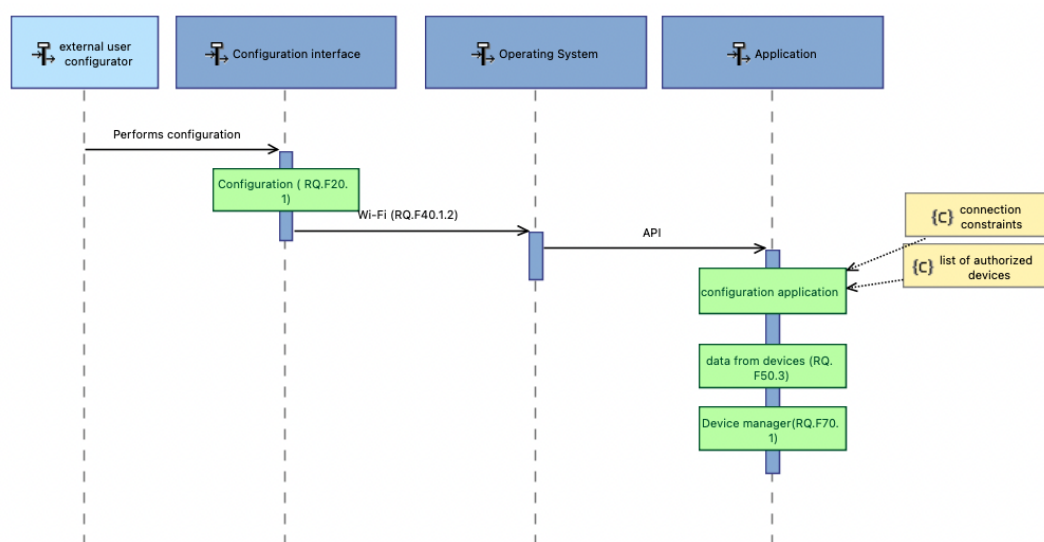


Fig. 3.11 Configuration of the Data Concentrator, part 2

2. Successful connection, normal data processing, Fig. 3.12 and 3.13

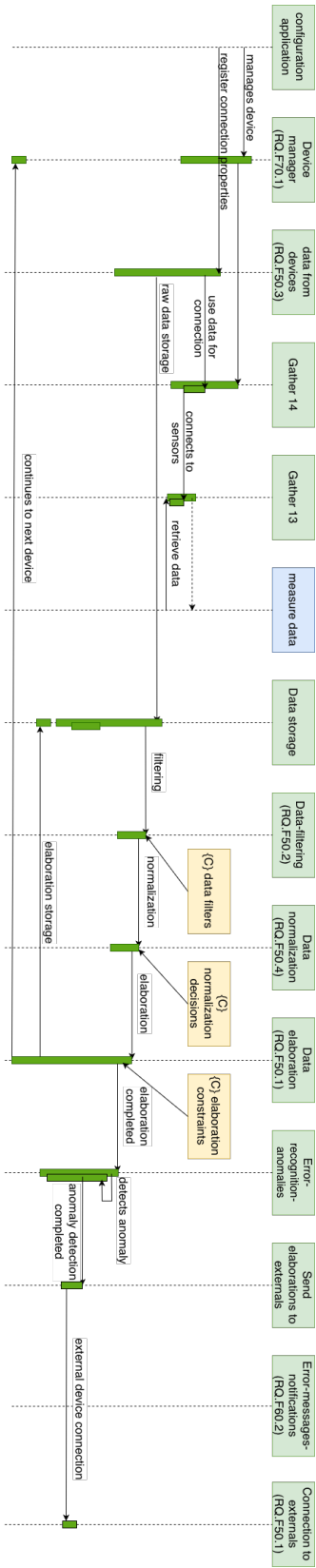


Fig. 3.12 Successful connection, part 1

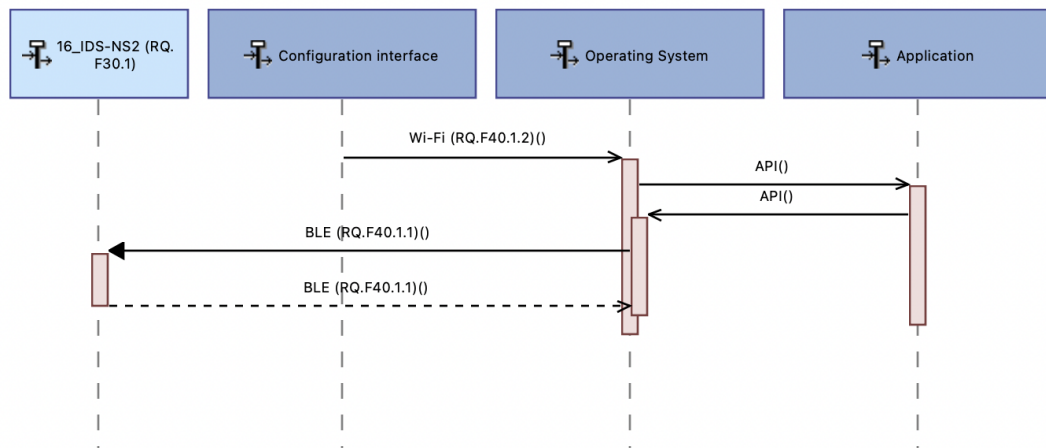


Fig. 3.13 Successful connection, part 2

3. Successful connection, anomaly detected, Fig. 3.14 and 3.15

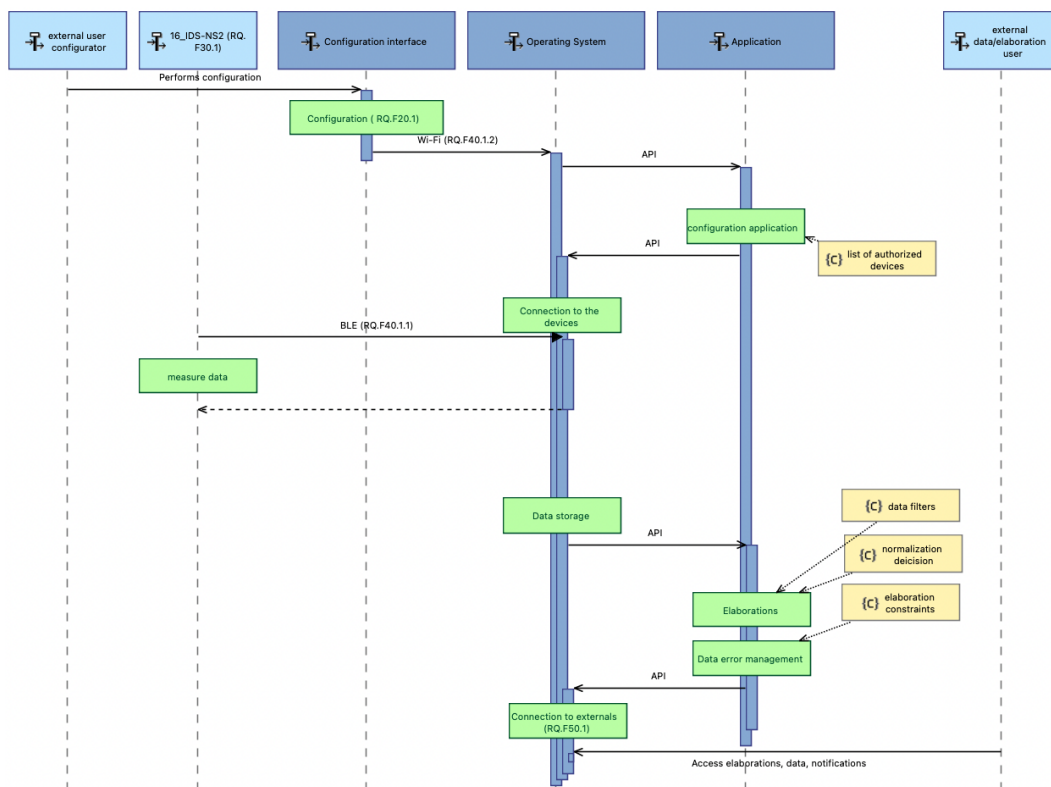


Fig. 3.15 Anomaly detected scenario, part 2

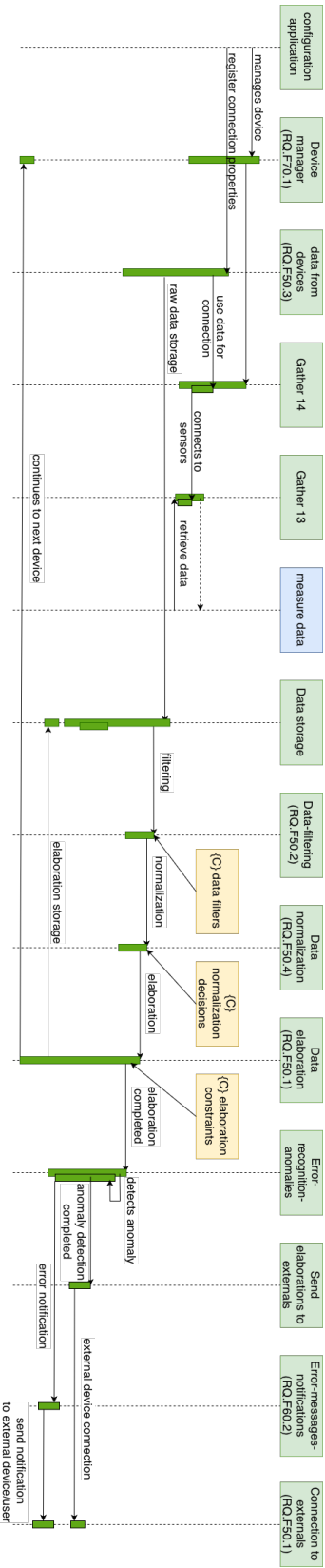


Fig. 3.14 Anomaly detected scenario, part 1

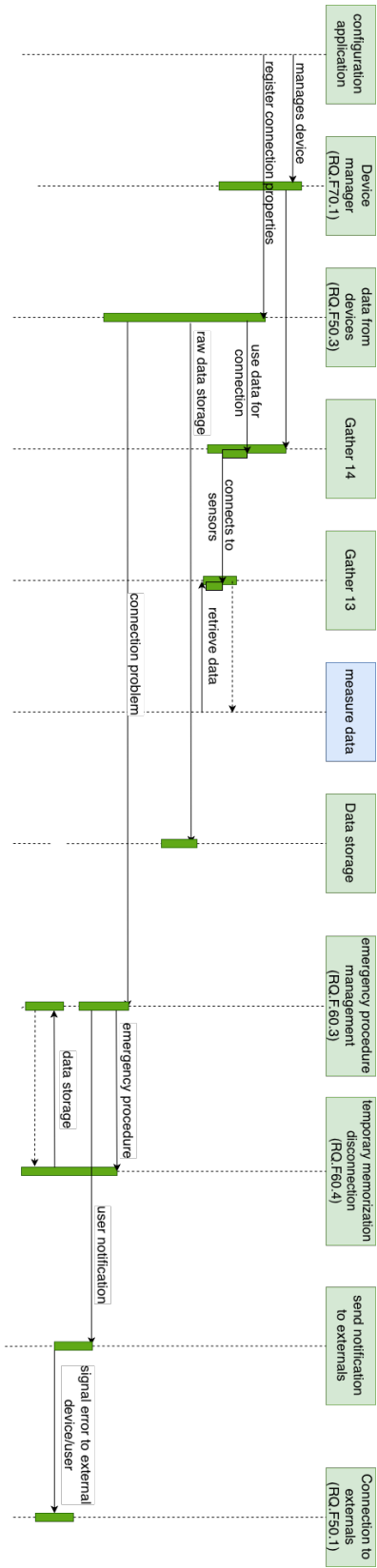


Fig. 3.16 No connection scenario, part 1

- Memory: includes volatile and non-volatile storage. Minimum 2 GB RAM is required. Supports file-system-based storage and external HDD/SSD.

The interaction between components is illustrated in Fig. 3.19 and Fig. 3.21.

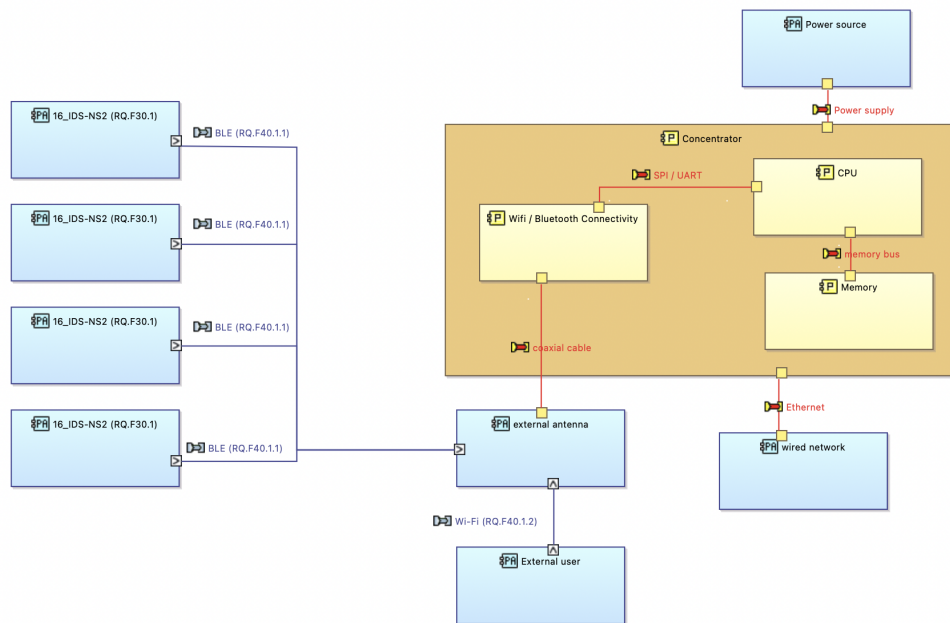


Fig. 3.18 Hardware physical architecture

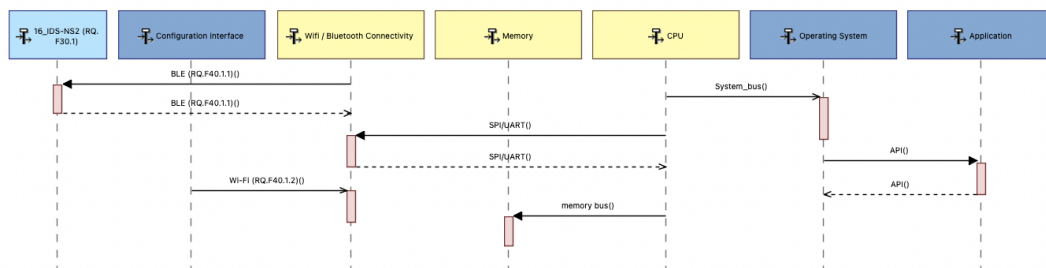


Fig. 3.19 Hardware interface scenario

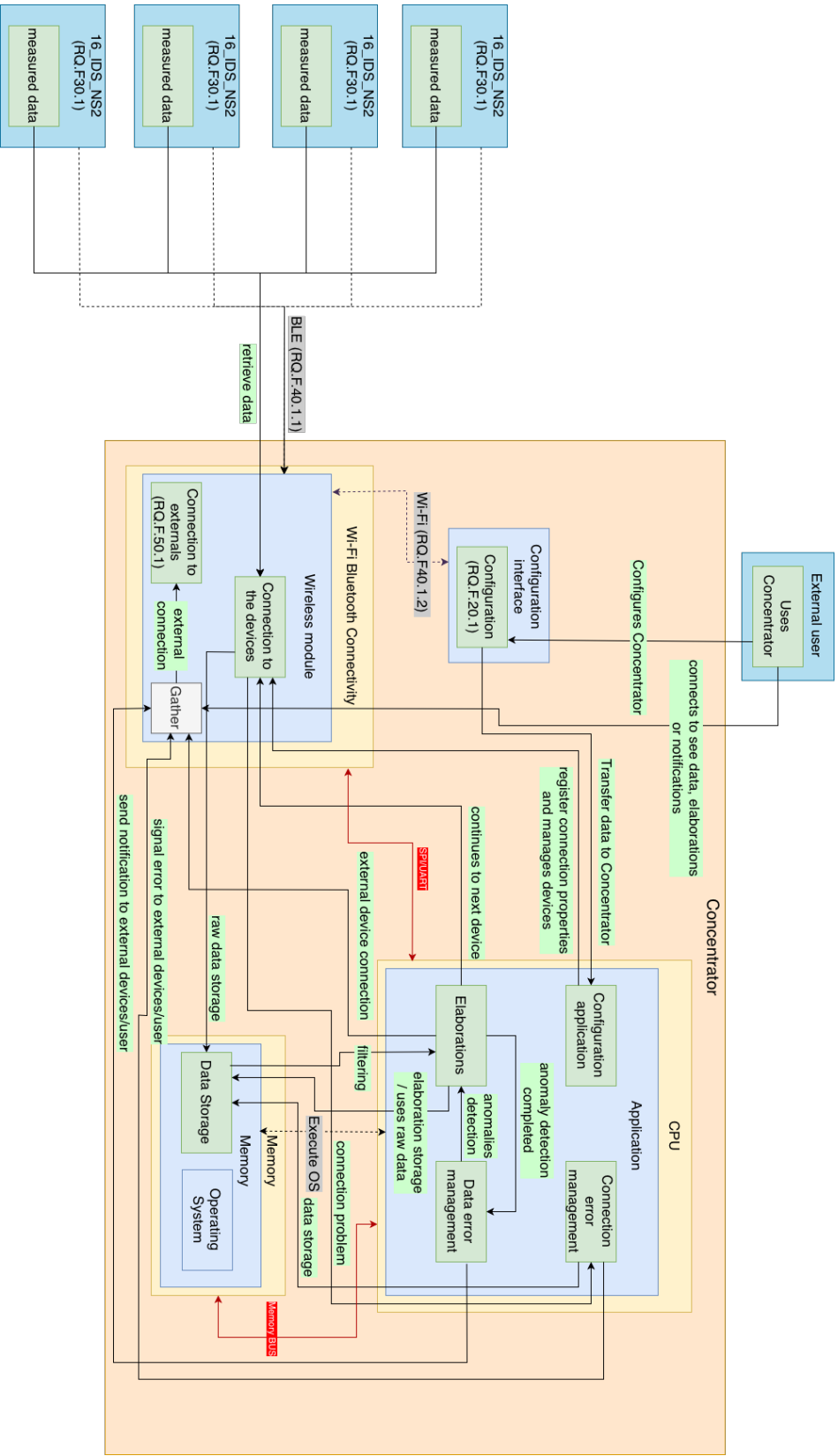


Fig. 3.20 Software physical architecture

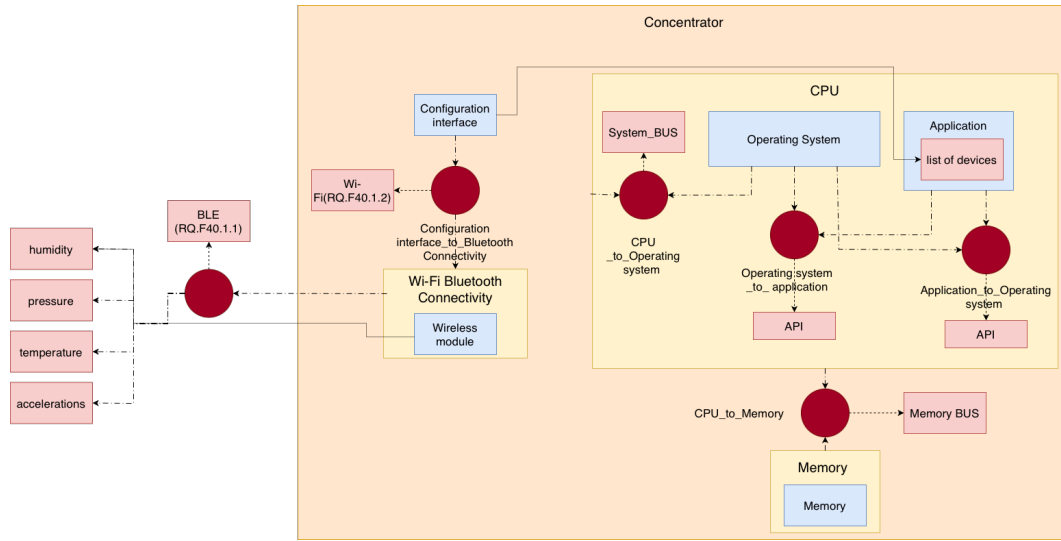


Fig. 3.21 Interface architecture of physical components

3.7 Hardware Selection

3.7.1 Hardware Requirements

The hardware selection for the Data Concentrator was guided by the need to balance functional and non-functional requirements, ensuring reliable, real-time operation in industrial environments while maintaining flexibility for future expansions (RQ.F50, RQ.F30, RQ.N20, RQ.N30, RQ.G10–G30, RQ.N10).

Key considerations included computational performance (RQ.F50.5), support for multiple communication protocols (RQ.F40, RQ.F30.6), energy efficiency and flexible power supply options (RQ.E10.1, RQ.E10.2), environmental robustness against vibrations, temperature fluctuations, and electromagnetic interference (RQ.G10, RQ.N20.1), maintainability and support for remote monitoring and updates (RQ.F80, RQ.N30.3), and overall cost-effectiveness (RQ.N10).

The Concentrator must manage multiple concurrent data streams from heterogeneous IoT and OT devices (RQ.F30.5, RQ.F30.6), perform local preprocessing and aggregation (RQ.F50.1–F50.4), and support containerized microservices (RQ.F50,

RQ.F80). Therefore, the hardware requires sufficient CPU and memory resources to enable low-latency processing and high sampling rates (RQ.F50.5), industrial-grade I/O interfaces for connectivity with sensors and PLCs (RQ.F40.2, RQ.F30), and reliable network interfaces capable of sustaining simultaneous connections (RQ.F30.5, RQ.F30.6).

Environmental resilience is critical, including tolerance to mechanical vibrations, thermal variations, and electromagnetic interference (RQ.G10, RQ.N20.1). Power supply flexibility, such as standard 24 V DC or Power over Ethernet (PoE), was also considered to ensure operational adaptability (RQ.E10.1, RQ.E10.2). Finally, the platform needed to be scalable for future sensor integration or software enhancements, supporting incremental addition of devices or deployment of new services without compromising stability or performance (RQ.F30.6, RQ.F50.5, RQ.F80).

3.7.2 Candidate Hardware Platforms

Four hardware platforms were evaluated for the Data Concentrator:

Standard Linux PC



Fig. 3.22 HP EliteBook Laptop used for prototyping

An HP EliteBook laptop equipped with an Intel Core i7 processor running Ubuntu 20.04.5 LTS 64-bit was employed for initial prototyping and software development. This platform provided a high-performance, flexible, and reliable environment for testing the concentrator software before deployment on embedded or industrial hardware.

Technical features:

- CPU: Intel Core i7 quad- or hexa-core, x86-64 architecture with Hyper-Threading.
- RAM: 8–32 GB DDR4, suitable for multi-threaded processing and containerized services.
- Storage: SSD (typically 256 GB+), providing fast I/O for local databases and sensor data caching.
- Connectivity: Gigabit Ethernet, dual-band Wi-Fi 802.11ac, Bluetooth 5.0, multiple USB ports, HDMI and internal display outputs.

Advantages: high computational performance, full Linux environment, extensive peripheral connectivity.

Limitations: lack of industrial-grade robustness, higher power consumption, and unsuitable form factor for industrial integration.

Raspberry Pi 4 Model B

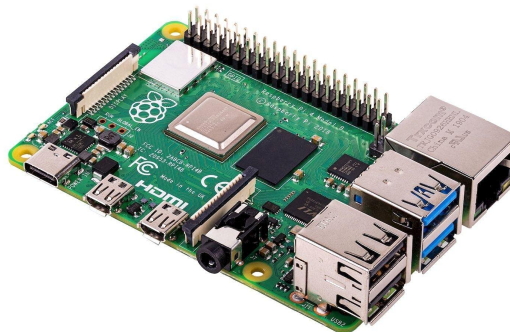


Fig. 3.23 Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B (RPi 4B) is a compact single-board computer suitable for prototyping and embedded applications. Despite its low cost, it provides sufficient resources for functional testing and rapid development.

Technical features:

- CPU: Broadcom BCM2711, quad-core ARM Cortex-A72 (1.5 GHz).
- RAM: 2–8 GB LPDDR4.
- Storage: microSD card or USB mass storage.
- Connectivity: Gigabit Ethernet, dual-band Wi-Fi, Bluetooth 5.0, 2× USB 3.0 + 2× USB 2.0, HDMI, CSI/DSI connectors, 40-pin GPIO header.
- Power: 5 V/3 A USB-C, optional PoE via HAT module.

Strengths: small form factor, low cost, rapid prototyping.

Limitations: limited environmental tolerance, electromagnetic shielding, and mechanical robustness, making it unsuitable for harsh industrial conditions.

Arduino Portenta X8

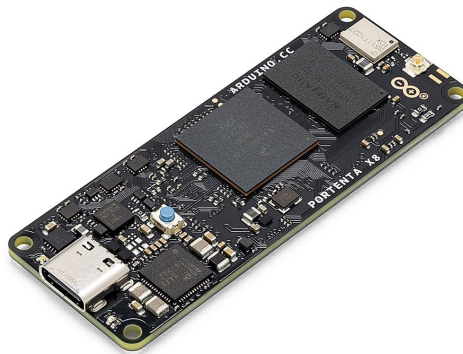


Fig. 3.24 Arduino Portenta X8

The Arduino Portenta X8 is an industrial-grade System-on-Module (SoM) designed for high-performance computing and robust connectivity in embedded and IoT applications.

Technical features:

- CPU: NXP i.MX 8M Mini, quad-core Cortex-A53 (up to 1.8 GHz) + Cortex-M4 co-processor.
- Microcontroller: Microchip SAMD21 for secure element management.
- RAM/Storage: 4 GB LPDDR4, 16 GB eMMC.
- Connectivity: Gigabit Ethernet, Wi-Fi, Bluetooth 5.0, USB 2.0/OTG, UART, SPI, I²C, CAN, GPIO.
- Security: cryptographic co-processor, secure boot, trusted execution support.
- Power: flexible industrial input.

Strengths: dual-core architecture for Linux and real-time control, industrial-grade robustness, extended temperature range, EMI resistance.

Limitations: higher cost and increased system complexity compared to lightweight prototyping platforms.

EdgeBox RPi 200



Fig. 3.25 EdgeBox RPi-200

The EdgeBox RPi-200 is a ruggedized industrial edge computing controller based on the Raspberry Pi Compute Module 4, designed for industrial environments.

Technical features:

- CPU: Quad-core ARM Cortex-A72 (1.5 GHz).

- RAM/Storage: 4 GB LPDDR4, 16 GB eMMC; M.2 expansion supported.
- Connectivity: Gigabit Ethernet, dual-band Wi-Fi, Bluetooth 5.0, USB 2.0, HDMI, Mini PCIe (4G, LoRa, Zigbee), SIM card slot.
- Industrial I/O: RS232/RS485, digital inputs/outputs (DI/DO), isolated.
- Power: 9–36 V DC, DIN rail mounting.
- Environmental: -20°C to 60°C, aluminum enclosure, fanless.

Applications: IIoT gateway, PLC controller, data acquisition system, edge computing node.

Strengths: industrial-grade robustness, extended temperature range, versatile connectivity, and scalable architecture.

Table 3.1 Comparison of Hardware Platforms for the Data Concentrator

Feature	Raspberry Pi 4B	Portenta X8	EdgeBox RPi 200	HP EliteBook i7
Processor	Quad-core ARM Cortex-A72 1.5 GHz	Quad-core Cortex-A53 + Cortex-M4	Quad-core ARM Cortex-A72 1.5 GHz	Quad-/Hexa-core Intel Core i7
RAM	2–8 GB LPDDR4	Up to 4 GB LPDDR4	4 GB LPDDR4	8–32 GB DDR4
Storage	microSD	16 GB eMMC	16 GB eMMC + M.2/SSD	SSD 256 GB+
Connectivity	Ethernet, Wi-Fi, BLE, USB, HDMI, GPIO	Ethernet, Wi-Fi, BLE, USB, UART, SPI, CAN, GPIO	Ethernet, Wi-Fi, BLE, RS232/485, DI/DO, PCIe	Ethernet, Wi-Fi, BLE, USB, HDMI
Industrial Suitability	Low	High	High	Low
Power	5 V/3 A USB-C	Flexible industrial input	9–36 V DC	AC Adapter / Battery
Primary Use	Rapid prototyping	Industrial edge	Industrial edge	Dev. / Prototyping
Cost	Low	High	Medium-High	High

3.7.3 Final Selection Justification

The selection of the final hardware platform for the Data Concentrator was based on a systematic evaluation of computational performance, industrial robustness, form factor, power supply flexibility, and scalability, weighing the trade-offs of each candidate.

HP EliteBook i7: While offering the highest computational performance and ample storage, the standard laptop lacks industrial-grade robustness, EMI shielding, and flexible power supply options. Its large form factor and high power consumption make it unsuitable for continuous deployment in industrial environments, despite being ideal for prototyping and software development.

Raspberry Pi 4B: This platform is cost-effective, compact, and suitable for rapid prototyping. However, it does not provide industrial-grade protection, has limited tolerance to temperature variations and vibrations, and relies on standard USB-C power input, making it unreliable for continuous operation on industrial machinery.

Arduino Portenta X8: The Portenta X8 combines industrial resilience with a compact form factor suitable for machine integration. Its dual-core architecture allows concurrent Linux and real-time processing. However, its computational resources (CPU, RAM, storage) are limited compared to the EdgeBox or a standard PC, potentially constraining containerized analytics and future expansion. Additionally, its higher cost and increased deployment complexity make scaling more challenging.

EdgeBox RPi-200: The EdgeBox RPi-200 represents the best compromise between performance, industrial suitability, and scalability. Based on the Raspberry Pi Compute Module 4 but ruggedized for industrial deployment, it provides sufficient CPU and memory for low-latency, containerized feature extraction and analytics. Its fanless design, extended operating temperature range (-20°C to 60°C), EMI resilience, flexible power input (9–36 V DC), and native industrial I/O interfaces (RS232/RS485, digital I/O) ensure reliable operation under harsh conditions. The platform's modularity, including M.2 and Mini PCIe expansion, supports future sensor integration or communication upgrades without compromising stability.

Overall, while each candidate offers specific advantages, the EdgeBox RPi-200 fulfills the full set of functional and non-functional requirements (robustness, connectivity, scalability, real-time capability) necessary for an industrial-grade, high-performance Data Concentrator. Its selection reflects a deliberate trade-off between

computational capacity, form factor, deployment reliability, and future extensibility, making it the most suitable choice for the intended operational context.

3.8 Software Architecture

3.8.1 Overview

The software architecture of the Data Concentrator has been designed with a clear objective: to maximize the value of industrial data streams by transforming raw measurements into actionable information. Rather than acting merely as a gateway or relay, the Concentrator provides a structured environment where data is acquired, preprocessed, enriched, and flexibly distributed to downstream systems. In this sense, it operates as a central element in the data value chain, bridging heterogeneous industrial devices and higher-level applications deployed at the edge or in the cloud [22, 111].

The architecture ensures that industrial data is not only collected, but also systematically refined into features that can serve predictive maintenance, process optimization, and advanced analytics. This approach empowers both machine producers and third-party integrators to create customized, data-driven services, shifting the focus from hardware infrastructure to the economic and operational value generated by data.

The overall architecture is depicted in Figure 3.26.

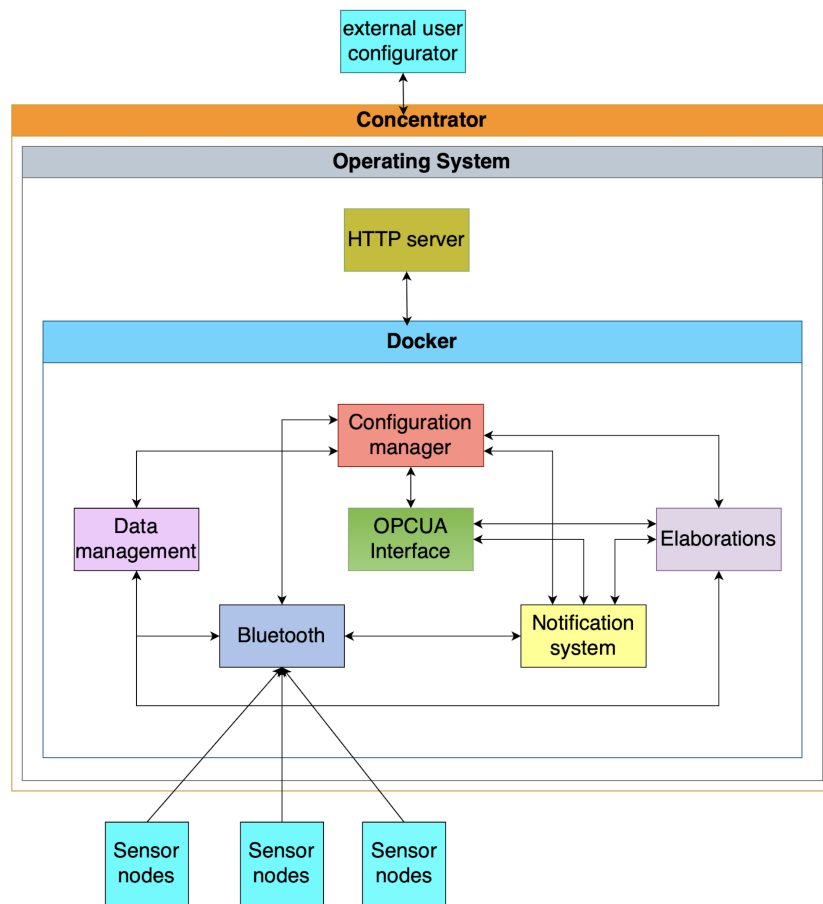


Fig. 3.26 Software Architecture of the Data Concentrator

3.8.2 Key Architectural Components

The software is organized into modular layers, each responsible for a specific phase of the data lifecycle. This layered design supports scalability, robustness, and user-driven customization:

- **Device Interface Layer:** Provides connectivity to heterogeneous IoT and OT devices, including PLCs, proprietary sensors, and legacy equipment. It supports multiple protocols such as OPC UA, Modbus, MQTT, BLE, and TCP/IP. This layer abstracts protocol-specific details, offering a uniform interface for subsequent data processing.
- **Data Preprocessing and Feature Generation Layer:** At the core of the architecture, this layer ensures that raw sensor streams are cleaned, synchronized,

and transformed into features of higher semantic value. Feature generation can rely on built-in algorithms (e.g., spectral descriptors, statistical indicators) or on user-defined routines contributed by integrators. This duality ensures that the Concentrator is not limited to predefined metrics but becomes a platform for custom feature engineering tailored to diverse operational contexts.

- **Container Host Layer:** The system natively supports containerized microservices, deployed through Docker. Containers allow third-party developers to implement and execute their own preprocessing pipelines, feature generators, or lightweight machine learning models directly at the edge, without modifying the Concentrator's core software. This design fosters personalization, ensures isolation between services, and enables rapid prototyping of domain-specific logic.
- **Data Aggregation and Storage Layer:** Enriched data and derived features are aggregated locally and can be cached in lightweight edge databases. This provides resilience against network outages, supports high-throughput scenarios, and allows near-real-time access for local analytics services.
- **Communication and Export Layer:** Processed data is exported to cloud platforms, MES, ERP systems, or other consumers. This layer supports configurable, secure output channels, including REST APIs, MQTT brokers, and OPC UA servers. By decoupling local feature generation from external data consumption, it ensures interoperability across heterogeneous ecosystems.

3.8.3 Data-Centric Value of the Architecture

The Concentrator's software architecture is designed not simply to move data, but to enhance its value. Its modularity enables three core advantages:

1. **Real-time feature generation:** Local preprocessing supports predictive maintenance and anomaly detection directly on the shop floor, reducing latency and enabling immediate response.
2. **Scalable and modular deployment:** Containerized services allow incremental expansion of analytical capabilities—such as the deployment of new models or domain-specific feature generators—without affecting system stability.

3. **Data enrichment and personalization:** By supporting customized feature engineering, the system transforms generic sensor readings into actionable insights, tailored to the needs of specific stakeholders and industrial contexts.

3.8.4 Illustration of Containerized Customization

Figure 3.27 illustrates the role of the container host as an execution environment for user-defined data services. Each container can implement bespoke logic—such as generating composite features, integrating external datasets, or deploying anomaly detection models—while operating independently from the Concentrator’s core. This isolation ensures robustness and security, while enabling experimentation and customization.

Such flexibility provides a competitive advantage: integrators and service providers can develop unique, domain-specific insights from the same raw data. In this way, the Concentrator becomes not a closed hardware solution, but an open platform for data-driven innovation.

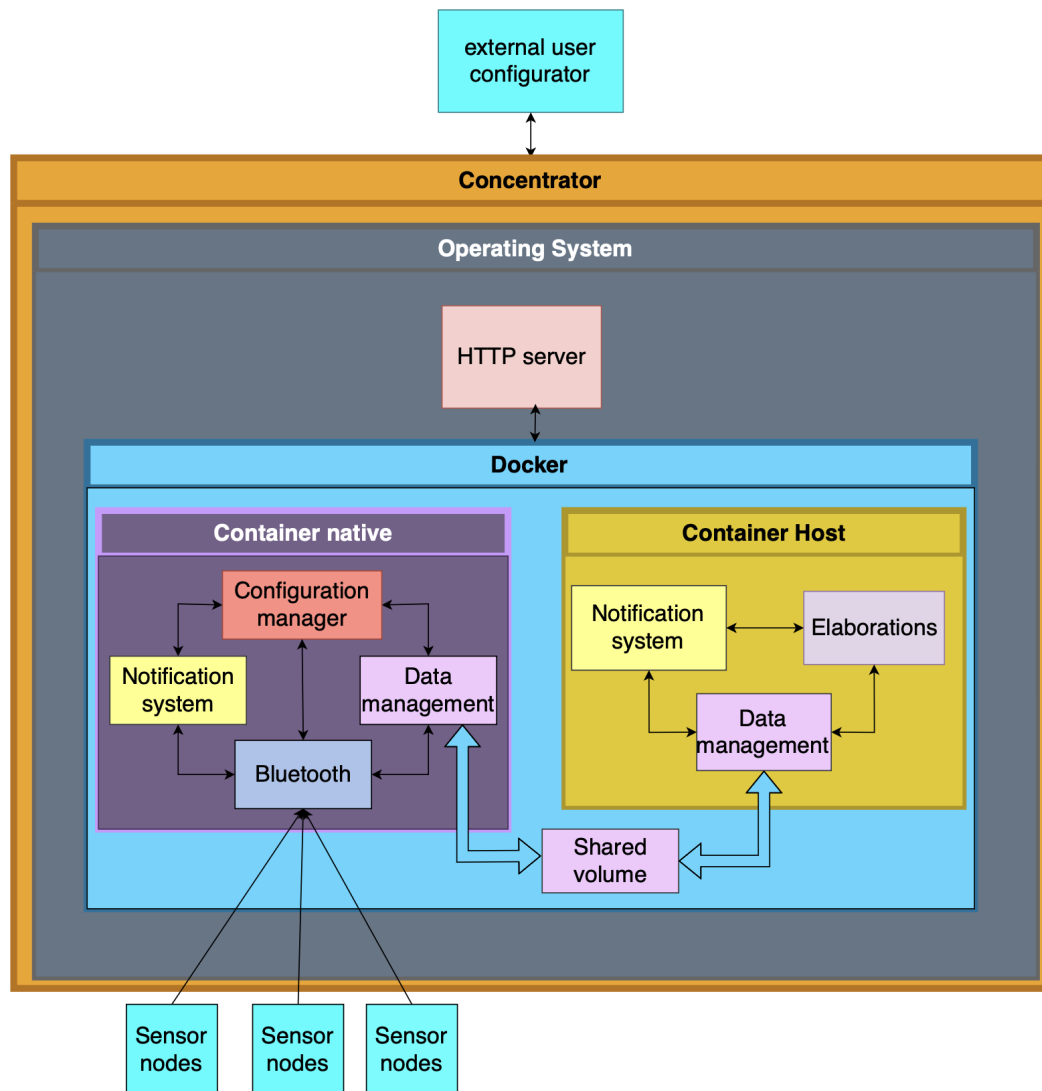


Fig. 3.27 Containerized Microservices for Custom Feature Generation

3.8.5 Summary

In summary, the software architecture of the Data Concentrator:

- acquires and harmonizes data from heterogeneous industrial sources;
- generates features through both native and user-defined algorithms;
- supports modular customization via containerized microservices;

- provides secure, robust, and interoperable data export mechanisms.

Its primary contribution lies in shifting the focus from raw data collection to the systematic generation of actionable insights. By enabling personalization and extensibility, the Concentrator strengthens the role of data as a strategic asset in predictive maintenance and data-driven industrial operations.

Chapter 4

Predictive Maintenance through Slow Signals

Predictive maintenance has become a critical strategy in modern industrial systems, aiming to anticipate equipment failures and optimize maintenance schedules in order to minimize downtime and operational costs. In complex machinery, temperature variations are often early indicators of anomalous behavior, reflecting gradual degradation processes that remain hidden under conventional monitoring approaches. These so-called *slow signals* evolve over extended periods of time, making their analysis particularly suitable for data-driven models capable of capturing long-term temporal dependencies.

In this chapter, we present the implementation of a containerized Long Short-Term Memory (LSTM) neural network for forecasting temperature dynamics from slow signals. LSTM networks are particularly well suited for this task due to their ability to retain long-term information in sequential data and to capture subtle trends that may signal the onset of faults. By integrating slow signal analysis with advanced sequence modeling techniques, this study demonstrates how edge-level machine learning can enhance the reliability and efficiency of industrial systems.

The chapter is structured as follows. Section 4.1 introduces the theoretical background of LSTM networks. Section 4.2 describes the dataset and preprocessing pipeline. Section 4.3 details the architecture and design of the LSTM container. Section 4.4 presents the training and validation strategy. Finally, Sections 4.5 and

4.6 discuss the experimental results and implications for predictive maintenance strategies.

4.1 Long Short-Term Memory Networks for Predictive Maintenance

Long Short-Term Memory (LSTM) networks [112] are a specialized type of recurrent neural network (RNN) designed to model sequential data with long-range dependencies. Unlike standard RNNs, which often suffer from vanishing or exploding gradients, LSTMs employ a memory cell and gating mechanisms that regulate the flow of information. This enables the network to selectively retain or discard information across long time horizons, making it particularly suitable for predictive maintenance applications where slow-evolving signals, such as temperature, play a crucial role.

As shown in Figure 4.1, an LSTM unit is composed of a memory cell $\mathbf{c}_t$ and three gates: input $\mathbf{i}_t$, forget $\mathbf{f}_t$, and output $\mathbf{o}_t$. Their interactions are governed by the following equations:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (4.1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (4.2)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (4.3)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \quad (4.4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad (4.5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (4.6)$$

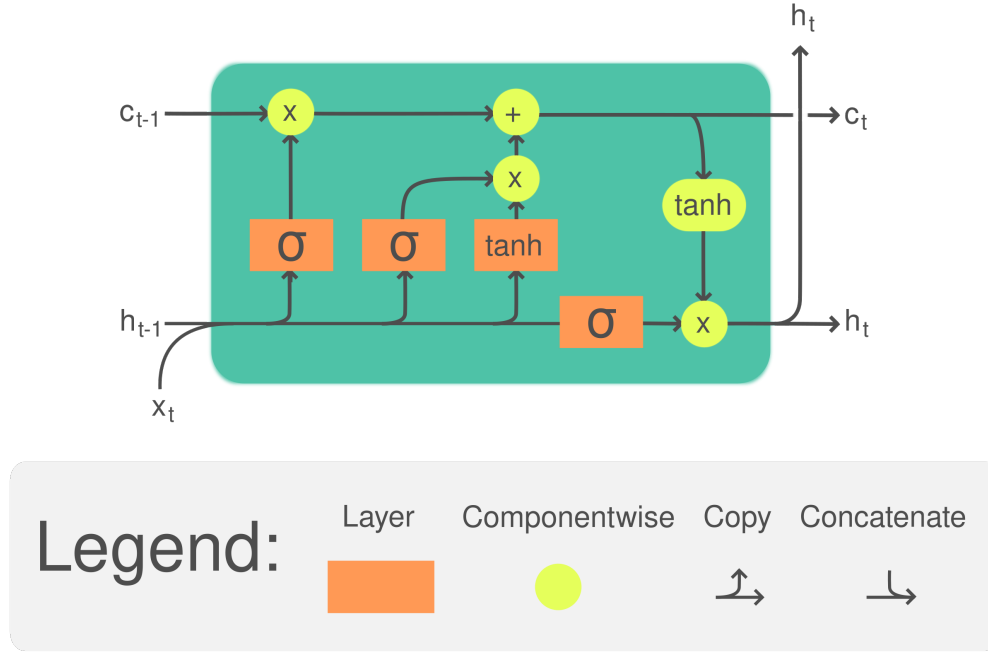


Fig. 4.1 General structure of an LSTM unit.

where $\mathbf{x}_t$ is the input vector, $\mathbf{h}_{t-1}$ the hidden state from the previous time step, $\sigma(\cdot)$ the sigmoid function, and $\odot$ the element-wise product. The matrices $\mathbf{W}_*$, $\mathbf{U}_*$, and bias vectors $\mathbf{b}_*$ are learnable parameters.

The forget gate $\mathbf{f}_t$ controls the retention of past information, while the input gate $\mathbf{i}_t$ decides how much of the candidate memory $\tilde{\mathbf{c}}_t$ should be incorporated. The output gate $\mathbf{o}_t$ determines which parts of the memory state are exposed to the hidden state $\mathbf{h}_t$, enabling the network to produce predictions.

Slow signals such as temperature variations exhibit gradual changes and long-range temporal correlations. The memory structure of LSTMs allows the network to learn degradation patterns and anomalous trends that develop across extended time windows, which are essential for predictive maintenance. The model can be trained to predict the next temperature value in a sequence, thus providing short-term forecasts that serve as early indicators of equipment malfunction.

LSTMs are trained by minimizing a regression loss function, typically the Mean Squared Error (MSE):

$$\mathcal{L} = \frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2, \quad (4.7)$$

where y_t is the observed temperature and $\hat{y}_t$ the predicted value at time t . Optimization is carried out using Backpropagation Through Time (BPTT), which adapts the gating parameters to improve prediction accuracy.

The advantages of LSTMs in predictive maintenance include their robustness against vanishing gradients, their ability to integrate multiple sensor inputs, and their effectiveness in providing early warnings by recognizing subtle, long-term changes in system behavior.

4.2 Dataset and Preprocessing

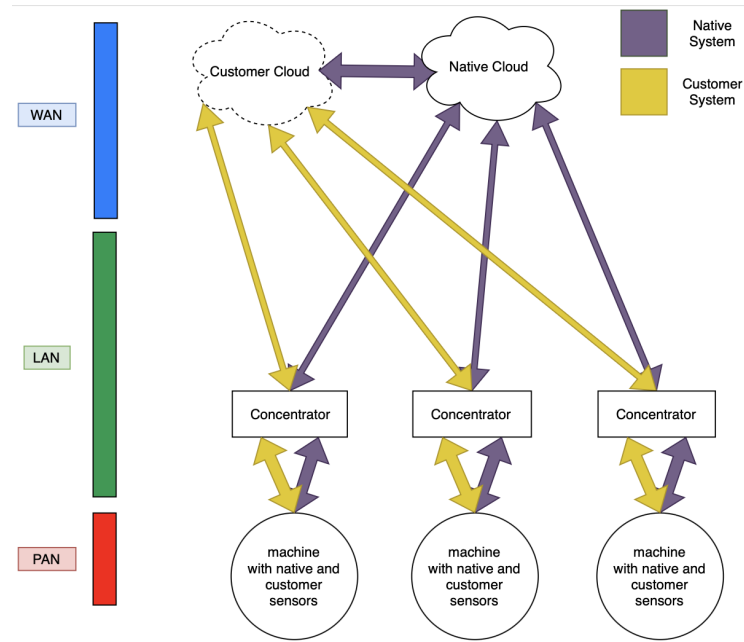


Fig. 4.2 Fog Computing layers

The experimental dataset was collected within an industrial fog computing framework previously introduced in [22, 111]. The architecture consists of three main layers:

1. The capping machine (PAN);

2. The concentrator (LAN);
3. The external users (WAN);

as illustrated in Figure 4.2. The concentrator acts as the central hub, managing data aggregation, elaboration, and user-defined applications deployed in Docker containers [113, 114].

Data were acquired using Nicla Sense ME devices, which integrate four Bosch Sensortec sensors capable of recording motion and environmental parameters. For this study, 50 nodes transmitted temperature, humidity, and pressure data via Bluetooth Low Energy (BLE) 4.2 to the concentrator, equipped with a EdgeBox RPi-200, described above.

Two experimental phases were carried out:

1. A laboratory test with all 50 nodes monitoring room temperature under controlled conditions;
2. An industrial deployment on an AROL Closure System machine, including evaluation within a thermal chamber for stress testing.

The data pipeline (Figure 4.3) was implemented with two containers:

- **Native container:** developed in Node.js (v16), responsible for acquiring raw data from sensors and publishing them to a TCP server. Each record contained a timestamp, device MAC address, and measured values.
- **Host container (LSTM):** developed in Python 3.11 with TensorFlow 2.15.0, NumPy, Pandas, and Sci-kit Learn. This container parsed the raw files, eliminated missing or invalid values, filtered outliers (outside 20–43°C), and applied node-specific MinMax normalization to map sensor values to [0,1].

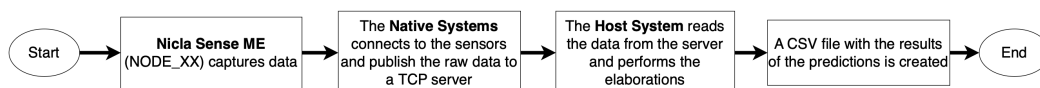


Fig. 4.3 Flowchart of the experiment

To model temporal dependencies, sequences of $n_{steps} = 20$ historical values were used to predict the subsequent temperature. Formally:

$$X_t = [x_t, x_{t+1}, \dots, x_{t+n_{steps}-1}], \quad (4.8)$$

$$y_t = x_{t+n_{steps}}. \quad (4.9)$$

The final dataset included 1,019,663 measurements, partitioned into training (80%), validation (10%), and test (10%). All preprocessing steps were consistently applied across splits, and the concentrator was programmed to automatically reconnect in case of communication failures, minimizing data loss.

The LSTM container deployed on the EdgeBox RPi-200 concentrator serves as the computational core for forecasting tasks. Its design emphasizes accuracy, computational efficiency, and modularity.

The host container is responsible for:

- Receiving preprocessed data from the native container via TCP.
- Maintaining sliding windows of 20 historical values.
- Applying node-specific normalization.
- Loading the trained LSTM model.
- Producing real-time forecasts for the next temperature measurement.

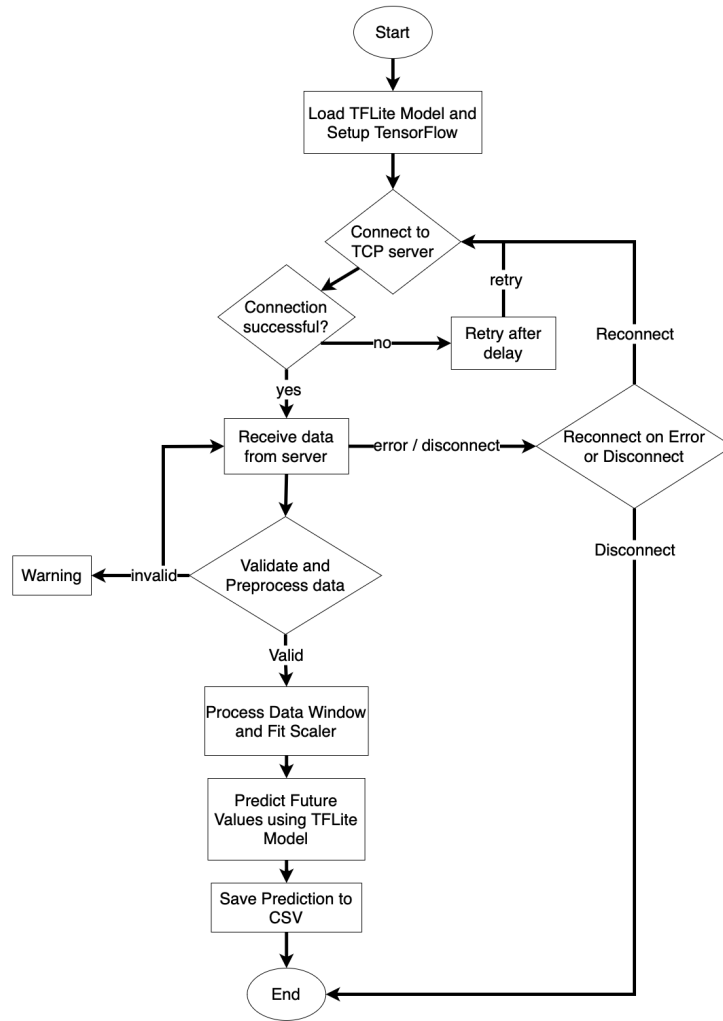


Fig. 4.4 Flowchart of the code

4.3 Design of the LSTM model

The implemented LSTM network consists of:

- Input layer with shape $(20, n_{features})$.
- LSTM layer with 50 units.
- Dense hidden layer with 25 ReLU neurons.
- Output layer with a single linear neuron.

This configuration achieved an RMSE of 0.1838 on the test set, with average inference latency of 200 ms per step.

Training was carried out on a high-performance server with the following settings:

- Optimizer: Adam.
- Loss function: RMSE.
- Batch size: 30.
- Epochs: 20, with early stopping (patience = 5).

The final model was exported in TensorFlow SavedModel format and converted to TensorFlow Lite for efficient deployment.

The container communicates with the native container for data input, external WAN users for prediction retrieval, and the embedded scheduler for resource management. This modular design ensures adaptability to various industrial setups and robustness for real-time predictive maintenance.

4.4 Training and Validation Strategy

The training and validation strategy was designed to guarantee generalization while preventing overfitting.

The dataset was divided into training (80%), validation (10%), and test (10%) subsets. Node-specific MinMax scaling was applied to account for sensor heterogeneity.

The model minimized RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}, \quad (4.10)$$

with training conducted using batch size 30, up to 20 epochs, and early stopping after 5 consecutive non-improving epochs.

Performance was evaluated using:

- Root Mean Squared Error (RMSE),
- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (4.11)$$

- Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{y_i}, \quad (4.12)$$

- Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (4.13)$$

During deployment, the LSTM container continuously:

1. Receives new temperature values from the native container.
2. Applies preprocessing and normalization.
3. Updates the sliding window.
4. Generates the next-step forecast.
5. Logs predictions and raises anomaly alerts if thresholds are exceeded.

The final model achieved an RMSE of 0.1838 on the test set, with inference latency around 200 ms. MAE, MAPE, and R^2 confirmed reliable predictive performance, demonstrating feasibility for industrial deployment on resource-constrained edge devices.

4.5 Results and Comparative Analysis

This section presents the results obtained from both laboratory and real-world industrial environments, followed by a comparative evaluation against existing state-of-the-art approaches. The analysis focuses on the system's predictive accuracy, scalability, resource consumption, and anomaly detection capabilities.

4.5.1 Laboratory Results

In the first phase, the system was tested in a controlled laboratory environment to assess predictive performance and anomaly detection. One sensor node (NODE_25) was deliberately altered in software to simulate anomalous behavior by producing irregular temperature fluctuations. This allowed evaluation of the LSTM container's ability to identify deviations without prior knowledge of which node was modified.

The predictive accuracy across all nodes was consistently high, with minimal error metrics. For example, NODE_36 reported a MAE of 0.0126 and RMSE of 0.0156, corresponding to an overall accuracy of 99.81%. Even in the case of NODE_25, where anomalies were intentionally introduced, the system detected significant deviations (MAE = 2.3526, RMSE = 11.5418, Accuracy $\approx$ 45%), confirming its robustness in identifying abnormal sensor behavior. Figure 4.5 illustrates the error distribution, highlighting the system's sensitivity to deviations.

In terms of computational efficiency, Table 4.1 reports the system's resource consumption. As the number of connected nodes increased from 1 to 50, both memory and CPU usage remained within manageable limits. For instance, when running Docker Compose with 50 nodes, the system consumed 1.4 GB of memory and 55% CPU usage, demonstrating its scalability for multi-sensor environments.

Table 4.1 LSTM experiment resource consumption.

Configuration	Nodes	Memory	CPU
None	0	570 MB	0%
Only Native	1	671 MB	10%
Only Native	50	700 MB	17%
Only Host	1	800 MB	10%
Only Host	50	1 GB	20%
Docker Compose	1	1.2 GB	35%
Docker Compose	50	1.4 GB	55%

Additionally, transmission efficiency was evaluated. When scaling from a single node (60 detections per minute) to 50 nodes, the detection rate per node decreased to 44 detections per minute, corresponding to a 26.53% reduction. Despite this drop,

the overall performance remained acceptable, highlighting the system's ability to sustain real-time monitoring in multi-node deployments.

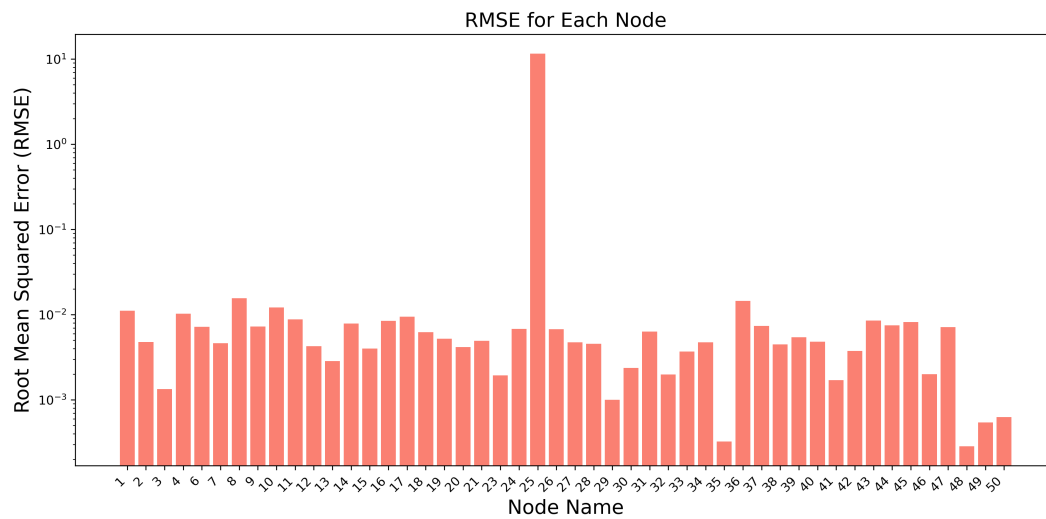


Fig. 4.5 MSE Results from Laboratory setting

4.5.2 Real-World Industrial Results

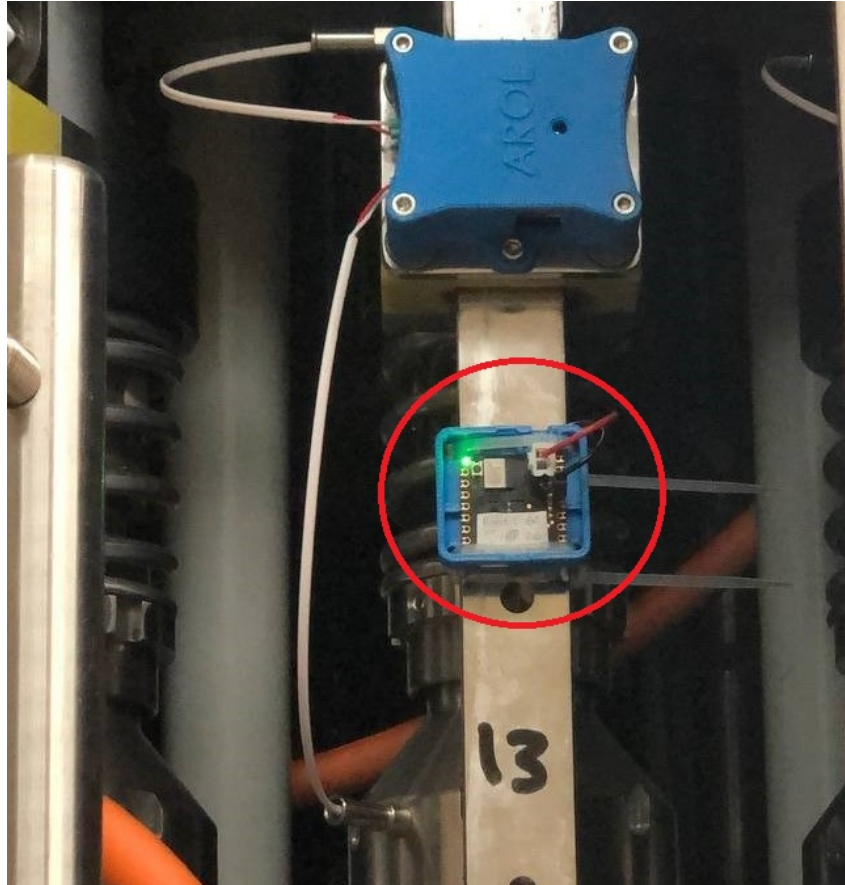


Fig. 4.6 Image of the sensor NODE_13 mounted on the machine, head 13

The second phase evaluated the system in a real industrial setting by deploying the sensors on a capping machine. Unlike the laboratory setup, no intentional anomalies were introduced. The results confirmed the consistency of the model's predictions across all nodes. The most significant deviation was observed for NODE_08, which achieved a MAE of 0.1821, RMSE of 0.288, R^2 of 0.8898, and MAPE of 0.7227%. The overall prediction accuracy remained above 99.2%, with deviations within ± 1 °C.

Although the errors in the industrial environment were higher compared to laboratory results, this outcome is attributed to environmental noise, electromagnetic interference, and metallic structures typical of real-world factory settings. Nevertheless, the predictive performance remained within acceptable operational thresholds,

demonstrating the robustness of the proposed architecture for practical industrial deployment.

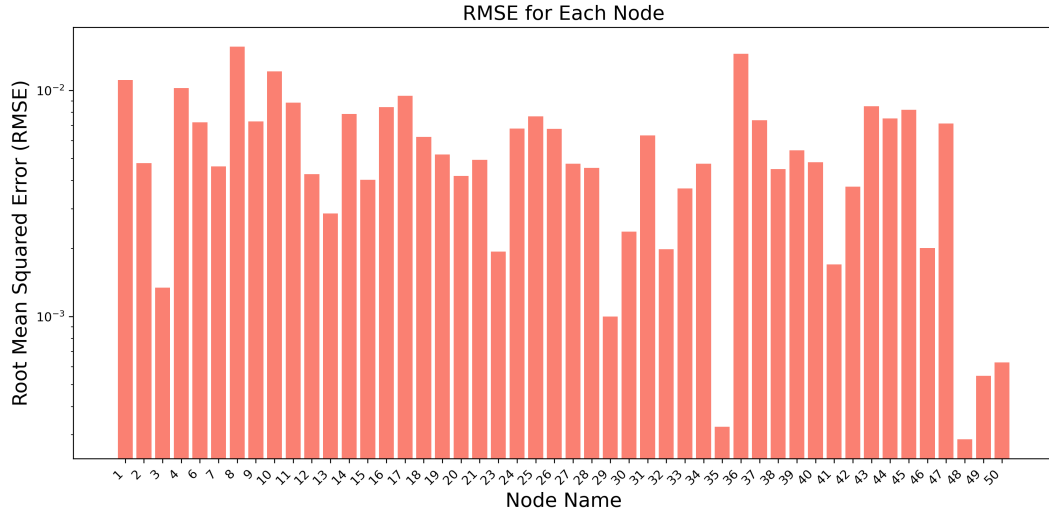


Fig. 4.7 MSE Results from Industrial setting

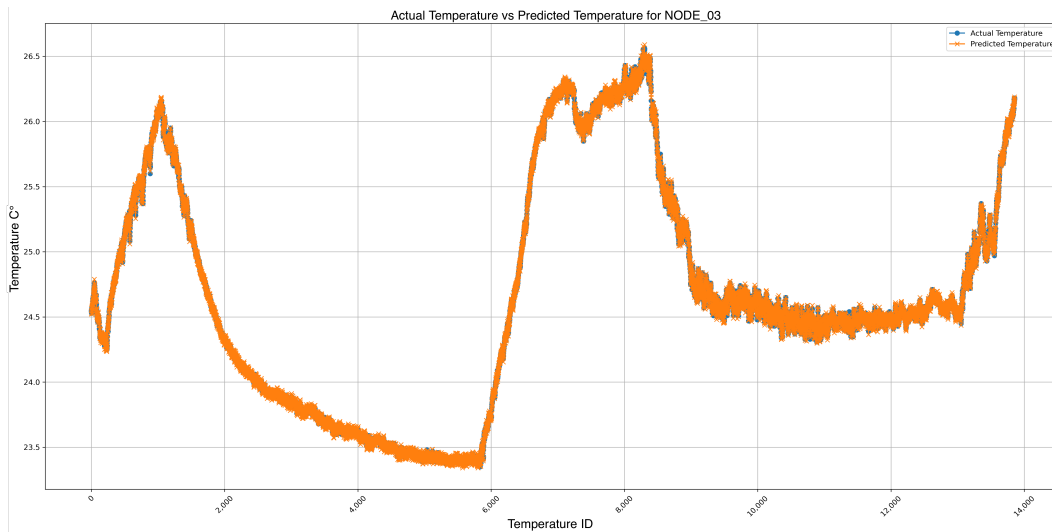


Fig. 4.8 Temperature prediction results from the Industrial setting

4.5.3 Comparison with Existing Approaches

The predictive performance of the LSTM-TFLite model was benchmarked against existing IoT predictive maintenance models, including FDPR (DC-MLP) [115],

FAQMP (Seq2Seq GRU Attention-TF Lite) [116], and DeepFogAQ (Transformers) [117]. Table 4.2 summarizes the results.

Table 4.2 Comparison of predictive performance with existing approaches.

Model	MAE	RMSE	MAPE	R^2	Inf. Time (ms)
Concentrator (LSTM-TF Lite)	0.1821	0.28	0.7	0.8898	200
FDPR (DC-MLP)	0.06	0.28	0.002	-	0.05
FAQMP (Seq2Seq GRU)	4.385	7.9016	22.01	0.6566	52 671.8
DeepFogAQ (Transf.)	3.15	3.79	-	-	-

The comparison reveals several key insights:

- FDPR (DC-MLP) achieves lower MAE and MAPE values, but at the cost of reduced model complexity and significantly less inference latency (0.05 ms vs 200 ms).
- The LSTM-TF Lite model outperforms FAQMP and DeepFogAQ across all error metrics, highlighting its superior accuracy and efficiency for resource-constrained IoT edge environments.
- Although inference time is longer compared to FDPR, the proposed model remains well-suited for real-time applications where millisecond-level latency is not critical.

Overall, the results demonstrate that the LSTM-TF Lite implementation offers a competitive balance between accuracy, efficiency, and scalability, establishing itself as a practical solution for predictive maintenance in industrial IoT deployments.

4.6 Remarks on the results

This chapter presented the design, implementation, and evaluation of an LSTM-based predictive maintenance system integrated into an edge computing framework. By leveraging slow signals such as temperature variations, the system demonstrated

its ability to provide accurate forecasts, anomaly detection, and scalability across heterogeneous IoT environments.

The key contributions can be summarized as follows:

- **Scalable edge deployment:** the Docker-based architecture enabled efficient resource management while supporting up to 50 nodes with limited performance degradation.
- **Accurate forecasting:** the LSTM-TFLite model achieved an RMSE of 0.1838 in laboratory conditions and 0.288 in industrial deployment, with accuracy consistently exceeding 99%.
- **Anomaly detection:** intentional modifications to NODE_25 validated the model's capacity to detect irregular sensor behavior and support proactive maintenance strategies.
- **Competitive benchmarking:** comparative analysis demonstrated that the LSTM-TFLite model outperforms more complex architectures such as Seq2Seq GRU Attention and Transformers in terms of error metrics and inference efficiency.

In conclusion, the proposed LSTM-TFLite container demonstrates the feasibility of deploying advanced machine learning algorithms on resource-constrained embedded systems for predictive maintenance. By ensuring accurate, reliable, and scalable operation, the system provides a strong foundation for the next generation of smart industrial IoT infrastructures.

4.7 Limitations and Scope of the Approach

While the experimental results presented in this chapter demonstrate the effectiveness of the proposed LSTM-based forecasting pipeline, it is important to critically analyze the scope and methodological boundaries of this approach.

From a strictly analytical perspective, the use of a Long Short-Term Memory (LSTM) network to predict temperature values at a one-second resolution could be considered over-engineered. Temperature signals in industrial environments typically

exhibit slow dynamics and high temporal correlation; in many practical scenarios, they can be effectively modeled through simpler techniques such as linear regression, ARIMA models, or basic exponential smoothing. For this specific task, a lightweight statistical model would likely achieve comparable accuracy with significantly lower computational cost.

However, the primary objective of this chapter is not to propose the LSTM as the unique optimal algorithm for temperature prediction. Instead, the goal is to validate the Data Concentrator architecture by deploying a computationally demanding machine learning model directly at the edge. The LSTM was deliberately selected as a representative "stress test" for the system, requiring non-trivial memory management, sequential processing, and real-time inference capabilities. In this context, the true contribution of this work lies in demonstrating that:

- the proposed containerized architecture is flexible enough to host advanced deep learning models;
- the Data Concentrator can perform real-time inference on embedded hardware (EdgeBox RPi-200) despite limited resources;
- TensorFlow Lite models can be effectively integrated within a production-ready industrial framework;
- the system maintains acceptable latency and stability even during resource-intensive tasks.

The LSTM model therefore serves as a technological benchmark. The achievement of accurate predictions is a positive secondary outcome, but the central result is the proven feasibility of running heavy analytics on a Raspberry-Pi-class platform.

Another fundamental limitation concerns the physical nature of the monitored signal. Temperature is often an indirect or "delayed" indicator of machine health; by the time a significant thermal increase is detected, the mechanical failure may already be at an advanced stage. This single-variable approach is sufficient for architectural validation but lacks the depth required for a comprehensive health index.

This limitation defines the natural evolution of the research. Having established that the infrastructure can handle complex computational workloads, the focus must shift toward more informative, high-frequency signals. In Chapter 5, this

same architecture will be exploited to analyze vibration and acceleration data. In that upcoming scenario, the adoption of sophisticated machine learning techniques transitions from being a mere architectural benchmark to a functional necessity, as the complexity of high-frequency dynamics far exceeds the capabilities of simple linear models.

Chapter 5

Predictive Maintenance through Fast Signals

Predictive maintenance strategies increasingly rely on the analysis of high-frequency vibration signals, which provide detailed information about the mechanical integrity of industrial equipment. Unlike slow signals such as temperature, vibration data capture transient and rapid dynamics associated with mechanical wear, imbalance, or misalignment, offering a complementary and highly sensitive diagnostic channel. In this chapter, we present the design, implementation, and evaluation of a containerized anomaly detection system for vibration-based predictive maintenance, specifically targeting industrial capping machinery.

The methodology combines classical signal processing with advanced machine learning techniques, enabling the detection of both gradual (drift-in-time) and transient anomalies. A hybrid architecture integrates diagnostic algorithms for temporal window analysis with punctual anomaly detection models, including Long Short-Term Memory (LSTM) networks, Autoencoders, and Isolation Forests. By embedding these techniques within a Docker container framework, the system ensures scalability, modularity, and deployability in industrial edge computing environments.

The chapter is organized as follows. Section 5.1 reviews related work on vibration-based anomaly detection. Section 5.2 introduces vibration analysis for predictive maintenance. Section 5.3 describes the dataset and preprocessing steps. Section 5.4 presents the laboratory experiment, including the design of the vibra-

tion container and diagnostic models. Finally, Section 5.5 discusses the results and concludes the chapter with key observations and future directions.

5.1 Related Work on Vibration-Based Anomaly Detection

Vibration-based anomaly detection in rotating machinery is a well-established research domain. Nevertheless, traditional approaches often face limitations under non-stationary operating conditions and in the presence of data drift. Early studies predominantly relied on supervised learning with extensively labeled datasets. For instance, Ali et al. [118] employed Support Vector Machines (SVM) and Random Forests (RF) on a custom testbed, while Pavan Kumar et al. [119] proposed a neural network model integrated with kernel density estimation (KDE) on the Case Western Reserve University (CWRU) dataset. Although effective in controlled settings, these approaches remain constrained by their dependency on large volumes of high-quality labeled data, which are rarely available in industrial environments.

To address this issue, recent works have shifted toward self-supervised and hybrid learning paradigms. Lee and Kim [120] combined self-supervised feature learning with statistical descriptors, achieving improved generalization and enhanced compatibility with edge devices. Drift-resilient frameworks have also been developed, such as the approach of Seo et al. [121], which integrates spectral and statistical features within a dynamic rolling window, and the BERT-inspired Transformer model introduced by Zhou et al. [122], designed to process angle-domain signals under non-stationary conditions.

Novel neural architectures and alternative signal representations have likewise been investigated. Styanidis et al. [123] proposed a one-dimensional convolutional autoencoder (1D-CAE) for unsupervised anomaly detection, while Hu et al. [124] introduced vibration vectors—combining amplitude and phase information—to facilitate earlier fault detection. On the hardware side, edge-executable implementations have gained increasing relevance, with Pavan Kumar et al. [119] and Hsia and Hsu [125] presenting FPGA-based solutions capable of resource-efficient, real-time analysis.

Handling variability in operating conditions continues to represent a central challenge. Shi et al. [126] addressed this by proposing an adversarial knowledge distillation framework designed to decouple machine health state from operational variability. Collectively, these contributions emphasize the importance of developing self-supervised, adaptive, and lightweight approaches that remain robust under drift-prone environments and computationally constrained edge deployments.

In this context, the present work introduces a novel two-stage, fog-computing-based anomaly detection framework. Unlike previous monolithic or cloud-dependent approaches, the proposed system integrates rule-based detection with an ensemble of machine learning models (LSTM forecaster, autoencoder, and Isolation Forest) encapsulated as containerized microservices. This modular architecture enables real-time inference on resource-constrained edge devices (e.g., EdgeBox RPi-200), addresses both data drift and sensor heterogeneity, and provides a practical, extensible solution for predictive maintenance in industrial bottling lines.

Tables 5.1 and 5.2 provide a comparative summary of existing contributions. Table 5.1 highlights the methods, labeling requirements, and application domains, while Table 5.2 compares architecture types, drift-handling capabilities, and deployability. In contrast to existing approaches, the proposed *Concentrator* system is modular, drift-resilient, fully edge-executable, and validated in a real industrial environment.

5.2 Vibration Analysis for Predictive Maintenance

Vibration analysis represents a cornerstone of condition monitoring in rotating machinery. While healthy machines generate stable vibration signatures, faults such as bushing wear, gear loosening, imbalance, or misalignment introduce distinctive anomalies in the signal's amplitude, frequency distribution, or temporal evolution.

Initial attempts focused on predictive modeling approaches, such as Long Short-Term Memory (LSTM) networks and Transformer architectures, aimed at forecasting acceleration values. However, this strategy proved impractical due to the high arrival rate and variability of the vibration data. Unlike temperature signals, accelerations often exhibit sudden spikes or instantaneous fluctuations that are not necessarily linked to actual fault conditions, thereby limiting the feasibility of direct forecasting.

Table 5.1 Methods and Data Requirements

Ref.	Method	Labels	Domain
[118]	SVM, RF (Supervised)	High	Custom testbed
[119]	Neural Network + KDE	High	CWRU dataset
[120]	Self-supervised + Statistical features	Low	Multiple public datasets
[121]	Drift-resilient self-supervised	None	NSA + Real-World data
[122]	Transformer with domain transformation	Low	Custom industrial data
[123]	1D Convolutional Autoencoder	None	Radar machinery
[124]	SVDD with vibration vectors	Medium	Simulated data
[125]	FPGA-based FFT + control system	None	Real-time industrial setup
[126]	Adversarial decoupling network	Low	Variable speed conditions
Concentrator	Hybrid ensemble (LSTM, AE, IF)	None	Real industrial bottling line

Table 5.2 Architecture and Deployment Characteristics

Ref.	Drift Handling	Edge Executable	Architecture Type
[118]	No	No	Monolithic
[119]	No	Yes	Monolithic
[120]	Partial	Yes	Monolithic
[121]	Yes	No	Modular
[122]	Yes	No	Monolithic
[123]	No	No	Monolithic
[124]	No	No	Monolithic
[125]	No	Yes	Hardware integrated
[126]	Yes	No	Monolithic
Concentrator	Yes	Yes	Modular

Subsequent efforts explored the use of Power Spectral Density (PSD) analysis, either for predictive purposes or for detailed signal characterization. Although this approach enabled the development of algorithms capable of accurate spectral predictions, it was unsuitable for real-world deployment. Specifically, such methods required access to the initial portion of the PSD curve, which is inherently unavailable in practical online monitoring scenarios.

For these reasons, the adopted methodology focuses on real-time analysis of acceleration signals across three complementary domains:

- **Time domain:** statistical descriptors such as root mean square (RMS), skewness, and kurtosis quantify energy distribution and impulsiveness.
- **Frequency domain:** Fourier-based methods, including Power Spectral Density (PSD), reveal frequency components associated with bearings, gears, and rotating shafts.
- **Time–frequency domain:** Short-Time Fourier Transform (STFT), wavelet transforms, and Hilbert–Huang Transform capture non-stationary effects and transient phenomena.

Anomalies are detected by comparing raw vibration signals against reference baselines. In the proposed approach, an ensemble of diagnostic algorithms ensures robust anomaly detection, further supported by statistical stationarity tests. Once anomalous temporal windows are identified, they are subjected to advanced deep learning and machine learning models for precise fault localization.

5.3 Dataset and Preprocessing

The experimental campaign was conducted in collaboration with AROL Closure Systems, using operational capping machinery under realistic production conditions [127].



Fig. 5.1 Capping Machine sensorized with ISM330DHCX IMU

5.4 Laboratory Experiment

5.4.1 Data Acquisition

Vibration signals were collected with an ISM330DHCX inertial measurement unit (IMU) , in Fig. 5.1, mounted directly on the machine's capping head or fork. The triaxial accelerometer was configured to a full-scale range of $\pm 16g$ with a nominal output data rate of 6667 Hz. Only accelerometric data were used, and the magnitude of the acceleration vector was computed.

Two sets of experiments simulated progressive mechanical degradation:

1. Bushing wear (in Fig. 5.2: nominal, slightly worn, and worn conditions.
2. Spring loosening: nominal, slight loosening, and loosened conditions.

Table 5.3 summarizes the experimental setup. Each acquisition produced over 800,000 samples, corresponding to about 64 machine cycles under steady-state conditions.



Fig. 5.2 Slightly worn bush

Table 5.3 Experimental test conditions and acquisition parameters.

ID	Component	Fault	Diameter [mm]	Duration [s]
B1	Bushing	Nominal	70.220	154.0
B2	Bushing	Slight Wear	70.320	154.0
B3	Bushing	Worn	70.530	154.0
S1	Spring	Nominal	70.120	125.8
S2	Spring	Slight Loosening	75.000	144.4
S3	Spring	Loosened	75.000	123.1

5.4.2 Preprocessing and Calibration

Unlike the slow-signal approach, no global filtering or normalization was applied to the vibration signals, in order to preserve their raw structure. For specific visualization tasks, local normalization was employed to enhance interpretability of individual segments.

The signals were divided into windows corresponding to machine cycles (2.4 seconds per cycle at 25 rpm). This segmentation enabled consistent comparisons across operating conditions while maintaining temporal coherence. Prior to anomaly detection, each detection model was calibrated using a dedicated set of normal operation data, allowing the system to establish thresholds for statistical, spectral, and predictive features. This calibration step ensures that subsequent anomaly

assessments are tailored to the machine's typical behavior and reduces false-positive detections.

5.4.3 Design of the Vibration Container and Models

The vibration container was developed as a modular pipeline comprising two phases:

1. **Diagnostic phase (temporal window analysis):** an ensemble of signal processing and statistical methods identifies anomalous temporal windows compared to a healthy baseline. Each method focuses on complementary aspects of the vibration signal, as summarized in Table 5.4.
2. **Punctual anomaly detection:** machine learning models (LSTM, Autoencoder, Isolation Forest) are applied within anomalous windows to identify precise fault instances. Their outputs are aggregated through majority voting, ensuring robustness.

5.4.4 Diagnostic Methods

To analyze the acceleration signals, state-of-the-art methods that combine both time and frequency information were first employed. Subsequently, machine learning algorithms were applied to enable the detection of punctual anomalies.

The so-called classical methods include:

Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is a well-established technique for measuring the similarity between two temporal sequences that may vary in length or speed. Unlike the Euclidean distance, which assumes a strict one-to-one alignment of time indices, DTW allows nonlinear temporal stretching and compression, thus enabling the comparison of signals that exhibit phase shifts or non-uniform sampling.

Formally, given two sequences $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_m)$, DTW computes the minimal cumulative distance over all possible alignment paths $p \in P$:

$$DTW(X, Y) = \min_{p \in P} \sum_{(i,j) \in p} d(x_i, y_j), \quad (5.1)$$

where $d(x_i, y_j)$ is a local distance metric, typically the squared Euclidean distance.

In the context of vibration analysis, DTW is particularly useful for detecting slow drifts and temporal distortions in machine dynamics, since it can align sequences even when operating cycles do not perfectly match. However, DTW is computationally intensive for long signals and its performance deteriorates in the presence of strong high-frequency noise. Thus, it is primarily suited for capturing long-term structural changes rather than transient impulsive events.

Short-Time Fourier Transform (STFT)

The Short-Time Fourier Transform (STFT) provides a joint representation of a signal in the time–frequency domain. It is defined by applying the Fourier transform over successive, overlapping windows of the signal:

$$STFT\{x\}(t, \omega) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j\omega\tau}d\tau, \quad (5.2)$$

where $w(\tau)$ is a window function centered at time t .

STFT is especially effective in analyzing non-stationary vibration signals, where spectral content varies with time. For example, faults in rotating machinery often manifest as transient frequency bursts or modulated harmonics that cannot be identified through stationary Fourier analysis. The main drawback lies in the intrinsic trade-off between temporal and spectral resolution, determined by the choice of window size: narrower windows capture rapid transients at the expense of frequency precision, while longer windows improve frequency resolution but obscure fast temporal dynamics.

Power Spectral Density (PSD)

The Power Spectral Density (PSD) quantifies how the signal's power is distributed across frequencies, thereby offering a frequency-domain fingerprint of machine behavior. For a stationary signal $x(t)$, PSD can be expressed as:

$$S_{xx}(\omega) = \lim_{T \rightarrow \infty} \frac{1}{2T} \mathbb{E} \left[|X_T(e^{j\omega})|^2 \right], \quad (5.3)$$

where X_T denotes the truncated Fourier transform of $x(t)$. In practice, PSD is typically estimated using Welch's method, which averages modified periodograms to reduce variance:

$$\hat{S}_{xx}(\omega) = \frac{1}{K} \sum_{k=1}^K \frac{1}{N} \left| \sum_{n=0}^{N-1} w[n] x_k[n] e^{-j\omega n} \right|^2. \quad (5.4)$$

PSD is widely adopted in condition monitoring since mechanical defects often produce distinct fault-related frequencies. Its limitation, however, is the lack of temporal localization: PSD characterizes frequency content but does not indicate when anomalies occur, making it less suitable for detecting transient events.

Statistical Descriptors

Time-domain statistical descriptors provide a compact yet informative characterization of vibration signals. They are particularly valuable due to their simplicity and interpretability. Common descriptors include:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (\text{mean}), \quad (5.5)$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2 \quad (\text{variance}), \quad (5.6)$$

$$\gamma = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^3}{\sigma^3} \quad (\text{skewness}), \quad (5.7)$$

$$\kappa = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^4}{\sigma^4} \quad (\text{kurtosis}). \quad (5.8)$$

These descriptors allow one to quantify key properties such as energy distribution, impulsiveness, and signal asymmetry. For instance, kurtosis is highly sensitive to impulsive transients, while skewness captures asymmetries that may indicate incipient faults. Nevertheless, statistical features may fail to capture complex non-stationary dynamics or localized anomalies, thus limiting their diagnostic resolution.

Envelope RMS

Envelope analysis is a classical technique in vibration diagnostics, particularly effective for detecting bearing defects and impulsive events. It relies on the Hilbert transform to extract the analytic signal, from which the envelope is computed:

$$x_H(t) = \mathcal{H}\{x(t)\}, \quad e(t) = \sqrt{x^2(t) + x_H^2(t)}, \quad (5.9)$$

and the RMS of the envelope is given by:

$$RMS_{env} = \sqrt{\frac{1}{N} \sum_{i=1}^N e^2(i)}. \quad (5.10)$$

This method highlights amplitude modulations induced by periodic impacts or faults in rolling elements. It is computationally efficient and highly sensitive to impulsive anomalies. However, it is less effective in capturing gradual drifts or structural changes that are not associated with high-energy impacts.

Augmented Dickey–Fuller (ADF) Test

The Augmented Dickey–Fuller (ADF) test is a statistical procedure used to determine whether a signal is stationary or exhibits a unit root. It is based on the following autoregressive model:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^p \delta_i \Delta y_{t-i} + \varepsilon_t, \quad (5.11)$$

where $\Delta y_t = y_t - y_{t-1}$, α is a constant, β captures deterministic trends, and ε_t is a white noise error term.

In vibration analysis, the ADF test is employed to detect slow drifts or regime transitions that render the signal non-stationary. Its advantage lies in providing statistical evidence of changes in signal properties. However, the method is sensitive to noise and parameter selection, and it does not provide temporal localization of anomalies. Consequently, it is most effective when combined with other diagnostic tools.

Table 5.4 provides an overview of the diagnostic techniques employed, describing what each method measures, the type of anomaly it can identify, its strengths, and its limitations.

Table 5.4 Summary of vibration analysis diagnostic methods

Method	Detects	Strengths	Limitations
PSD	Frequency energy	Highlights spectral anomalies	No temporal info
ADF	Stationarity	Detects drifts state changes	Sensitive to noise
Statistics	Signal variations	Lightweight interpretable	Limited for complex anomalies
Envelope RMS	Impulsive events	Good for shocks	Misses slow drifts
DTW (drift)	Long-term anomalies	Captures nonlinear trends	Sensitive to noise
STFT	Time-varying spectrum	Detects local changes	Limited time and freq resolution

The suitability of each method varies depending on the anomaly type. Table 5.5 maps diagnostic techniques to representative anomaly scenarios.

Table 5.5 Suitability of diagnostic methods for different anomaly types

Anomaly Type	PSD	ADF	Statistics	Envelope	DTW	STFT
Impact / spike	No	No	Yes	Best	No	Best
Sudden energy increase	Yes	No	Yes	Best	No	Best
Cyclic variation	Best	No	No	Yes	No	Best
Slow drift	No	Best	Yes	No	Yes	No
Stationarity change	No	Best	Yes	No	Yes	No
Band-specific anomaly	Best	No	No	No	No	Best
Slow morphological variation	No	Yes	Yes	No	Yes	No
Local non-energetic anomaly	No	No	No	No	No	Yes

5.4.5 Method Combinations

Since no single technique is sufficient to capture the full spectrum of possible anomalies, combining complementary methods provides more robust diagnostics. The following combinations proved particularly effective:

- **Envelope RMS + STFT:** optimal for detecting *transient anomalies*, as the envelope highlights impulsive events while STFT captures their spectral footprint.
- **PSD + ADF:** effective for identifying *drifts and slow trends*, combining spectral analysis with stationarity testing.
- **Statistics + DTW:** useful for monitoring *long-term nonlinear evolutions* of the signal.
- **Envelope RMS + DTW:** well-suited for mixed cases involving both impulsive shocks and slow deviations in signal shape.

The final diagnostic stage aggregates outputs from the individual methods using a voting mechanism, with a window flagged as anomalous if at least 60% of detectors agree. To mitigate false alarms due to noise, temporal filtering ensures that only consecutive anomalous windows of minimum length L are retained.

5.4.6 Training and Validation Strategy

The machine learning models were trained exclusively on vibration data collected under healthy operating conditions, with the objective of establishing reliable baselines for anomaly detection. The training phase was conducted on a dataset comprising 6,082,575 points, ensuring statistical robustness across multiple operational scenarios.

LSTM-Based Forecasting

The first approach employed a Long Short-Term Memory (LSTM) forecaster to predict the subsequent Root Mean Square (RMS) value within a given time series. Anomalies were identified when prediction errors, measured through Mean Squared Error (MSE) or Mean Absolute Error (MAE), significantly exceeded a calibrated baseline threshold.

The adopted architecture consisted of two stacked LSTM layers with 64 and 32 hidden units, respectively, followed by batch normalization to stabilize training dynamics, dropout layers to mitigate overfitting, and a fully connected dense layer for single-step prediction.

The training results are summarized as follows:

- **Training set:** $\text{MSE} = 0.0028$, $\text{MAE} = 0.0345$, $R^2 = 0.7404$
- **Validation set:** $\text{MSE} = 0.0028$, $\text{MAE} = 0.0346$, $R^2 = 0.7386$

These results confirm that the model achieves consistent predictive accuracy across both training and validation datasets, making it suitable for real-time anomaly detection.

LSTM Autoencoder

In parallel, an LSTM-based autoencoder was implemented to reconstruct input sequences. Anomalies were detected when the reconstruction error—quantified through MSE and MAE—exceeded a healthy baseline threshold.

The autoencoder architecture mirrored that of the forecaster, incorporating two LSTM layers (64 and 32 hidden units), batch normalization, dropout, and a dense output layer.

The model performance was as follows:

- **Training set:** MSE = 0.0, MAE = 0.024, $R^2 = 0.88$
- **Validation set:** MSE = 0.0, MAE = 0.03, $R^2 = 0.89$

Compared to the forecasting model, the autoencoder demonstrated superior reconstruction fidelity, particularly in capturing long-term temporal correlations within the vibration signals.

Isolation Forest

A complementary approach was based on the Isolation Forest algorithm, which assigns an anomaly score to each feature vector constructed from statistical descriptors (mean, variance, skewness, and kurtosis) computed over segmented windows. Anomalies correspond to samples that are easily isolated from the majority of the data distribution.

The model hyperparameters were optimized via grid search, yielding the following configuration:

- contamination = auto
- max_features = 0.5
- max_samples = auto
- n_estimators = 100
- random_state = 42

This configuration provided the most balanced trade-off between anomaly detection sensitivity and computational efficiency.

5.4.7 Deployment and Resource Profiling

To validate the proposed architecture, a semi-worn polymer bushing was installed on a capping machine (see Fig. 5.4 and Fig. 5.5), thereby simulating a progressive fault condition. The system was deployed using three Docker containers:

1. a TCP-based data collector,
2. an LSTM-based temperature forecaster,
3. and the vibration anomaly detector described above.

Resource profiling during real-time inference revealed a combined memory footprint of approximately 400 MB, an average CPU utilization of 27%, and an inference latency of 200 ms per prediction window. These values confirm the feasibility of deploying the system in resource-constrained industrial edge environments. In Fig. 5.3 is shown the flowchart of the experiment.

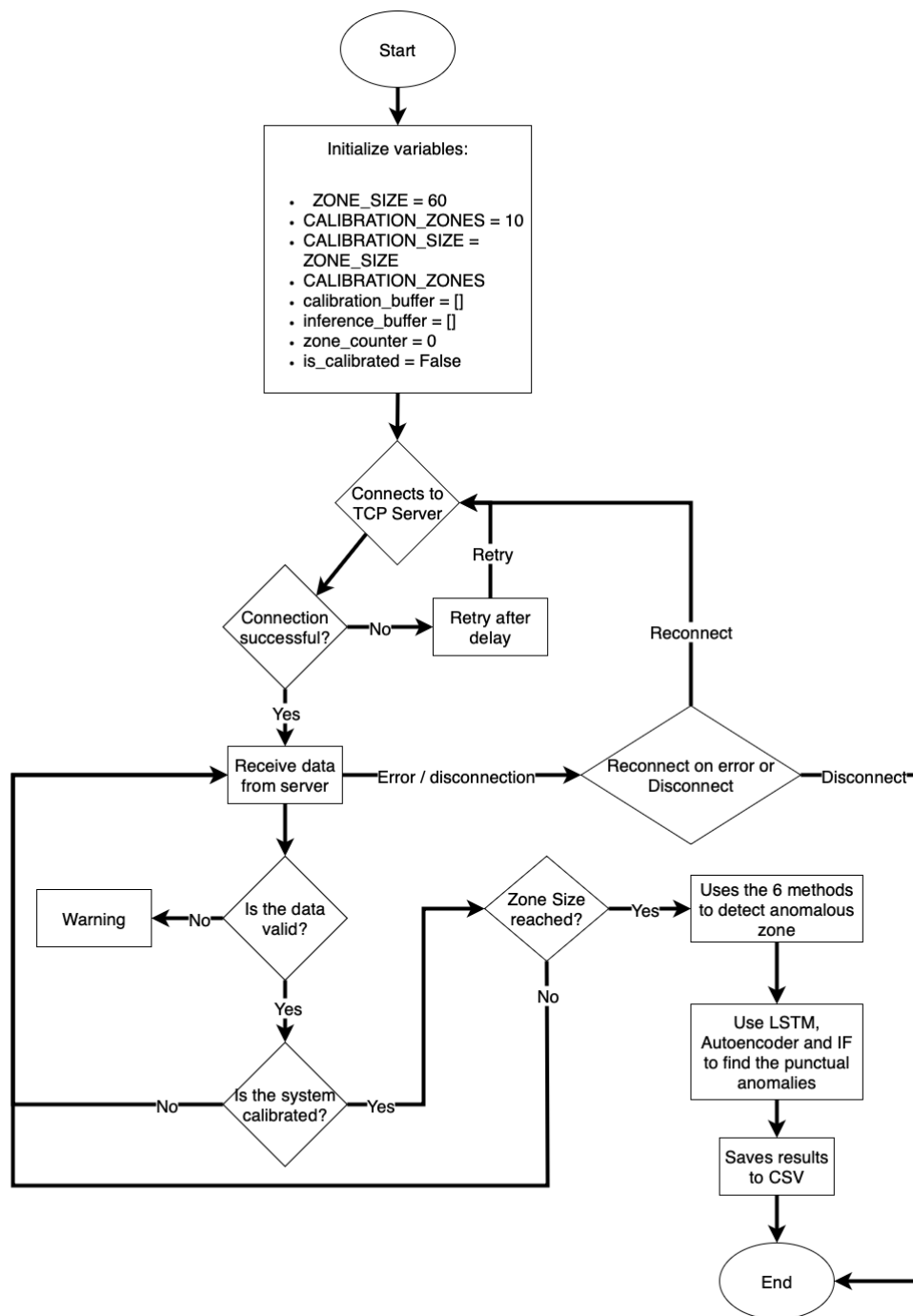


Fig. 5.3 Flowchart of the experiment

5.4.8 Results

Drift-in-Time Anomaly Detection — First Experiment

In the first experiment, the system was evaluated for its ability to detect drift-in-time anomalies associated with bushing wear. Table 5.6 summarizes the anomaly detection timing and classification performance metrics.

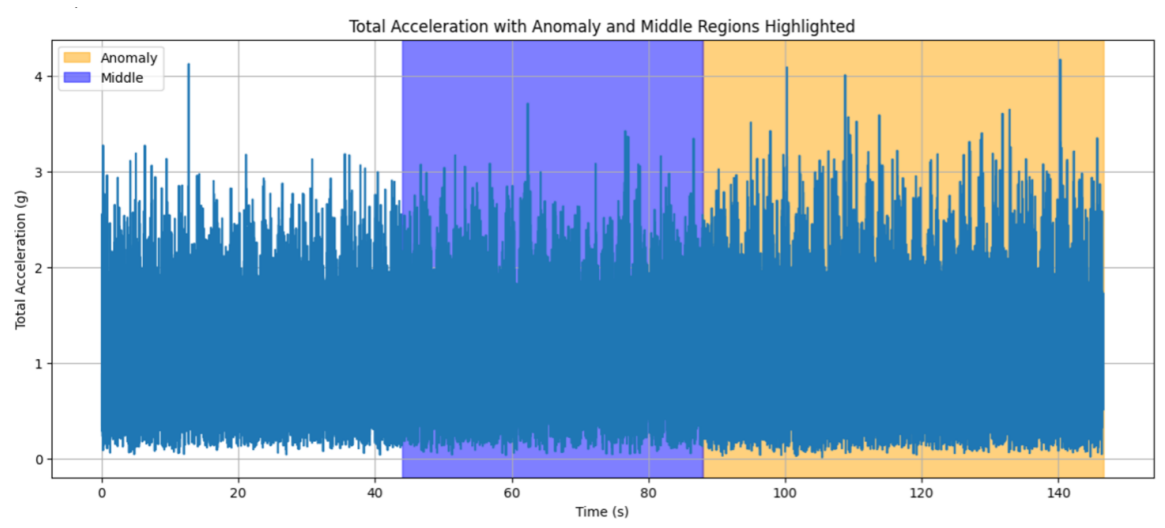


Fig. 5.4 First experiment, drift anomaly, band anomalies

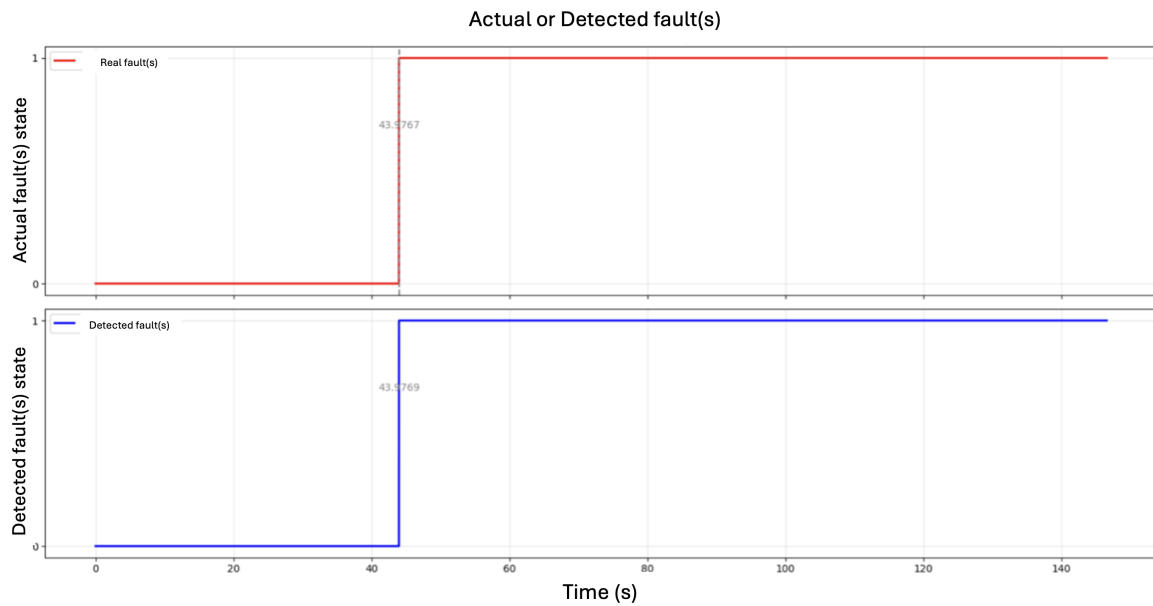


Fig. 5.5 First experiment, drift anomaly, point anomalies

Table 5.6 Anomaly Detection Timing and Performance Metrics — First Experiment

No.	Actual Anomaly (s)	Detection (s)	Difference (s)	
1	43.9767	43.9773	-0.0006	
Label	Precision	Recall	F1-Score	Support
Good	0.999	1.00	0.999	300,000
Anomaly	1.00	0.999	0.999	700,000
Accuracy	—	—	0.999	1,000,000
Macro avg	0.999	0.999	0.999	1,000,000
Weighted avg	0.999	0.999	0.999	1,000,000

The model achieved near-perfect classification, correctly identifying 299,996 out of 300,000 samples labeled as "Middle" (99.9986%) and all 400,000 samples in the "Anomaly" class (100%). Only four false negatives were observed, resulting in an overall error rate of 0.00057% across the anomalous dataset and 0.0013% specifically for the "Middle" anomaly category.

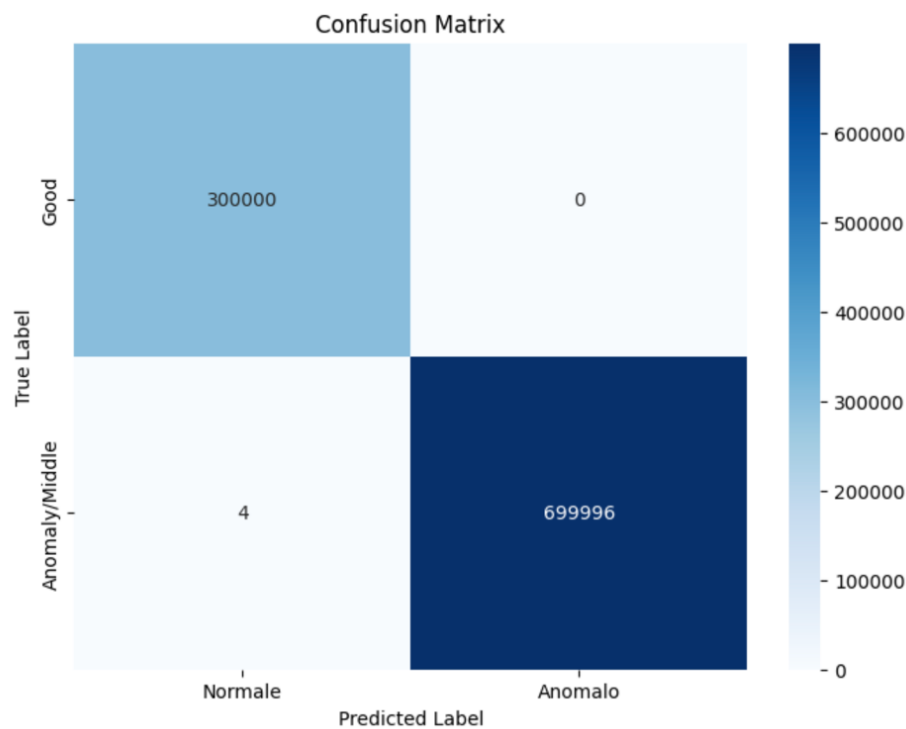


Fig. 5.6 First experiment, drift anomaly, confusion matrix

Drift-in-Time Anomaly Detection — Second Experiment

The second experiment assessed the system’s performance in detecting drift-in-time anomalies due to gear loosening. Detection timings and performance metrics are reported in Table 5.7.

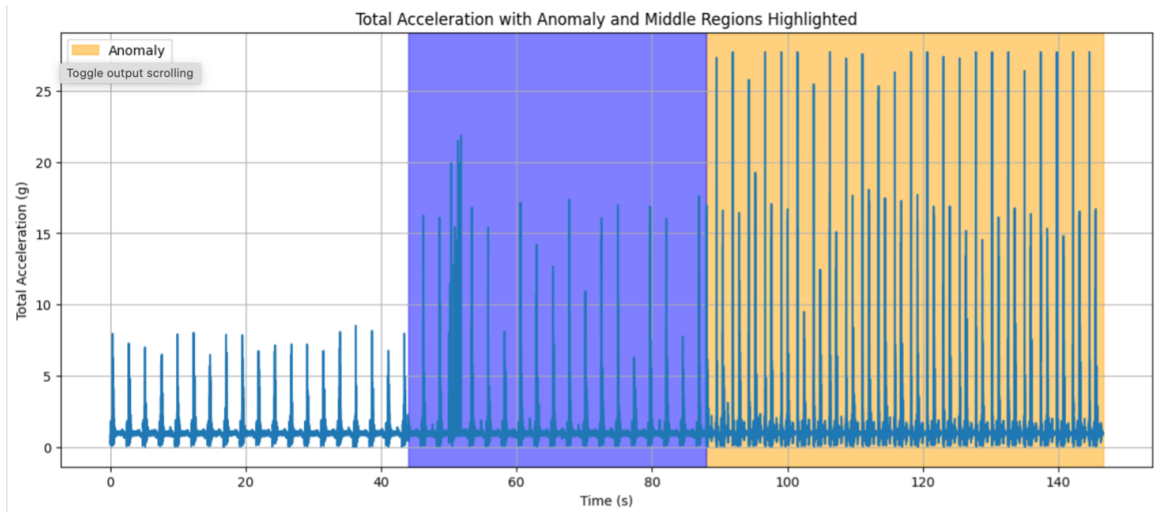


Fig. 5.7 Second experiment, drift anomaly, band anomalies

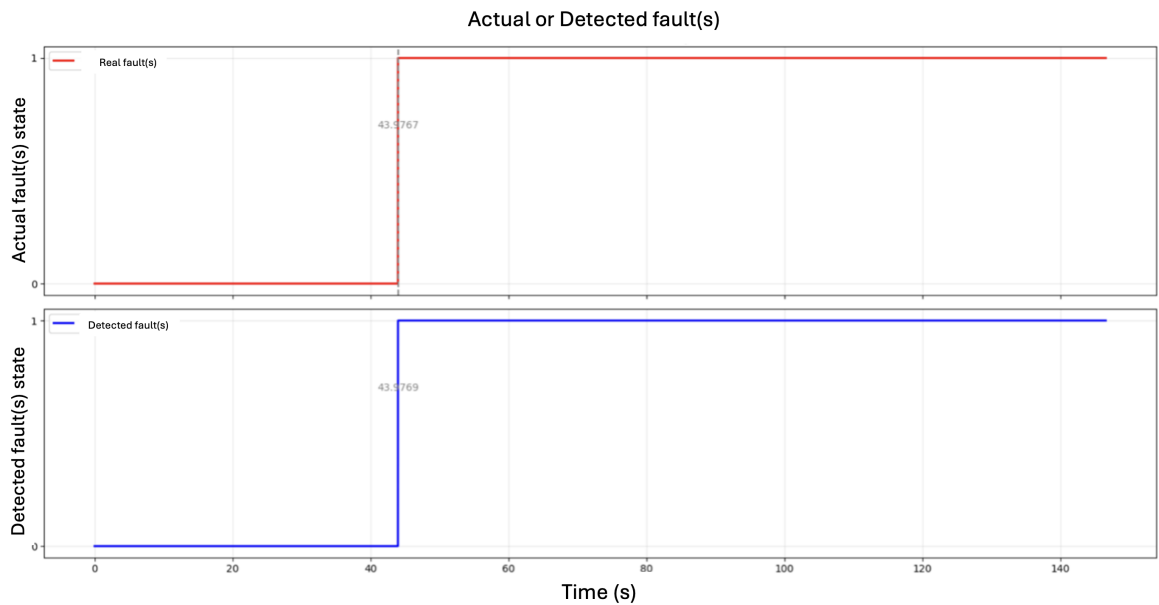


Fig. 5.8 Second experiment, drift anomaly, point anomalies

Classification results were consistent with the first experiment: 299,996 out of 300,000 "Middle" samples (99.9986%) and all 400,000 "Anomaly" samples (100%) were correctly classified, yielding four false negatives and error rates of 0.00057% overall and 0.0013% for the "Middle" category.

Table 5.7 Anomaly Detection Timing and Performance Metrics — Second Experiment

No.	Actual Anomaly (s)	Detection (s)	Difference (s)
1	43.9767	43.9773	-0.0006

Label	Precision	Recall	F1-Score	Support
Good	0.999	1.00	0.999	300,000
Anomaly	1.00	0.999	0.999	700,000
Accuracy	—	—	0.999	1,000,000
Macro avg	0.999	0.999	0.999	1,000,000
Weighted avg	0.999	0.999	0.999	1,000,000

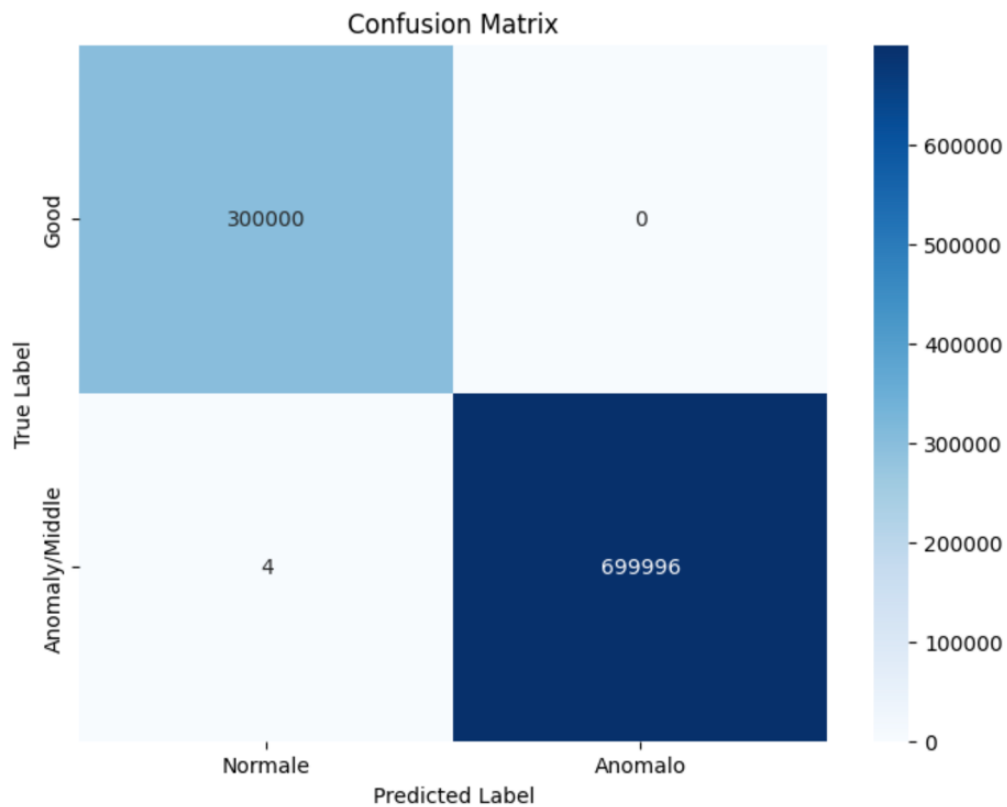


Fig. 5.9 Second experiment, drift anomaly, confusion matrix

Transient Anomaly Detection — First Experiment

The first transient anomaly experiment involved introducing brief anomalies within otherwise "Good" data. Table 5.8 presents the detection timing and classification metrics.

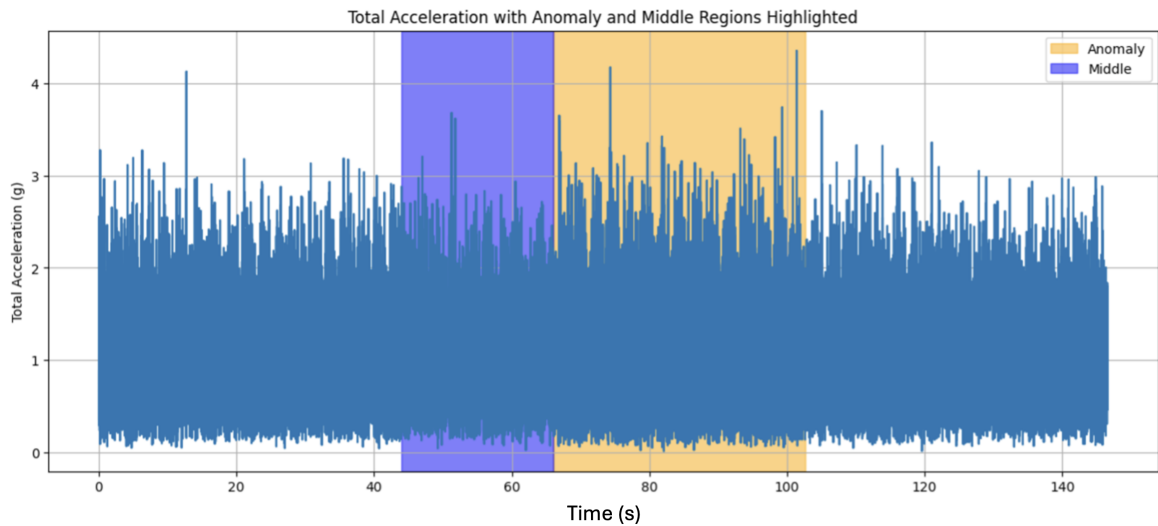


Fig. 5.10 First experiment, transient anomaly, band anomalies

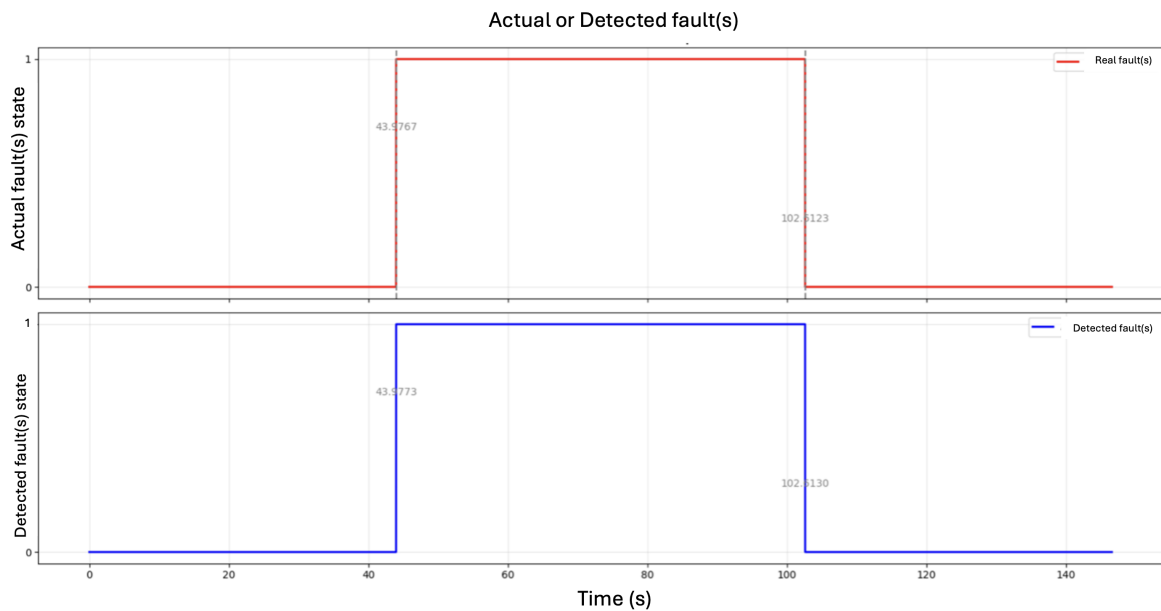


Fig. 5.11 First experiment, transient anomaly, point anomalies

Table 5.8 Anomaly Detection Timing and Performance Metrics — First Transient Experiment

No.	Actual Anomaly (s)	Detection (s)	Difference (s)	
1	43.9767	43.9773	-0.0006	
2	102.6123	102.6130	-0.0007	

Label	Precision	Recall	F1-Score	Support
Good	0.999	0.999	0.999	600,000
Anomaly	0.999	0.999	0.999	400,000
Accuracy	—	—	0.999	1,000,000
Macro avg	0.999	0.999	0.999	1,000,000
Weighted avg	0.999	0.999	0.999	1,000,000

The system correctly classified 149,996 out of 150,000 "Middle" samples (99.997%) and all 250,000 "Anomaly" samples (100%), corresponding to only four false negatives. This yields an overall error rate of 0.001% for the anomalous dataset and 0.0026% for the "Middle" category, while five false positives led to a 0.00083% error rate within the "Good" dataset.

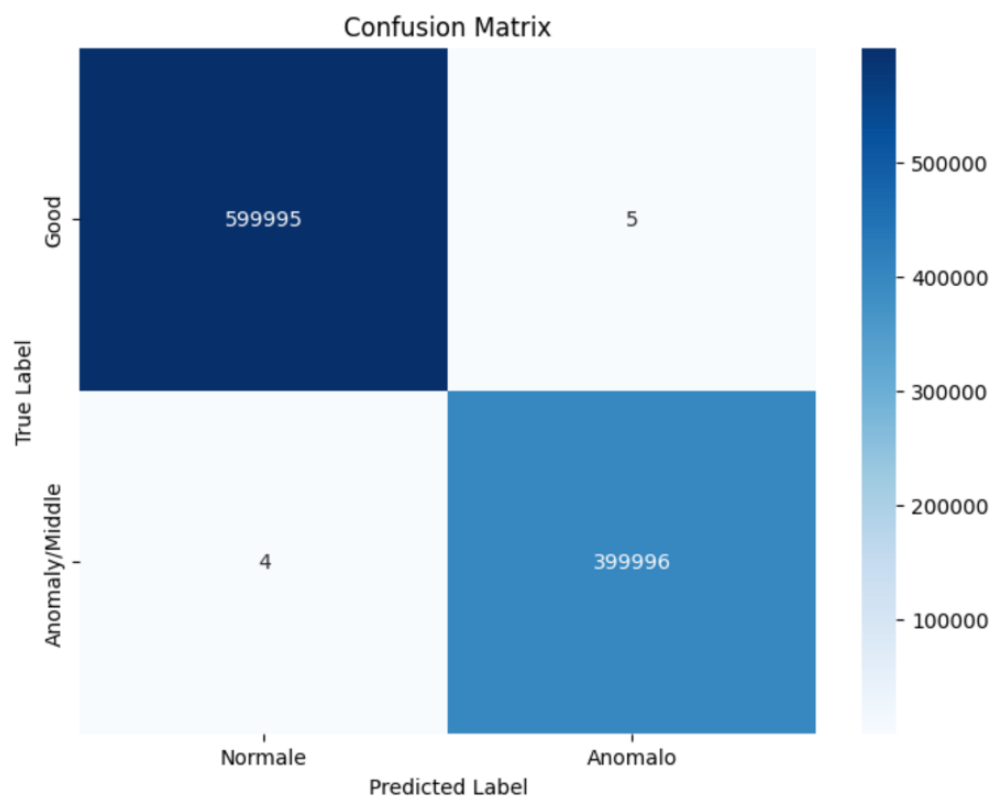


Fig. 5.12 First experiment, transient anomaly, confusion matrix

Transient Anomaly Detection — Second Experiment

The second transient experiment introduced anomalies within sequences of "Good" data to assess responsiveness. Results are shown in Table 5.9.

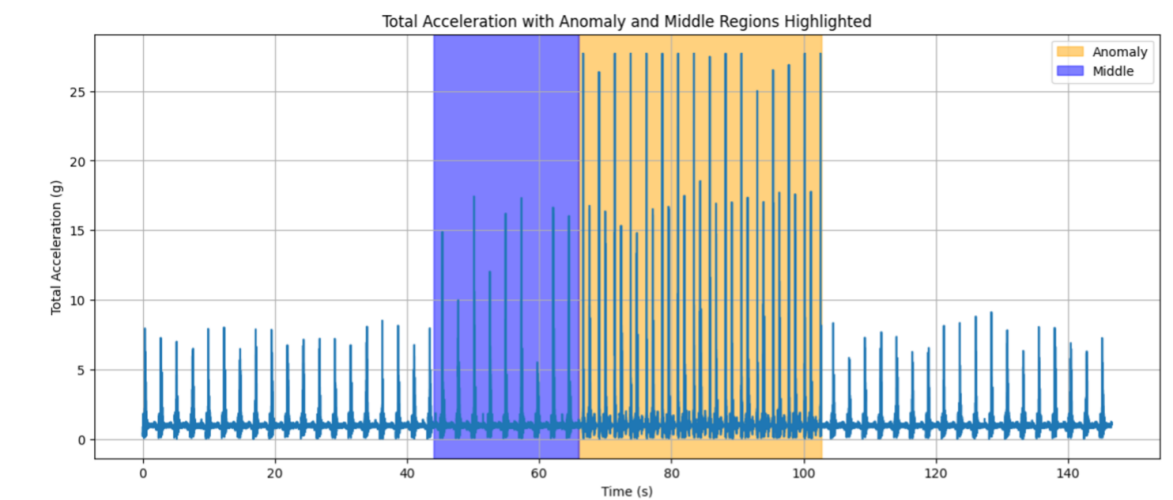


Fig. 5.13 Second experiment, transient anomaly, band anomalies

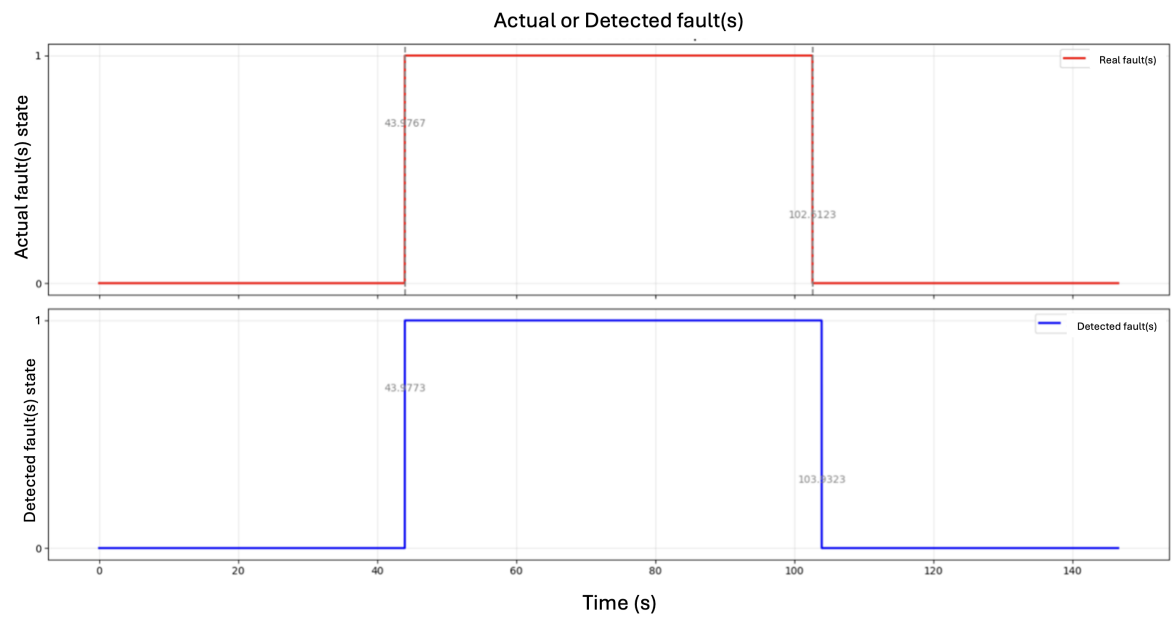


Fig. 5.14 Second experiment, transient anomaly, point anomalies

The system achieved perfect classification for the "Anomaly" samples (100%) and 149,996 out of 150,000 "Middle" samples (99.9986%). Four false negatives occurred, while 9,005 false positives within the "Good" data produced a 1.5% error rate for that category.

Table 5.9 Anomaly Detection Timing and Performance Metrics — Second Transient Experiment

No.	Actual Anomaly (s)	Detection (s)	Difference (s)
1	43.9767	43.9773	-0.0006
2	102.6123	103.9323	1.32

Label	Precision	Recall	F1-Score	Support
Good	0.999	0.984	0.992	600,000
Anomaly	0.978	0.999	0.989	400,000
Accuracy	—	—	0.991	1,000,000
Macro avg	0.989	0.992	0.991	1,000,000
Weighted avg	0.991	0.991	0.991	1,000,000

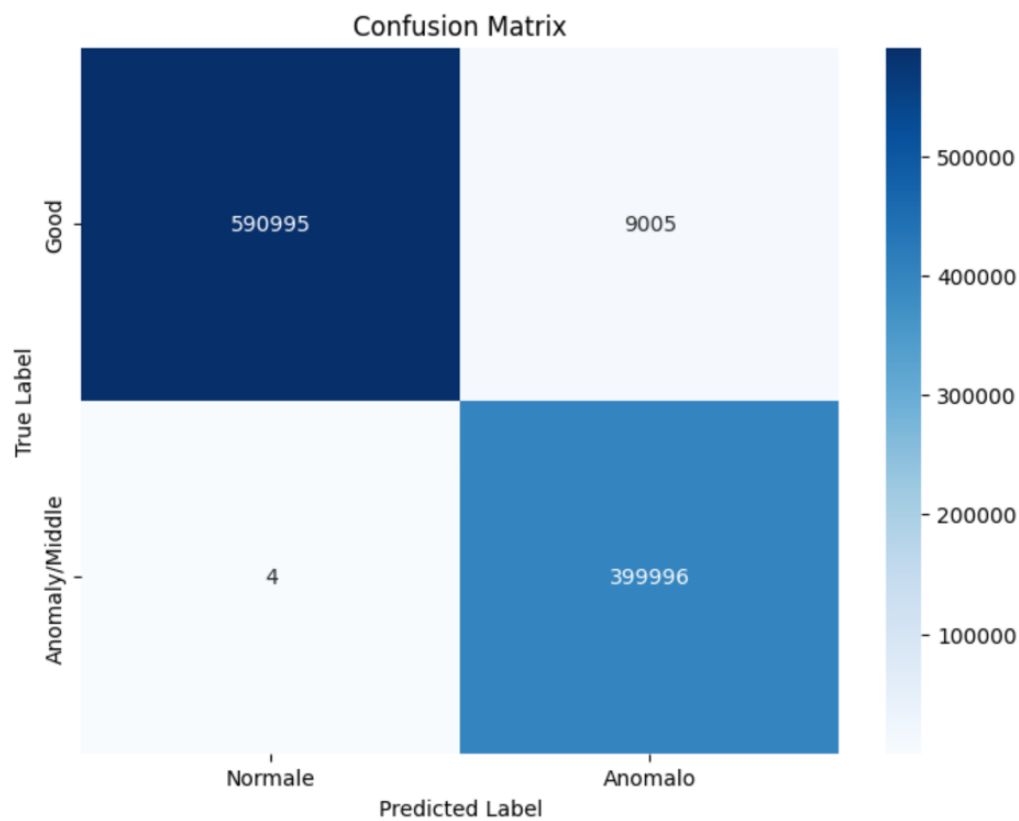


Fig. 5.15 Second experiment, transient anomaly, confusion matrix

Mixed Anomalies — First Experiment

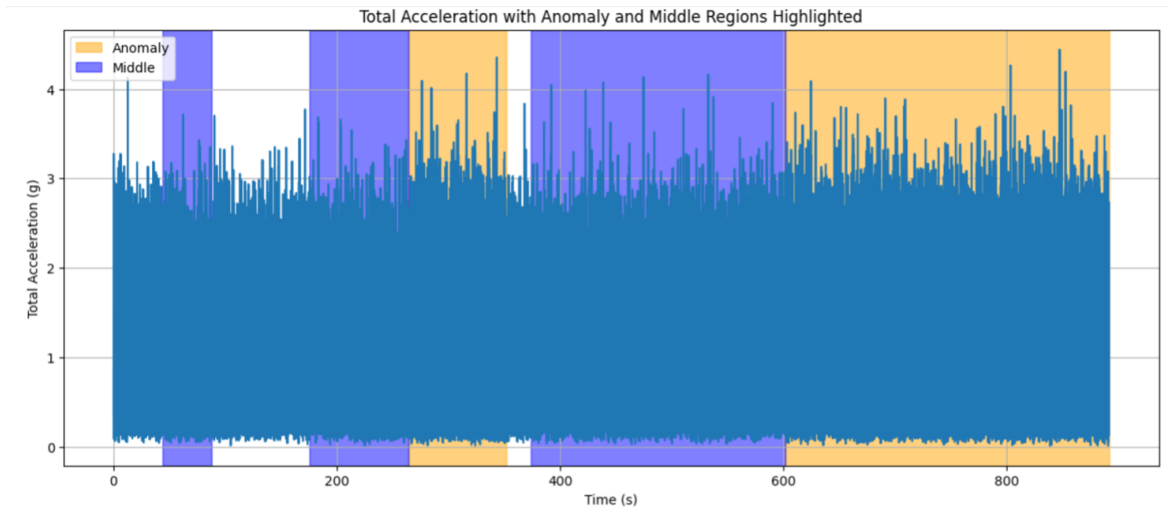


Fig. 5.16 First experiment, mixed anomaly, band anomalies

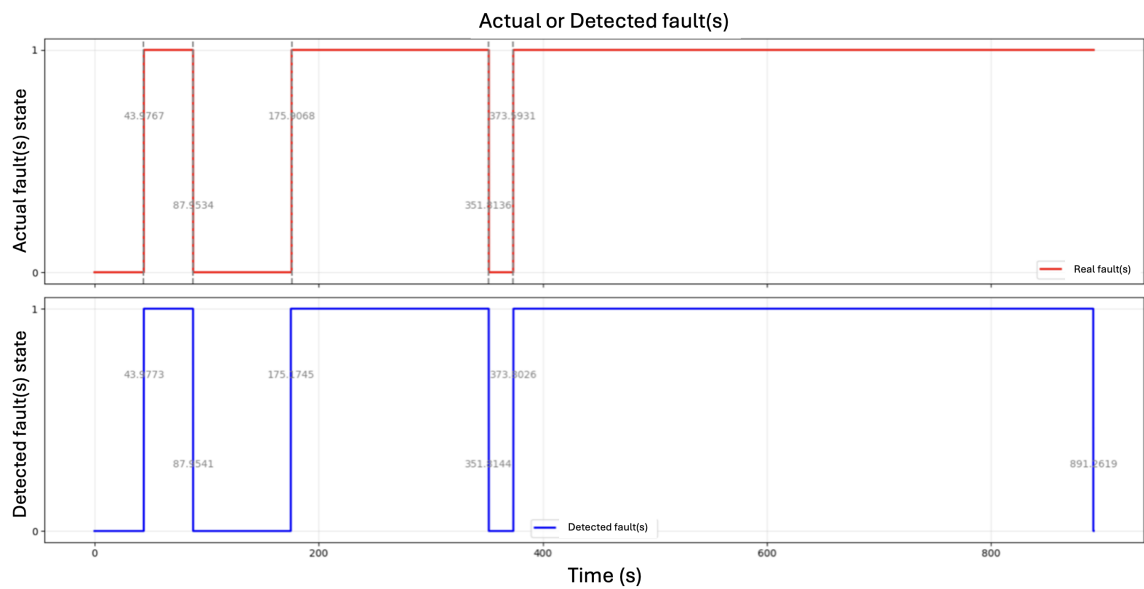


Fig. 5.17 First experiment, mixed anomaly, point anomalies

To evaluate robustness under diverse fault conditions, a composite dataset containing multiple anomaly types was tested. Table 5.10 reports detection timing and classification metrics.

Table 5.10 Anomaly Detection Timing and Performance Metrics — Mixed Anomalies, First Experiment

N°	Real Anomaly (s)	Detection (s)	Difference (s)	
1	43.9767	43.9773	-0.0006	
2	87.9534	87.9541	-0.0007	
3	175.9068	175.1745	0.7323	
4	351.8136	351.8144	-0.0008	
5	373.5931	373.8026	-0.2095	
6	891.6385	891.2619	0.3766	

Label	Precision	Recall	F1-Score	Support
Good	0.996	0.995	0.995	1,048,575
Anomaly	0.999	0.999	0.999	5,034,000
Accuracy	—	—	0.998	6,082,575
Macro avg	0.997	0.997	0.997	6,082,575
Weighted avg	0.998	0.998	0.998	6,082,575

The system correctly classified 2,456,567 out of 2,458,000 "Middle" samples (99.941%) and 2,573,430 out of 2,576,000 "Anomaly" samples (99.900%). This corresponded to 4,003 false negatives and 5,006 false positives, yielding error rates of 0.079% and 0.479%, respectively, across the anomalous and good datasets.

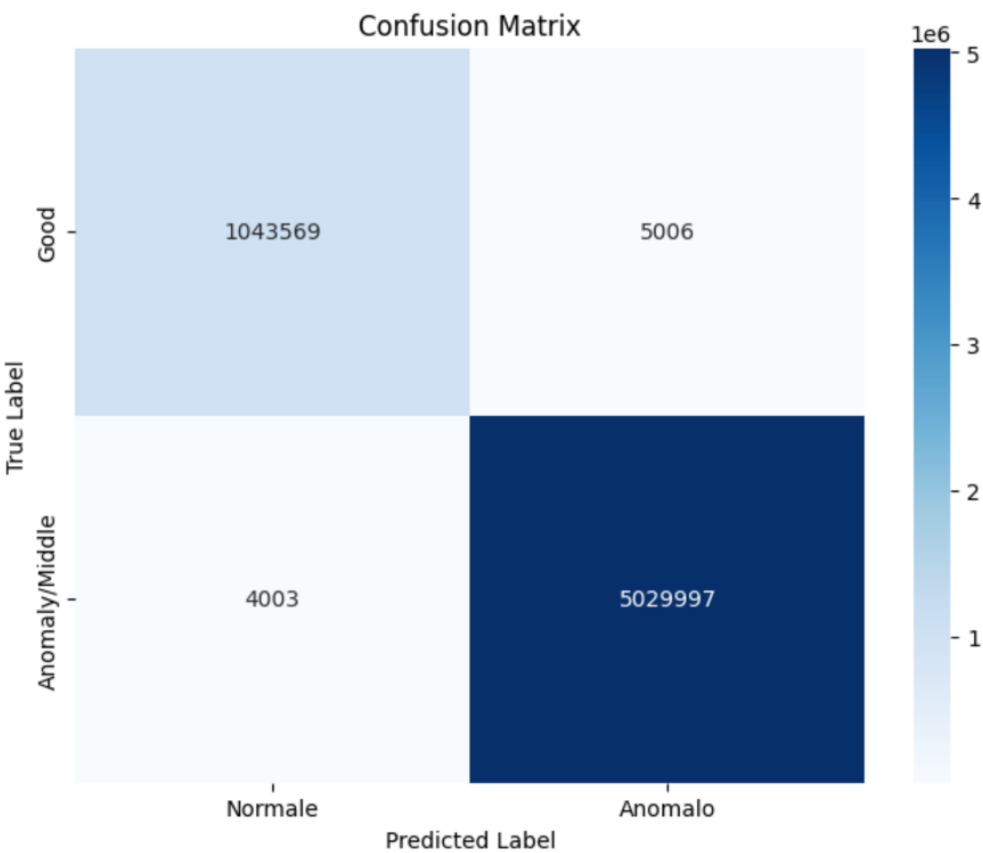


Fig. 5.18 First experiment, mixed anomaly, confusion matrix

Mixed Anomalies — Second Experiment

A second mixed anomaly experiment evaluated detection performance under more challenging conditions. Table 5.11 presents the results.

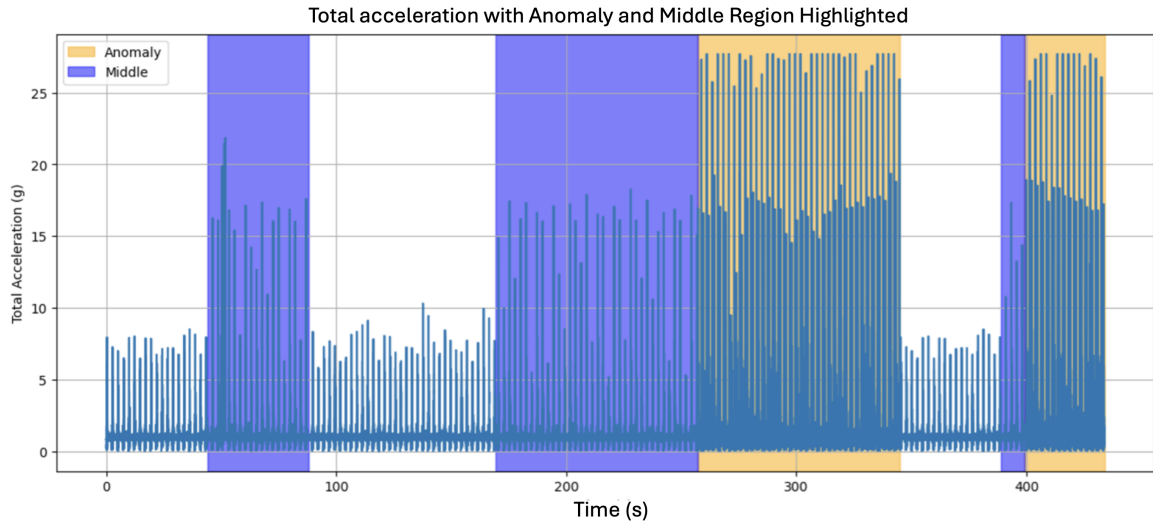


Fig. 5.19 Second experiment, mixed anomaly, band anomalies

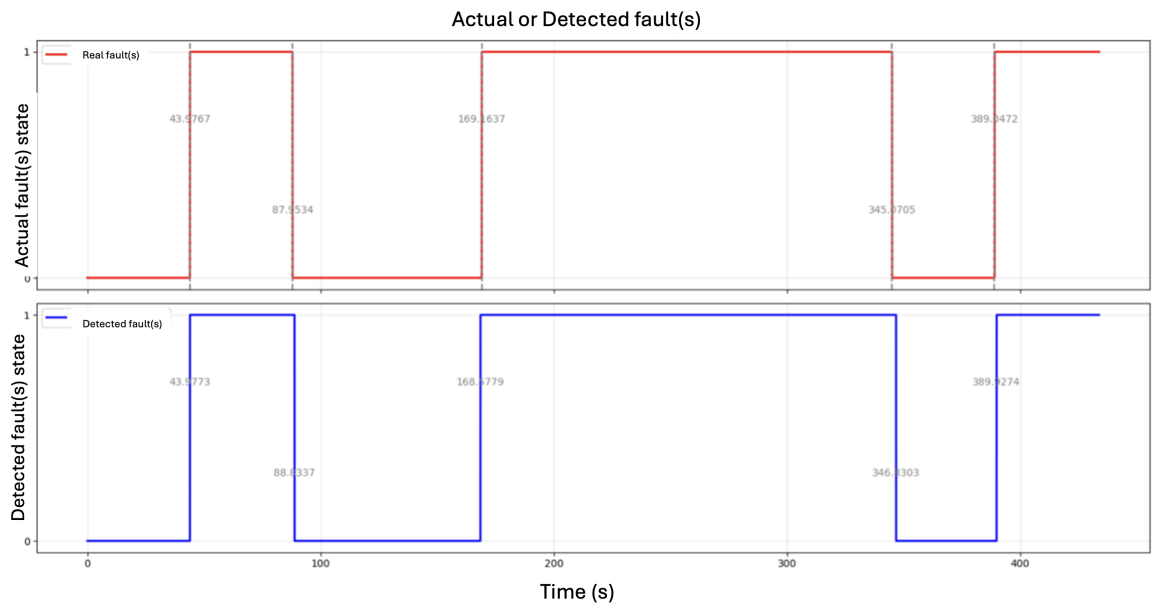


Fig. 5.20 Second experiment, mixed anomaly, point anomalies

The model correctly classified 963,992 out of 970,000 "Middle" samples (99.38%) and all 836,000 "Anomaly" samples (100%). The experiment yielded 6,008 false negatives (0.332% error) and 22,006 false positives (1.9% error) across the good dataset.

Table 5.11 Anomaly Detection Timing and Performance Metrics — Mixed Anomalies, Second Experiment

N°	Real Anomaly (s)	Detection (s)	Difference (s)
1	43.9767	43.9773	-0.0006
2	87.9534	87.8337	0.1197
3	169.1637	168.6779	0.4858
4	345.0705	346.8303	-1.7598
5	389.0472	389.9274	-0.8802

Label	Precision	Recall	F1-Score	Support
Good	0.994	0.980	0.987	1,154,000
Anomaly	0.987	0.999	0.996	1,806,000
Accuracy	—	—	0.991	2,960,000
Macro avg	0.991	0.988	0.990	2,960,000
Weighted avg	0.991	0.991	0.991	2,960,000

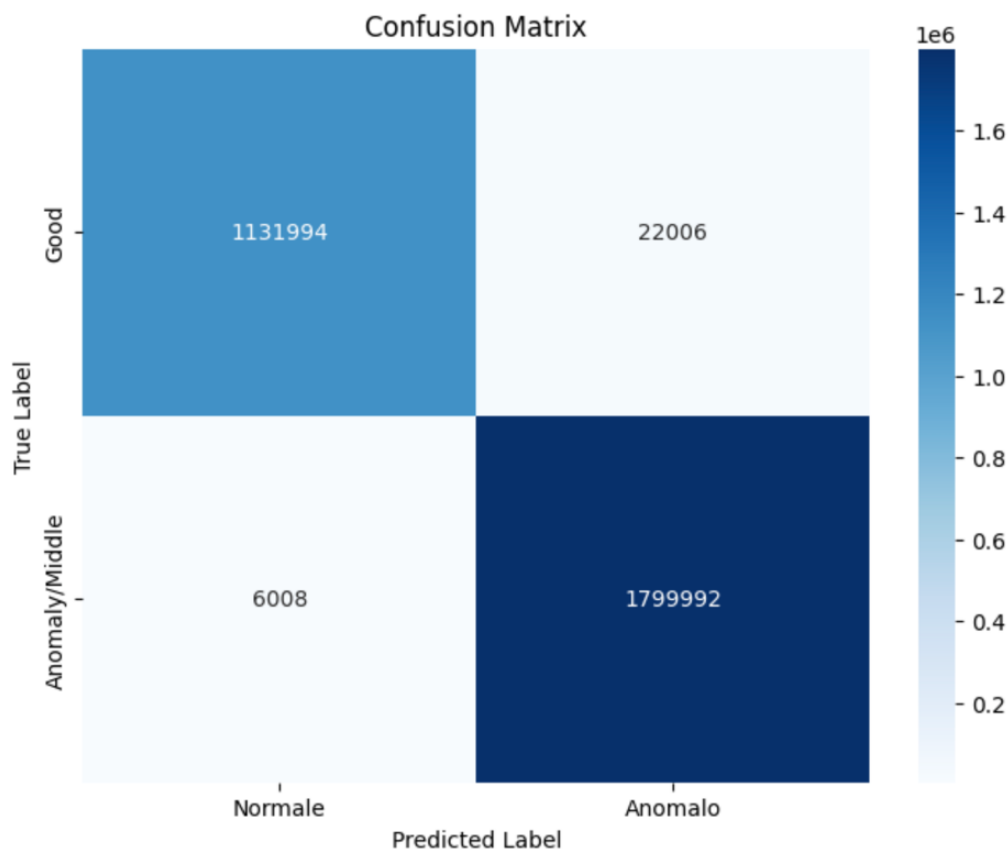


Fig. 5.21 Second experiment, mixed anomaly, confusion matrix

5.5 Concluding Remarks

The experimental campaign provides important insights into the feasibility of edge-based prognostics for industrial rotating machinery. Overall, the study demonstrates that the proposed hybrid pipeline, combining classical signal processing with machine learning models, is technically viable and deployable on resource-constrained edge devices. Average latencies were well below one second, supporting near real-time operation. Detection performance was high in controlled experiments, reflecting the effectiveness of anomaly-specific feature engineering.

Despite these positive results, several limitations and considerations must be acknowledged:

- **Controlled experimental conditions:** Anomalies were introduced under laboratory conditions, producing relatively clean signals. In real industrial environments, operational variability, noise, and non-stationary regimes are expected to reduce accuracy. Therefore, reported metrics should be interpreted as optimistic upper bounds rather than direct predictions of field performance.
- **False positives:** Transient anomaly experiments revealed a small but non-negligible false positive rate (1.5%). While seemingly low, at scale this could generate a substantial number of unnecessary alerts, potentially causing alert fatigue and reducing trust in the system.
- **Industrial validation constraints:**
 - Long-term degradation phenomena, such as bushing wear and spring loosening, require extended periods to manifest.
 - Access to production machinery was limited, as machines were either continuously in use or already compromised, making full-scale validation impractical within the project timeframe.
- **Generalizability:** The study focused on two degradation modes on a single machine type. While sufficient to demonstrate feasibility, broader validation across heterogeneous assets and longer operational horizons is needed before claiming industrial readiness.

In summary, the proposed edge-based prognostics pipeline is technically sound and capable of capturing subtle operational variations. Future work will focus on:

- Enhancing generalization to diverse machinery and operating conditions.
- Integrating certified diagnostic references.
- Exploring richer feature sets and multi-modal data integration.
- Scaling the framework for deployment across heterogeneous industrial environments.

These efforts aim to bridge the gap between laboratory-grade performance and robust, trustworthy field deployment.

Chapter 6

Conclusions

This chapter summarizes the contributions, key findings, and industrial implications of this doctoral research. The main objective was to design and validate a data-centric fog computing framework capable of supporting predictive maintenance in industrial environments, while providing flexibility for future expansion and third-party integration.

6.1 Key Findings and Contributions

6.1.1 From Raw Data to Actionable Features

A central contribution of this work is the demonstration that industrial value is generated not merely by collecting sensor data, but by transforming it into actionable features. The proposed framework performs local preprocessing, normalization, and feature extraction to convert raw signals into structured descriptors suitable for predictive analytics.

Signal selection rationale: Slowly varying signals (temperature, pressure) were modeled using LSTM networks achieving forecasting accuracy above 99%, while high-frequency signals (vibration) were processed through hybrid feature extraction (signal processing + ML), reaching F1-scores above 98%. These choices were motivated by industrial relevance: temperature and pressure capture gradual degradation,

vibration captures sudden or progressive anomalies, and this combination maximizes detection coverage while optimizing computational load at the edge.

6.1.2 Decoupled and Customizable Intelligence

The framework enables decoupling from OEM ecosystems via containerized microservices, allowing:

- Deployment of user-defined feature extractors without modifying proprietary firmware.
- Creation of domain-specific analytics for integrators or researchers.
- Collaborative and federated intelligence where multiple stakeholders can innovate on shared data streams while preserving operational autonomy.

This approach clearly defines the candidate's original contribution: the system-level integration and industrial validation of a modular, customizable edge analytics framework.

6.1.3 Hybrid Predictive Maintenance Paradigm

By combining slow-signal forecasting with fast-signal anomaly detection, the system achieves:

- Responsiveness on edge devices without sacrificing accuracy.
- Enhanced interpretability of features, increasing operator trust in automated recommendations.
- Demonstrated industrial value, reducing unplanned downtime and improving maintenance scheduling (validated on a production line with measurable improvements in anomaly detection latency and predictive maintenance effectiveness).

6.2 Industrial and Scientific Impact

The framework supports IT-OT convergence, breaking silos between operational technologies and modern data infrastructures. Open and interoperable architecture fosters third-party innovation, and the focus on data as a primary asset aligns industrial competitiveness with intelligent feature exploitation rather than mere hardware ownership.

Quantitative industrial impact includes:

- Enabled early detection of anomalies in industrial machinery through real-time feature extraction and hybrid predictive models, providing operators with actionable insights where previously failures were only observed post-facto.
- Deployment on heterogeneous machines without OEM firmware modification.
- Potential scalability to multi-node scenarios, supporting future factory-wide predictive maintenance systems.

6.3 Limitations and Future Work

Limitations:

- Large-scale, heterogeneous deployments beyond the tested 50-node scenario require further evaluation for the study of accelerations in a real industrial scenario.
- Formal real-time guarantees (deadline enforcement, WCET analysis) were not implemented.
- Dataset and experimental procedures are fully documented and collected, but external availability is limited due to industrial constraints; providing anonymized or synthetic datasets could enhance reproducibility and support independent validation of the framework.

Future Directions:

- Integration with deterministic networking and next-generation communication protocols (5G, TSN, LoRaWAN).
- Federated and distributed learning across multiple fog nodes for privacy-preserving model training.
- Lifecycle management automation, adaptive scaling, and fault-tolerant deployment.
- Operator-in-the-loop strategies to combine automated predictions with human expertise.

6.4 Final Remarks

This dissertation demonstrates that industrial digital transformation depends on actionable feature generation, not merely sensor proliferation. By validating a modular, edge-deployable framework with hardware-aware design, containerized analytics, and hybrid predictive maintenance, it establishes a system-level contribution that bridges research innovation with industrial applicability.

The research positions data as a strategic asset, enabling flexible, customized analytics, collaborative ecosystems, and resilient business models for the next generation of industrial systems.

References

- [1] Jameel Shehu Yalli, Mohd Hilmi Hasan, and Aisha Abubakar Badawi. Internet of things (iot): Origins, embedded technologies, smart applications, and its growth in the last decade. *IEEE Access*, 12:91357–91382, 2024.
- [2] Mumin Adam, Mohammad Hammoudeh, Rana Alrawashdeh, and Basil Alsulaimy. A survey on security, privacy, trust, and architectural challenges in iot systems. *IEEE Access*, 12:57128–57149, 2024.
- [3] Ilja Behnke and Henrik Austad. Real-time performance of industrial iot communication technologies: A review. *IEEE Internet of Things Journal*, 11(5):7399–7410, 2024.
- [4] Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2025.
- [5] Hikmet Erbiyik. Definition of maintenance and maintenance types with due care on preventive maintenance. In *Maintenance Management-Current Challenges, New Developments, and Future Directions*. IntechOpen, 2022.
- [6] Tianyu Zhang, Chuanyu Xue, Jiachen Wang, Zelin Yun, Natong Lin, and Song Han. A survey on industrial internet of things (iiot) testbeds for connectivity research. *arXiv preprint arXiv:2404.17485*, 2024.
- [7] Yujiao Hu, Qingmin Jia, Yuan Yao, Yong Lee, Mengjie Lee, Chenyi Wang, Xiaomao Zhou, Renchao Xie, and F. Richard Yu. Industrial internet of things intelligence empowering smart manufacturing: A literature review. *IEEE Internet of Things Journal*, 11(11):19143–19167, 2024.
- [8] Dr. A. Shaji George. The fourth industrial revolution: A primer on industry 4.0 and its transformative impact. *Partners Universal Innovative Research Publication*, 2(1):16–40, 2024.
- [9] Rajnish Rakholia, Andrés L. Suárez-Cetrulo, Manokamna Singh, and Ricardo Simón Carbajo. Advancing manufacturing through artificial intelligence: Current landscape, perspectives, best practices, challenges, and future direction. *IEEE Access*, 12:131621–131637, 2024.

- [10] McKinsey and Company. The internet of things: catching up to an accelerating opportunity. Technical report, McKinsey and Company, November 2021.
- [11] A.E.H Gabsi. Integrating artificial intelligence in industry 4.0: insights, challenges, and future prospects—a literature review. *Ann Oper Res*, 2024.
- [12] Mohammad Abidur Rahman, Md Farhan Shahrior, Kamran Iqbal, and Ali A Abushaiba. Enabling intelligent industrial automation: A review of machine learning applications with digital twin and edge ai integration. *Automation*, 6(3):37, 2025.
- [13] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, 2024.
- [14] Ce Yan, Yuanqing Xia, Hongjiu Yang, and Yufeng Zhan. Cloud control for iiot in a cloud-edge environment. *Journal of Systems Engineering and Electronics*, 35(4):1013–1027, 2024.
- [15] Kamran Sattar Awaisi, Qiang Ye, and Srinivas Sampalli. A survey of industrial aiot: Opportunities, challenges, and directions. *IEEE Access*, 12:96946–96996, 2024.
- [16] Guneet Kaur Walia, Mohit Kumar, and Sukhpal Singh Gill. Ai-empowered fog/edge resource management for iot applications: A comprehensive review, research challenges, and future perspectives. *IEEE Communications Surveys & Tutorials*, 26(1):619–669, 2024.
- [17] Ons Aouedi, Thai-Hoc Vu, Alessio Sacco, Dinh C. Nguyen, Kandaraj Piamrat, Guido Marchetto, and Quoc-Viet Pham. A survey on intelligent internet of things: Applications, security, privacy, and future directions. *IEEE Communications Surveys & Tutorials*, 27(2):1238–1292, 2025.
- [18] T. et al Pauli. Inhibitors and enablers of leverage in digital industrial platform ecosystems. *Business & Information Systems Engineering*, 2025.
- [19] Maxim Chernyshev, Zubair Baig, Oladayo Bello, and Sherali Zeadally. Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, PP:1–1, 12 2017.
- [20] Carlos Carvalhais, Luísa Antunes Ribeiro, and Cristiana C. Pereira. Unveiling stress vulnerability and occupational noise perception as burnout predictors: Results of an exploratory study in industrial environments. *Environments*, 12(6), 2025.
- [21] Korodi A. et al. Integration of legacy industrial equipment in a building-management system industry 5.0 scenario. *IEEE Open Journal of the Communications Society*, 13, 2024.

- [22] Pietro d'Agostino, Massimo Violante, and Gianpaolo Macario. A user-extensible solution for deploying fog computing in industrial applications. In *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, pages 1–6, 2023.
- [23] Bilal Babayigit and Mohammed Abubaker. Industrial internet of things: A review of improvements over traditional scada systems for industrial automation. *IEEE Systems Journal*, 18(1):120–133, 2024.
- [24] Ijaz Ahmad, Felipe Rodriguez, Tanesh Kumar, Jani Suomalainen, Senthil Kumar Jagatheesaperumal, Stefan Walter, Muhammad Zeeshan Asghar, Gaolei Li, Nikolaos Papakonstantinou, Mika Ylianttila, Jyrki Huusko, Thilo Sauter, and Erkki Harjula. Communications security in industry x: A survey. *IEEE Open Journal of the Communications Society*, 5:982–1025, 2024.
- [25] Precious Odebisi and Lawrence Joseph Canny Smith. Cyber-physical system (cps) vulnerabilities in industrial iot applications. ResearchGate.
- [26] Fujun Qiu, Ashwini Kumar, Jiang Hu, Poorva Sharma, Yu Bing Tang, Yang Xu Xiang, and Jie Hong. A review on integrating iot, iiot, and industry 4.0: A pathway to smart manufacturing and digital transformation. *IET Information Security*, 2025(1):9275962, 2025.
- [27] M. R. McCormick, Mohammed Shafae, and Thorsten Wuest. Machine tool interoperability in smart manufacturing and industry 4.0. *IEEE Access*, 13:117867–117913, 2025.
- [28] Pietro D'Agostino. Experimental application for bluetooth low energy industrial systems. Master's thesis, Politecnico di Torino, 2022.
- [29] Massimiliano Giacalone, Chiara Marciano, Claudia Pipino, Gianfranco Piscopo, and Stefano Marra. Technological innovation and the role of smart surveys in the industrial context. *Applied Sciences*, 15(16):8832, 2025.
- [30] Maryam Karimi and Roland Shaefer. Iiot communication protocols compatibility and security an in-depth review. *Authorea Preprints*, 2025.
- [31] K. Rameshkumar, Kaviarasu Nataraj, P. Krishnakumar, and M. Saimurugan. Machine learning approach for predicting the solid particle lubricant contamination in a spherical roller bearing. *IEEE Access*, 12:78680–78700, 2024.
- [32] Aleksandra Ptak and Tomasz Lis. Digital transformation in manufacturing enterprises: The role and impact of iiot. *Procedia Computer Science*, 246:5214–5223, 2024.
- [33] Priyanka Mudgal. A data-centric framework for implementing ai in smart manufacturing. 06 2025.

- [34] Oluwatosin Selesi-Aina, Nseabasi Edet Obot, Anthony Obulor Olisa, Michael Olayinka Gbadebo, Omobolaji Olateju, and Oluwaseun Oladeji Olaniyi. The future of work: A human-centric approach to ai, robotics, and cloud computing. *Journal of Engineering Research and Reports*, 26(11):10–9734, 2024.
- [35] Vishnu Vardhan Baligodugula, Ashutosh Ghimire, and Fathi Amsaad. An overview of secure network segmentation in connected iiot environments. *Computing&AI Connect*, 1(1):1–10, 2024.
- [36] Shiji Zhou, Bo Yuan, Kangming Xu, Mingxuan Zhang, and Wenxuan Zheng. The impact of pricing schemes on cloud computing and distributed systems. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, 3(3):193–205, 2024.
- [37] Roman Danel, Božena Gajdzik, and Hana Štverková. Risk factors in implementation of industry 4.0 for erp/mes in small and medium enterprises. *IDIMT-2024: Changes to ICT, Management, and Business Processes through AI*, 2024.
- [38] Vinayak Raja et al. Exploring challenges and solutions in cloud computing: A review of data security and privacy concerns. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, 4(1):121–144, 2024.
- [39] Heorhii Kuchuk and Eduard Malokhvii. Integration of iot with cloud, fog, and edge computing: a review. *Advanced Information Systems*, 8(2):65–78, 2024.
- [40] Praveen Borra. An overview of cloud computing and leading cloud service providers. *International Journal of Computer Engineering and Technology (IJCET) Volume*, 15:122–133, 2024.
- [41] Saleha Alharthi, Afra Alshamsi, Anoud Alseiari, and Abdulmalik Alwarafy. Auto-scaling techniques in cloud computing: Issues and research directions. *Sensors*, 24(17):5551, 2024.
- [42] Yifan Zhang, Bo Liu, Yulu Gong, Jiaxin Huang, Jingyu Xu, and Weixiang Wan. Application of machine learning optimization in cloud computing resource scheduling and management. In *Proceedings of the 5th international conference on computer information and big data applications*, pages 171–175, 2024.
- [43] Rahmah Alhamyani and Majid Alshammari. Machine learning-driven detection of cross-site scripting attacks. *Information*, 15(7):420, 2024.
- [44] Nisha Devi, Sandeep Dalal, Kamna Solanki, Surjeet Dalal, Umesh Kumar Lilhore, Sarita Simaiya, and Nasratullah Nuristani. A systematic literature review for load balancing and task scheduling techniques in cloud computing. *Artificial Intelligence Review*, 57(10):276, 2024.

- [45] Mohammed Jawad Al-Dujaili, Haider TH ALRikabi, Mohammad K Abdul-Hussein, Huda Abbas Kanber, and Ibtiha1 Razaq Niama ALRubeei. E-learning in the cloud computing environment: Features, architecture, challenges, and solutions. *International Journal of Engineering Pedagogy*, 14(1), 2024.
- [46] Sreedhar Yalamati. Data privacy, compliance, and security in cloud computing for finance. In *Practical Applications of Data Processing, Algorithms, and Modeling*, pages 127–144. IGI Global Scientific Publishing, 2024.
- [47] Ali Ateeq, Mohammed Abdulrazzaq Alaghbari, Ranyia Ali Ateeq, and Ammar Yousif Ahmed. Understanding and addressing data security and privacy concerns in modern cloud computing systems. In *2024 ASU international conference in emerging technologies for sustainability and intelligent systems (ICETISIS)*, pages 220–224. IEEE, 2024.
- [48] Mona Abughazalah, Wafaa Alsaggaf, Shireen Saifuddin, and Shahenda Sarhan. Centralized vs. decentralized cloud computing in healthcare. *Applied Sciences*, 14(17):7765, 2024.
- [49] Megha Sharma, Abhinav Tomar, and Abhishek Hazra. Edge computing for industry 5.0: Fundamental, applications, and research challenges. *IEEE Internet of Things Journal*, 11(11):19070–19093, 2024.
- [50] Oluwole Temidayo Modupe, Aanuoluwapo Ayodeji Otitoola, Oluwatayo Jacob Oladapo, Oluwatosin Oluwatimileyin Abiona, Oyekunle Claudius Oyeniran, Adebunmi Okechukwu Adewusi, Abiola Moshood Komolafe, and Amaka Obijuru. Reviewing the transformational impact of edge computing on real-time data processing and analytics. *Computer Science & IT Research Journal*, 5(3):693–702, 2024.
- [51] Rania A Alharbey and Faisal Jamil. Federated learning framework for real-time activity and context monitoring using edge devices. *Sensors*, 25(4):1266, 2025.
- [52] Javier Palomares, Estela Carmona-Cejudo, Cristina Cervelló-Pastor, Estefanía Coronado, Hatim Chergui, and Muhammad Shuaib Siddiqui. Inter-agv scheduling and a novel multi-agent collaborative protocol for intra-agv resource allocation in mec-enabled multi-agv scenarios. *IEEE Open Journal of the Communications Society*, 6:4238–4259, 2025.
- [53] Khaled M Matrouk, Punithavathi Rasappan, Priyanka Bhutani, Shikha Mittal, A Sahaya Anselin Nisha, and Reddy Madhavi Konduru. Development of heuristic strategy with hybrid encryption for energy efficient and secure data storage scheme in blockchain-based mobile edge computing. *Transactions on Emerging Telecommunications Technologies*, 36(2):e70057, 2025.
- [54] R Ala’a, Faten Khalid Karim, and Yu Wang. Optimizing resource allocation in industrial iot with federated machine learning and edge computing integration. *Results in Engineering*, page 106387, 2025.

- [55] Yuqing Wang and Xiao Yang. Research on edge computing and cloud collaborative resource scheduling optimization based on deep reinforcement learning. In *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, pages 2065–2073. IEEE, 2025.
- [56] Yiwen Xu, Tariq M Khan, Yang Song, and Erik Meijering. Edge deep learning in computer vision and medical diagnostics: a comprehensive survey. *Artificial Intelligence Review*, 58(3):93, 2025.
- [57] Akhilesh Bollam. Emerging trends in data synchronization for edge computing. *Journal of Computer Science and Technology Studies*, 7(5):102–108, 2025.
- [58] Wu Chia Chuan, Shams Ul Arfeen Laghari, Selvakumar Manickam, Erum Ashraf, and Shankar Karuppayah. Challenges and opportunities in fog computing scheduling: A literature review. *IEEE Access*, 13:14702–14726, 2025.
- [59] Ali Asghari and Mohammad Karim Sohrabi. Server placement in mobile cloud computing: A comprehensive survey for edge computing, fog computing and cloudlet. *Computer Science Review*, 51:100616, 2024.
- [60] Long Wen, Yu Zhang, Markus Rickert, Jianjie Lin, Fengjunjie Pan, and Alois Knoll. Cloud-native fog robotics: Model-based deployment and evaluation of real-time applications. *IEEE Robotics and Automation Letters*, 10(1):398–405, 2025.
- [61] Zahraa A Jaaz, Mohd Ezanee Bin Rusli, Nur Azzammudin Rahmat, and Maythem Kamal Abbas Al-Adilee. Latency optimization approaches for healthcare internet of things and fog computing: A comprehensive review. *Journal of Intelligent Systems*, 34(1):20240543, 2025.
- [62] Goran Maqdid, Shavan Askar, Diana Hayder Hussein Hussein, et al. Deep learning applications in fog computing environments: a review. *The Indonesian Journal of Computer Science*, 14(1), 2025.
- [63] Cisco DevNet. Frequently asked questions (faq). Technical report, Cisco DevNet, 2025-09-03.
- [64] Eclipse Foundation. Eclipse fog05. Technical report, Eclipse Foundation, 2025-09-30.
- [65] Nour El Houda Boubaker, Karim Zarour, Nawal Guermouche, and Djamel Benmerzoug. A comprehensive survey on resource management for iot applications in edge-fog-cloud environments. *IEEE Access*, 13:111892–111925, 2025.
- [66] G Navamani and L Jacqueline. Heuristic model implementation for automation in smart factories using fog computing. In *2025 International Conference on Networks and Cryptology (NETCRYPT)*, pages 846–851. IEEE, 2025.

- [67] Varun Varma Sangaraju, SV Achuta Rao, Amjan Shaik, VG Prasuna, and PB Nagaraju. Review on fog computing—applications, security, and solutions. *Proceedings on Engineering*, 7(1):447–458, 2025.
- [68] YIN Zengshan, WU Changhao, GUO Chongbin, LI Yuanchun, XU Mengwei, GAO Weiwei, and CHI Chuanxiu. A comprehensive survey of orbital edge computing: Systems, applications, and algorithms. *Chinese Journal of Aeronautics*, 38(7):103316, 2025.
- [69] Elias Dritsas and Maria Trigka. A survey on the applications of cloud computing in the industrial internet of things. *Big data and cognitive computing*, 9(2):44, 2025.
- [70] Paulo H Leocadio. Aws master class chapter 04: Content delivery network. *Authorea Preprints*, 2025.
- [71] Praveen et al. Borra. Azure edge computing: Enhancing iot deployments with microsoft azure. In *International Journal of Advanced Research in Science, Communication and Technology*, pages 417–424, 2025.
- [72] K. et al. Sathupadi. Edge-cloud synergy for ai-enhanced sensor network data: A real-time predictive maintenance framework. *Sensors*, 24:7918, 2024.
- [73] Juan et al Montes-Sánchez. Predictive maintenance edge artificial intelligence application study using recurrent neural networks for early aging detection in peristaltic pumps. *IEEE Transactions on Reliability*, pages 1–15, 2024.
- [74] Nivethitha Somu and Nirupam Sannagowdara Dasappa. An edge-cloud iiot framework for predictive maintenance in manufacturing systems. *Advanced Engineering Informatics*, 65:103388, 2025.
- [75] Woojin Kwon, Jieun Yang, Seunghwa Song, Jun Lee, and Hyungjung Kim. Real-time digital-twin-based cobot-worker collision risk prediction using unity, ros, and uwb. *IEEE Access*, 13:85967–85978, 2025.
- [76] Dejan Vukobratovic, Nikolaos Bartzoudis, Mona Ghassemian, Firooz B Saghezchi, Peizheng Li, Adnan Aijaz, Ricardo Martinez, Xueli An, Ranga Rao Venkatesha Prasad, Helge Luders, et al. Distributed sensing, computing, communication, and control fabric: A unified architecture for new 6g era. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2024.
- [77] Yihong Zou. Application of cross platform integration technology in modern communication systems. *European Journal of AI, Computing & Informatics*, 1(1):64–70, 2025.
- [78] Chi-Yao Weng, Chun-Ta Li, Chin-Ling Chen, Cheng-Chi Lee, and Yong-Yuan Deng. A lightweight anonymous authentication and secure communication scheme for fog computing services. *IEEE Access*, PP:1–1, 10 2021.

- [79] Nils Jahnke, Marieke Rohde, and Tom Kraus. Edge computing for digital sovereignty in the data economy. In *New Digital Work II: Digital Sovereignty of Companies and Organizations*, pages 247–263. Springer Nature Switzerland Cham, 2025.
- [80] Mario Lepore, Domenico Serra, and Raffaele Maccioni. Leveraging artificial intelligence and optimization for agile agv scheduling in an edge-to-cloud manufacturing framework. *Soft Computing*, pages 1–29, 2025.
- [81] Bethini Kavya and Kakarla Hari Kishore. Internet of robotic things (iort): An intelligent framework for automated manufacturing. *IIRJET*, 10(3), 2025.
- [82] Massimiliano Gaffurini, Dennis Brandao, Stefano Rinaldi, Alessandra Flammini, Emiliano Sisinni, and Paolo Ferrari. Characterizing the real-time communication performance of virtual plc in industrial edge platform. *IEEE Open Journal of Instrumentation and Measurement*, 2025.
- [83] Moritz Walker, Steffen Wörtz, Michael Neubauer, Armin Lechler, and Alexander Verl. Scheduling for the orchestration of distributed real-time applications. In *2025 IEEE 8th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 01–08. IEEE, 2025.
- [84] N. et al. Fernando. On edge-fog-cloud collaboration and reaping its benefits: A heterogeneous multi-tier edge computing architecture. *Future Internet*, 17(22), 2025.
- [85] Diyaz Yakubov and David Hästbacka. Comparative analysis of lightweight kubernetes distributions for edge computing: Performance and resource efficiency. In *European Conference on Service-Oriented and Cloud Computing*, pages 81–95. Springer, 2025.
- [86] Abdullahi Dauda, Olivier Flauzac, and Francois Nolot. A survey on iot application architectures. *Sensors*, 24(16):5320, 2024.
- [87] W. Onate and R. Sanz. Integration of fog computing in a distributed manufacturing execution system under the rami 4.0 framework. *Applied Sciences*, 14(22):10539, 2024.
- [88] S. Hong, S. Park, H. Youn, J. Lee, and S. Kwon. Implementation of smart farm systems based on fog computing in artificial intelligence of things environments. *Sensors*, 24(20):6689, 2024.
- [89] Pan S et al. Optimizing internet of things fog computing: Through lyapunov-based long short-term memory particle swarm optimization algorithm for energy consumption optimization. *Sensors*, 24(4):1165, 2024.
- [90] Shahidullah Kaiser, Ali Saman Tosun, and Turgay Korkmaz. Exploring the viability of unikernels for arm-powered edge computing. *arXiv preprint arXiv:2412.03030*, 2024.

- [91] Shahidullah Kaiser, Ali Saman Tosun, and Turgay Korkmaz. Edge system design using containers and unikernels for iot applications. *arXiv preprint arXiv:2412.03032*, 2024.
- [92] Hikmet Erbiyik. Definition of maintenance and maintenance types with due care on preventive maintenance. In *Maintenance Management-Current Challenges, New Developments, and Future Directions*. IntechOpen, 2022.
- [93] Mohammad Reza Rezaee, Nor Asilah Wati Abdul Hamid, Masnida Hussin, and Zuriati Ahmad Zukarnain. Fog offloading and task management in iot-fog-cloud environment: Review of algorithms, networks, and sdn application. *IEEE Access*, 12:39058–39080, 2024.
- [94] A. Kazem, G. Pierre, and L. Longuevergne. Enhancing environmental observatories with fog computing. *Frontiers in Environmental Science*, 13:1568016, 2025.
- [95] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.
- [96] Khalid Lefrouni and Saoudi Taibi. Artificial intelligence techniques for industrial predictive maintenance: A systematic review of recent advances. *Journal Europeen des Systemes Automatises*, 58(4), 2025.
- [97] Marco Belim, Tiago Meireles, Gil Goncalves, and Rui Pinto. Forecasting models analysis for predictive maintenance. *Frontiers in Manufacturing Technology*, 4:1475078, 2024.
- [98] Tianwen Zhu, Yongyi Ran, Xin Zhou, and Yonggang Wen. A survey on intelligent predictive maintenance (ipdm) in the era of fully connected intelligence. *IEEE Communications Surveys & Tutorials*, 2025.
- [99] Md Zikar Hossan and Taslima Sultana. Ai for predictive maintenance in smart manufacturing. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 17(03):25–33, 2025.
- [100] Uzair Khan, Dong Cheng, Francesco Setti, Franco Fummi, Marco Cristani, and Luigi Capogrosso. A comprehensive survey on deep learning-based predictive maintenance. *ACM Transactions on Embedded Computing Systems*, 2025.
- [101] Najim Alshahrani. Machine learning approaches for predictive maintenance in industrial operations. In *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 365–372. IEEE, 2024.
- [102] Abdalla Omar Dagroum Ali and Houssein M.A Elasad. A review of methodologies, techniques, and applications for predictive maintenance. *International Journal of Engineering Research and Management (IJERM)*, 12(2), 2025.

- [103] Rajeev Kumar Bhaskar and Balasubramaniam Kumaraswamy. Harnessing cnn and lstm models for enhanced predictive maintenance in industrial iot systems. In *2025 International Conference on Automation and Computation (AUTOCOM)*, pages 247–251. IEEE, 2025.
- [104] Yi Di et al. Boon. Recent advances on the design automation for performance-optimized fiber reinforced polymer composite components. *Journal of Composites Science*, 4(61):247–251, 2020.
- [105] R Archana and PS Eliahim Jeevaraj. Deep learning models for digital image processing: a review. *Artificial Intelligence Review*, 57(1):11, 2024.
- [106] Ibomoiye Domor Mienye, Theo G Swart, and George Obaido. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9):517, 2024.
- [107] Wasim Zaman, Muhammad Farooq Siddique, Shafi Ullah Khan, and Jong-Myon Kim. A new dual-input cnn for multimodal fault classification using acoustic emission and vibration signals. *Engineering Failure Analysis*, page 109787, 2025.
- [108] Camelia Adela Maican, Cristina Floriana Pană, Daniela Maria Pătrașcu-Pană, and Virginia Maria Rădulescu. Review of fault detection and diagnosis methods in power plants: Algorithms, architectures, and trends. *Applied Sciences*, 15(11):6334, 2025.
- [109] Zehui Zhao, Laith Alzubaidi, Jinglan Zhang, Ye Duan, and Yuantong Gu. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Systems with Applications*, 242:122807, 2024.
- [110] A. Karimi, N. Razi, and J. Rezazadeh. An iot healthcare system based on fog computing and data mining: A diabetic use case. *Applied Sciences*, 14(17):7924, 2024.
- [111] Pietro D’Agostino, Massimo Violante, and Gianpaolo Macario. An embedded low-cost solution for a fog computing device on the internet of things. In *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 284–291, 2023.
- [112] Wikipedia. Long short-term memory. Technical report, Wikipedia, 2025-09-22.
- [113] Pietro D’Agostino, Massimo Violante, and Gianpaolo Macario. Optimizing lstm-based temperature prediction algorithm for embedded system deployment. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 01–07, 2024.
- [114] Pietro D’Agostino, Massimo Violante, and Gianpaolo Macario. A scalable fog computing solution for industrial predictive maintenance and customization. *Electronics*, 24, 12 2025.

- [115] Made Adi Paramartha Putra, Ade Pitra Hermawan, Cosmas Ifeanyi Nwakanma, Dong-Seong Kim, and Jae-Min Lee. Fdpr: A novel fog data prediction and recovery using efficient dl in iot networks. *IEEE Internet of Things Journal*, 10(19):16895–16906, 2023.
- [116] D.B. et al Pazhanivel. Design and enhancement of a fog-enabled air quality monitoring and prediction system: An optimized lightweight deep learning model for a smart fog environmental gateway. *Sensors*, 24(5069), 2024.
- [117] Suat Özdemir Mehmet Ulvi Şimsek, İbrahim Kök. Deepfogaq: A fog-assisted decentralized air quality prediction and event detection system. *Expert Systems with Applications*, 251(123920), 2024.
- [118] S. Z. Ali, S. H. Ali, A. Javeed, I. Ullah, S. A. Memon, and S. Manzoor. Supervised learning approaches for vibration analysis and anomaly detection in rotating machines. In *Proceedings of the 2024 3rd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)*, Lahore, Pakistan, 2024. IEEE.
- [119] M. P. P. Kumar, J.-W. Liang, and K.-C. Chen. Neural network-based anomaly detection architecture for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 74:1–14, 2025.
- [120] S. Lee and T. Kim. Hybrid approach for machine anomaly detection across diverse operating conditions: Combining self-supervised learning with vibration statistics. *IEEE Transactions on Instrumentation and Measurement*, 74:1–15, 2025.
- [121] M.-K. Seo, B. H. Ryu, Y. J. Jeon, J. H. Hong, and H. J. Moon. Self-supervised drift-resilient classification for time series industrial anomaly detection. *IEEE Access*, 13:78021–78043, 2025.
- [122] X. Zhou, Y. Wang, J. Li, and S. Chen. Anomaly detection in rotating machinery vibration signals based on domain transformation and transformer. In *Proceedings of the 2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2024. IEEE.
- [123] K. A. Styanidis, M. L. Papadopoulos, S. D. Dimitriou, D. I. Ioannidis, D. I. Tzovaras, and I. K. Koutras. Vibration-based anomaly detection on weather radar rotating machinery using one dimensional convolutional autoencoder. In *Proceedings of the 2024 2nd International Conference on Intelligent Perception and Computer Vision (CIPCV)*, Xiamen, China, 2024. IEEE.
- [124] D. Hu, C. Zhang, T. Yu, and Y. Gao. An intelligent anomaly detection method for rotating machinery based on vibration vectors. *IEEE Sensors Journal*, 22(14):14294–14305, 2022.

- [125] S.-C. Hsia and C.-Y. Hsu. Real-time monitor and control system for abnormal motor vibrations. *IEEE Transactions on Instrumentation and Measurement*, 73:1–11, 2024.
- [126] Z. Shi, J. Yang, Y. Zhao, and C. Zhou. Decouplingnet: A stable knowledge distillation decoupling net for fault detection of rotating machines under varying speeds. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):11276–11289, 2024.
- [127] Pietro D’Agostino, Massimo Violante, and Gianpaolo Macario. A hybrid vibration-based anomaly detection system for predictive maintenance of capping equipment. In *IEEE AISTA 2025 - 2025 Artificial Intelligence and Smart Technology Applications Symposium*, 2025.