

## Article

# Non-Intrusive Low-Cost IoT-Based Hardware System for Sustainable Predictive Maintenance of Industrial Pump Systems

Sérgio Duarte Brito <sup>1,\*</sup>, Gonçalo José Azinheira <sup>1</sup>, Jorge Filipe Semião <sup>1</sup>, Nelson Manuel Sousa <sup>1</sup>  
and Salvador Pérez Litrán <sup>2</sup>

<sup>1</sup> Department of Electrical Engineering, Instituto Superior de Engenharia, University of Algarve, 8005-139 Faro, Portugal; gjazinheira@ualg.pt (G.J.A.); jsemiao@ualg.pt (J.F.S.); nsousa@ualg.pt (N.M.S.)

<sup>2</sup> Departamento de Ingeniería Eléctrica y Térmica, de Diseño y Proyectos, Escuela Técnica Superior de Ingeniería, Universidad de Huelva, 21007 Huelva, Spain; salvador@dfaie.uhu.es

\* Correspondence: sdbrito@ualg.pt

## Abstract

Industrial maintenance has shifted from reactive repairs and calendar-based servicing toward data-driven predictive strategies. This paper presents a non-intrusive, low-cost IoT hardware platform for sustainable predictive maintenance of rotating machinery. The system integrates an ESP32-S3 sensor node that captures vibration (100 kHz) and temperature data, performs local logging, and communicates wirelessly. An automated spectral band segmentation framework is introduced, comparing equal-energy, linear-width, nonlinear, clustering, and peak–valley partitioning methods, followed by a weighted feature scheme that emphasizes high-value bands. Three unsupervised one-class classifiers—transformer autoencoders, GANomaly, and Isolation Forest—are evaluated on these weighted spectral features. Experiments conducted on a custom pump test bench with controlled anomaly severities demonstrate strong anomaly classification performance across multiple configurations, supported by detailed threshold-characterization metrics. Among 150 model–segmentation configurations, 25 achieved perfect classification (100% precision, recall, and F1 score) with ROC-AUC = 1.0, 43 configurations achieved  $\geq 90\%$  accuracy, and the lowest-performing setup maintained 81.8% accuracy. The proposed end-to-end solution reduces the downtime, lowers maintenance costs, and extends the asset life, offering a scalable, predictive maintenance approach for diverse industrial settings.



Academic Editor: Marcin Witczak

Received: 11 June 2025

Revised: 10 July 2025

Accepted: 15 July 2025

Published: 21 July 2025

**Citation:** Brito, S.D.; Azinheira, G.J.; Semião, J.F.; Sousa, N.M.; Litrán, S.P. Non-Intrusive Low-Cost IoT-Based Hardware System for Sustainable Predictive Maintenance of Industrial Pump Systems. *Electronics* **2025**, *14*, 2913. <https://doi.org/10.3390/electronics14142913>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** predictive maintenance; IoT; vibration analysis; wireless sensor networks; machine learning; low-cost sensors; FFT; anomaly detection

## 1. Introduction

Industrial maintenance strategies have evolved from reactive “run-to-failure” repairs to scheduled preventive servicing and now toward data-driven predictive approaches. In corrective maintenance, equipment is repaired only after a breakdown, incurring unplanned downtime and high emergency costs. Preventive maintenance relies on calendar- or usage-based schedules to replace or service components before failure. While safer than reactive servicing, it can lead to unnecessary interventions and wastage of parts. Predictive maintenance, by contrast, leverages real-time sensor data and analytics to forecast faults and optimize interventions, offering the potential for minimal downtime, reduced maintenance costs, and extended asset life.

Recent trends in Industry 4.0 and the Industrial Internet of Things (IIoT) have accelerated the adoption of predictive maintenance across manufacturing, utilities, transportation,

and building services. Advances in affordable sensors, edge computing, and machine learning enable continuous condition monitoring and local anomaly detection without constant cloud connectivity. Yet practical deployment faces challenges: integrating with legacy instrumentation, ensuring data quality under harsh environments, balancing model complexity against on-site compute resources, and tuning thresholds to minimize false alarms.

Rotating machinery—pumps, compressors, motors—is a primary target for vibration-based fault detection, as mechanical defects often manifest as distinct spectral signatures. Non-intrusive vibration sensing (e.g., clamp-on accelerometers) is attractive for retrofit scenarios, but real-world installations may restrict sensor placement or require integration with existing process instrumentation. Moreover, the simple thresholding of overall vibration levels can miss subtle or evolving faults, motivating more advanced spectral feature extraction and unsupervised anomaly detection tailored to limited “healthy” training data.

In this paper, we propose and validate an end-to-end predictive maintenance platform that combines the following:

- A low-power IoT sensor node, non-intrusively capturing 100 kHz vibration sampling rate and temperature data, with local data logging and wireless connectivity.
- An automated spectral band segmentation framework, evaluating five partitioning methods and scoring each band by repeatability, sensitivity, predictability, directional consistency, and smoothness.
- A feature-weighting scheme that emphasizes high-value bands and attenuates noisy ones, optionally integrating band powers or preserving full spectral slices with  $L_2$ -normalization.
- An unsupervised, one-class learning pipeline using three model families—Transformer Autoencoders (TAE), GANomaly, and Isolation Forest—chosen for their range of computational footprints and representational capacities.
- A comprehensive evaluation on a custom pump test bench with stratified anomaly severities, demonstrating perfect classification in many configurations and providing in-depth threshold-characterization metrics.

The remainder of this paper is organized as follows: Section 2 reviews related works in vibration-based predictive maintenance and unsupervised anomaly detection. Section 3 details our sensor hardware and embedded software architecture. Section 4 describes the experimental setup and dataset composition. Section 5 presents spectral preprocessing and band segmentation analyses. Section 6 covers the ML training and testing methodologies and quantitative results. Section 7 discusses implications, limitations, and deployment considerations. Finally, Section 8 concludes and outlines future works toward real-world pilot deployments and multimodal fusion.

Our test-bench evaluation encompassed 150 model-segmentation configurations, of which 25 yielded perfect anomaly classification (100% precision, recall, F1), and 43 more exceeded 90% accuracy, with the worst case still at 81.8% accuracy. By capturing 1 s, 100 kHz vibration snapshots—with a cross-DAQ waveform correlation above 0.98 and a spectral alignment within  $\pm 0.5$  Hz—on roughly 100 “good” and 8000 “anomalous” runs, we demonstrate a robust, one-class detection paradigm under severe class imbalance. These results confirm that low-cost, open-source edge sensing can reliably forecast pump faults before failure, offering a scalable path to reduce unplanned downtime and maintenance overhead in industrial settings.

## 2. Related Work

Predictive maintenance (PdM) has evolved from reactive and preventive strategies toward data-driven, real-time anomaly detection, with vibration analysis as a central modality for rotating machinery such as pumps, fans, and compressors. Early work on

one-class classification—most notably the support vector data description by Schölkopf et al.—demonstrated that boundaries around normal data can flag novel faults without labeled failure examples [1,2]. Isolation Forests later showed competitive performance by scoring anomalies via average path lengths in random trees, requiring minimal tuning and no anomalous samples [3].

### 2.1. Industry Standards and Normative Context

In the context of vibration-based diagnostics, several ISO standards provide guidelines for the evaluation and interpretation of mechanical vibration signals. Notably, ISO 10816-3:2009 [4] and ISO 10816-7:2009 [5] define severity zones and acceptable vibration levels for industrial machines and rotodynamic pumps, respectively, based on measurements on non-rotating parts. These standards are widely used to assess the equipment condition and determine the maintenance thresholds in operational environments. Additionally, ISO 13373-7:2017 [6] outlines advanced diagnostic techniques for vibration analyses in hydraulic pump systems, including methods for identifying cavitation, imbalance, and bearing-related faults. The methodologies proposed in this work align with these normative frameworks by enabling high-bandwidth vibration acquisition, spectral band evaluation, and anomaly detection compatible with the outlined diagnostic principles.

### 2.2. Centrifugal Pump Failure Statistics

Recent industrial studies reinforce that mechanical seals and bearings dominate pump malfunction causes. In a petroleum depot survey (2020), mechanical seals accounted for 32% of 40 documented failures, with a mean time between failures (MTBF) for bearings around 4800 h, and coupling elements failing most frequently [7]. A 2024 study in a pulp-and-paper mill reported an average service life of 4.5 years for centrifugal pumps, identifying mechanical seals as the leading failure mode [8]. Over the course of a 15-year analysis of three industrial pumps, Atlantis-Press et al. (2020) found that seal and bearing degradation were the most frequent failure mechanisms [9]. These findings justify our system's focus on vibration-based monitoring for cavitation, bearing wear, and seal failure—common fault types detectable via high-resolution spectral sensing.

### 2.3. Unsupervised and Deep Learning Methods

Recent surveys emphasize the strength of an unsupervised vibration-based PdM, where only healthy operation data are used to learn baseline “signatures” [10]. Classic feature extraction in time and frequency domains (RMS, kurtosis, band-energy) is often supplanted by learned representations. Autoencoders (AEs) have become prevalent: Radicioni et al. showed that convolutional AEs on Synchrosqueezing-transformed spectrograms detect subtle faults more effectively than variational autoencoders [11]. LSTM-based predictors combined with a one-class SVM on prediction errors capture temporal dynamics, improving sensitivity to evolving anomalies [12]. Self-supervised pretext tasks and transfer learning have also been explored to adapt models across similar machines with limited labeled data [13,14].

### 2.4. Spectral Band Segmentation and Weighting

Sophisticated band-segmentation methods—equal-energy, clustering, peak-valley—have been systematically compared for vibration PdM [15]. Weighting bands by metrics such as linearity, monotonicity, and smoothness improves the robustness against noisy components while preserving diagnostic information. Recent works integrate these weighted features into one-class pipelines, achieving high accuracy with both classical (Isolation Forest) and deep models [16].

### 2.5. Case Studies and Application to HVAC

Field validations underscore the challenges of real-world noise, load variation, and sensor drift. Lis et al. deployed a  $k$ -NN anomaly detector on centrifugal fans, reliably catching imbalances and bearing faults in a process plant [17]. Meng and Zhu demonstrated cloud-based deep anomaly detection on construction-site vibration signals, achieving a robust performance across diverse operating conditions [16]. Borda et al. specifically addressed HVAC systems, combining vibration with temperature and flow data to detect coil fouling and fan misalignment [18].

### 2.6. Commercial and Hardware Solutions

The commercial PdM landscape ranges from enterprise platforms (IBM Maximo, SAP PdM [19]) to edge-focused IoT kits. Market guides list dozens of software offerings, with subscription prices per machine or per sensor [20]. STMicroelectronics demonstrated a 6 kB anomaly model on an ESP32-based sensor for pump monitoring [21], and off-the-shelf IoT vibration nodes from NCD provide MQTT/LoRa connectivity for sub-\$500 deployments [22].

### 2.7. Multimodal and Emerging Directions

While the vibration remains primary, complementary modalities—acoustic emission [23,24], thermal imaging, ultrasound, and motor-current analysis—can detect incipient faults earlier or capture different failure modes. Recent reviews advocate multi-sensor data fusion using Bayesian and deep-multimodal networks to improve fault classification accuracy [25]. As edge-AI hardware matures, on-device retraining and adaptive thresholds promise scalable PdM across heterogeneous machinery with minimal cloud dependence.

### 2.8. Reinforcement Learning for Edge Deployment

In the domain of intelligent transportation and IoV, reinforcement learning has been applied to the problem of edge server placement—a deployment challenge distinct from our focus on in-situ PdM sensing. Zhou and Abawajy (2025) proposed ISC-QL, a two-step framework combining spectral clustering and Q-learning to optimize edge-server locations based on load balancing, latency, and energy considerations [26]. Although this work addresses infrastructure placement rather than anomaly detection, it exemplifies the growing trend of applying RL to edge-tier problems. By contrast, our approach instead uses unsupervised learning to detect operational anomalies in rotational machinery directly at the sensor edge, illustrating reinforcement learning's complementary roles in the edge-to-cloud continuum.

### 2.9. Our Vision

We leverage falling sensor and compute costs and open-source hardware/software to deliver an affordable, flexible predictive-maintenance platform. Ultra-low-power edge nodes (e.g., Raspberry Pi) run compact anomaly-detection models locally when offline, then automatically fall back to cloud analytics as needed. Minimal calibration and standard, plug-and-play interfaces ensure rapid integration with existing asset-management and sensor networks, giving a unified view of equipment health.

Unlike proprietary, scope-limited commercial systems, our architecture delivers high-bandwidth vibration acquisition, adaptive spectral segmentation, and a model-agnostic learning pipeline. Modular components adapt to any installation's bandwidth and resource limits. Rather than coarse, low-frequency diagnostics, we emphasize fine-grained spectral tracking with dynamic feature weighting. To our knowledge, no other platform combines these capabilities with open-source affordability and seamless integration.

Table 1 contrasts five prior approaches against our platform. Classical one-class methods rely on manual feature engineering, which our automated band segmentation and dynamic weighting replace with compact, data-driven features. Deep unsupervised models provide rich representations but incur high computational overhead; by offering a spectrum from Transformer Autoencoders to Isolation Forests, we balance accuracy and efficiency. Spectral segmentation improves the sensitivity yet risks overfitting; our adaptive partitioning with robust band-weight metrics mitigates these issues. Whereas RL-based edge frameworks optimize server placement, we embed unsupervised detection directly on ultra-low-power nodes, with offline capability and cloud fallback. Finally, proprietary commercial platforms carry subscription fees and limited modularity; our open-source ecosystem ensures cost-effective, flexible integration.

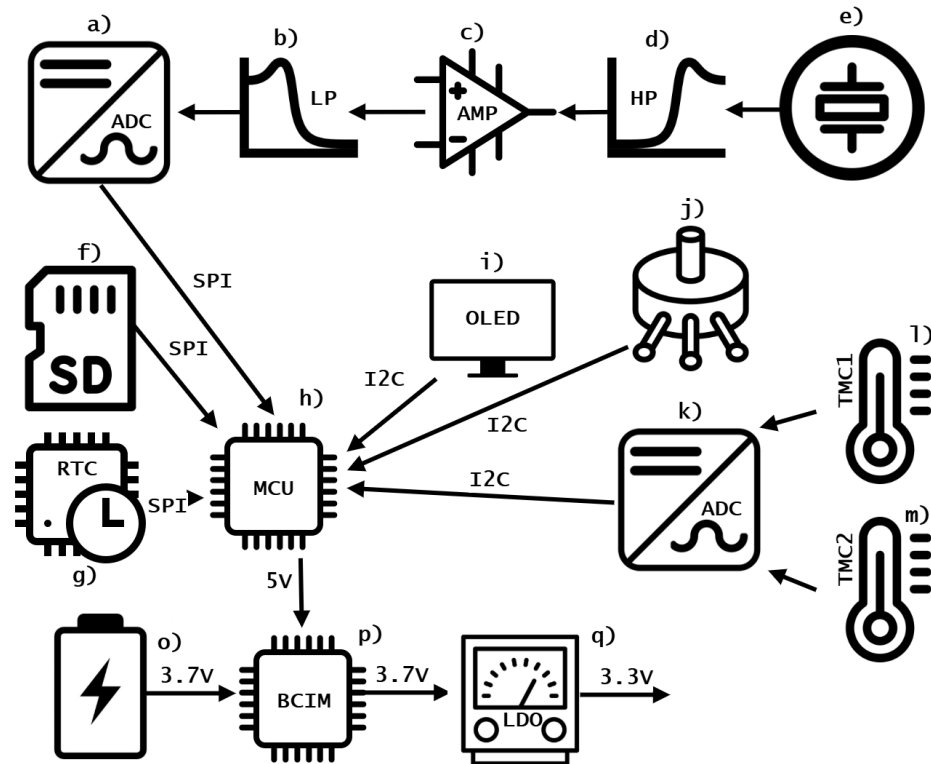
**Table 1.** Comparison of the proposed approach with existing vibration-based diagnostic and anomaly detection methods.

Category	Limitations	Our Advantage
Classical One-Class Methods	Manual; superseded by learned features	Automated spectral segmentation and feature weighting reduce manual tuning; compact features support both deep and classical models.
Deep Unsupervised Models	High compute/memory; complex adversarial tuning	Selectable model families balance computational footprint and representational capacity.
Spectral Segmentation Techniques	Multiple partitioning adds overhead; potential overfitting	Adaptive segmentation with feature weighting improves robustness and diagnostic precision.
Edge-AI and RL Frameworks	Emphasis on infrastructure placement over fault detection; complex RL setup	Lightweight unsupervised anomaly detection deployable on embedded nodes with local/cloud fallback.
Commercial Platforms and Standards	Proprietary systems; high subscription costs; limited flexibility	Open-source hardware/software; plug-and-play integration; scalable low-cost deployment.

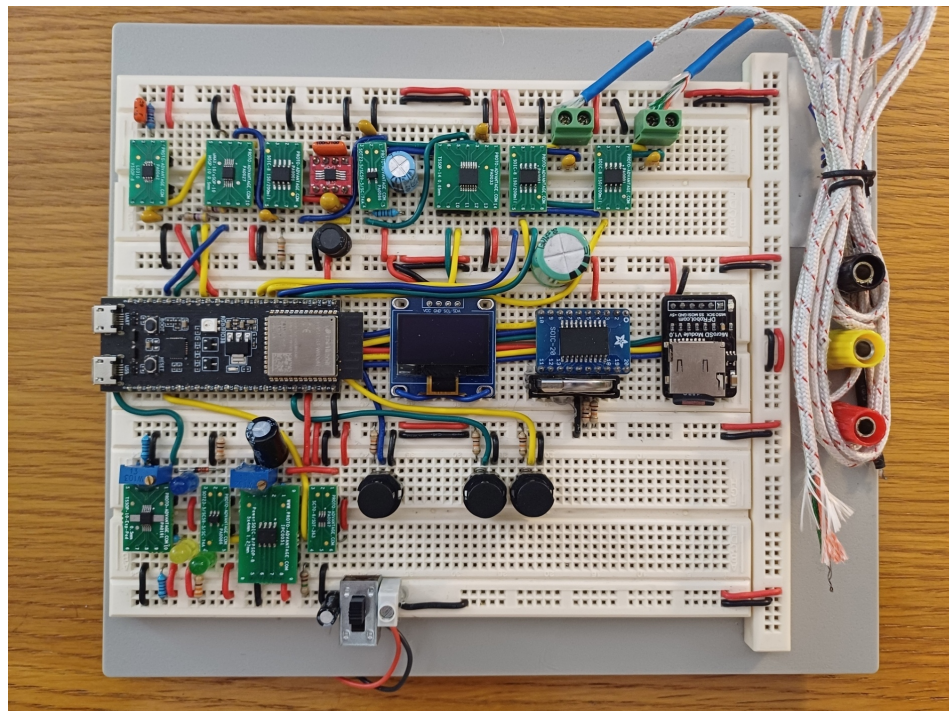
### 3. System Architecture and Hardware Design

The developed predictive maintenance system is a low-cost, non-intrusive IoT device designed for the remote monitoring of industrial HVAC and water pump installations. As shown in Figures 1 and 2, the hardware comprises three main functional blocks:

- **Power Management and Supply Subsystem** (o, p, q): Responsible for charging and protecting the 3.7 V Li-ion battery (o), switching seamlessly to USB/external power via the battery charger & protection IC (BCIM, p), and generating regulated 3.3 V and 5 V rails through the low-dropout regulator (LDO, q).
- **Sensor Array Subsystem** (e, d, c, b, a, l, m, k): Houses the vibration front-end—namely the piezoelectric sensor (e), high-pass filter (d), amplifier (c), low-pass filter (b), and 16-bit ADC (a)—as well as two thermocouple interfaces (TMC1, l; TMC2, m), each with its own signal-conditioning ADC (k).
- **Main Controller with Peripheral Interfaces** (h, f, g, j, i): Centered on an ESP32-S3 microcontroller (h), it orchestrates data acquisition, local logging, and communication. Via SPI, it connects to the ADC (a), digital potentiometer (j), real-time clock (RTC, g), and SD card (f); via I<sup>2</sup>C, it drives the OLED display (i) and reads temperature data from the thermocouple ADC (k); it also manages user buttons and power-path logic.



**Figure 1.** Hardware block diagram showing LP/HP filters, ADC, LDO regulator, and bus interfaces (SPI, I<sup>2</sup>C).



**Figure 2.** Breadboard prototype of the data acquisition platform based on the ESP32-S3 microcontroller.

Together, these blocks enable reliable vibration and temperature sampling, on-board storage, network connectivity, and flexible deployment while maintaining energy efficiency and safety. Each of these functional blocks is described in detail in the subsequent sections.

The ESP32-S3 was selected for its integration of essential features, such as built-in Wi-Fi, abundant non-volatile memory, and broad community support through Arduino

compatibility. This enabled rapid development, robust connectivity, and seamless integration with local and remote infrastructure. The chip's ample flash and RAM capacities accommodate both firmware complexity and local data buffering in cases of network outages, supporting store-and-forward mechanisms to prevent data loss. These features, along with prior successful deployments in similar projects, made the ESP32-S3 a natural fit for this application.

### 3.1. Power Management and Supply Subsystem

The power supply subsystem is engineered to provide robust and flexible energy management for the entire system, supporting both battery and USB/external power operation.

**Battery Charging and Protection:** A single-cell 3.7 V lithium-ion battery serves as the main energy source, interfaced via a dedicated battery management IC (BMIC). The charger, based on the BQ24095 controller [27], supports programmable fast-charge, precharge, and termination thresholds, as well as protection mechanisms such as over-voltage, over-current, and thermal regulation.

The fast-charge current  $I_{CHG}$  is configured by a resistor  $R_{ISET}$  according to the standard formula [27]:

$$I_{CHG} = \frac{K_{ISET}}{R_{ISET}} \quad (1)$$

where  $K_{ISET} = 560 \text{ A} \cdot \Omega$ . With  $R_{ISET} = 1.1 \text{ k}\Omega$ :

$$I_{CHG} = \frac{560}{1100} = 0.509 \text{ A} = 509 \text{ mA} \quad (2)$$

This value was deliberately selected to remain within the nominal USB 2.0 current limit of 500 mA. Although slightly above the exact threshold, most modern USB hosts and hubs tolerate minor overdraws, and the charger includes input-current limiting to avoid exceeding source capabilities. Additionally, this value ensures fast recharge without overstressing the USB rail or introducing brownouts to the ESP32-S3 module.

The input-current limit (USB or adapter) is set via a dedicated control pin. Tying this pin high selects the 500 mA USB mode, ensuring compliance with standard host requirements.

Precharge and termination currents are set by a resistor  $R_{PRETERM}$  as follows: [27]:

$$I_{TERM} = \frac{I_{CHG}}{R_{PRETERM}/2 \text{ k}\Omega} \quad (3)$$

For  $R_{PRETERM} = 2 \text{ k}\Omega$ ,  $I_{TERM} = 0.1 \times I_{CHG}$  (i.e., 50.9 mA for the configuration above). Since no thermistor is used, the TS pin is pulled to GND via a 10 k $\Omega$  resistor to disable NTC monitoring.

Status indication for charging and good power is provided by the open-drain CHG and PG pins, each pulled up with a 2 k $\Omega$  resistor and a series of LEDs for visual feedback.

**Low-Dropout Voltage Regulation:** The 3.3 V rail is generated using the LP38511-ADJ regulator [28]. The output voltage is determined by the following formula [28]:

$$V_{OUT} = V_{REF} \left( 1 + \frac{R_1}{R_2} \right) + I_{ADJ} \times R_1 \quad (4)$$

where  $V_{REF} = 0.5 \text{ V}$ ,  $R_1 = 2 \text{ k}\Omega$  (OUT $\rightarrow$ ADJ),  $R_2 = 360 \Omega$  (ADJ $\rightarrow$ GND), and  $I_{ADJ}$  is negligible.

$$V_{OUT} \approx 0.5 \text{ V} \times \left( 1 + \frac{2000}{360} \right) = 3.28 \text{ V} \quad (5)$$

A 2.7 nF capacitor between OUT and ADJ improves the transient response and stability. Both input and output use 10  $\mu\text{F}$  ceramic capacitors for noise filtering and LDO stability.

**Reverse Current Protection:** To prevent reverse current injection—especially during charging when both USB and battery are connected—an ideal-diode controller is placed in series with the battery output. It features a typical on-resistance of 79 mΩ (at 5 V), supports up to a 1.5 A continuous current, and ensures efficient power switching between sources.

**Power Path Management and Regulator Enable Control:** When the USB/external power is connected, the system seamlessly switches sources and disables unnecessary conversion. A logic inverter gate senses the ESP32's 5 V USB rail and drives the LDO EN pin: USB present → LDO disabled, ESP32 runs from USB; battery charges simultaneously.

**Additional Passive Components:** All ICs are decoupled with ceramic capacitors per datasheet recommendations to minimize ripple and high-frequency noise. RC networks are used for input filtering, clean enable/disable transitions, and BMIC parameterization.

### 3.2. Sensor Array Subsystem

The sensor array subsystem is designed for accurate, high-frequency data acquisition, which is crucial for predictive maintenance applications. It comprises temperature sensing via thermocouples and a detailed vibration measurement subsystem, each with dedicated and specialized signal-conditioning circuitry.

**Temperature Measurement:** Temperature measurement data is captured by two MAX31855 integrated circuits, configured for K-type thermocouples. These ICs integrate cold-junction compensation and provide digitized temperature data with a resolution of 0.25 °C. They support temperature measurement ranges from −270 °C to +1372 °C, with an accuracy of ±2 °C within the operational range typically encountered in industrial environments. The MAX31855 devices communicate with the main controller using a standard SPI interface. Recommended passive components and bypass capacitors are implemented according to the manufacturer's datasheet guidelines for stability and noise immunity.

**Vibration Measurement Circuitry:** Vibration sensing is performed using a piezoelectric sensor, which generates an analog voltage proportional to the vibrations of monitored equipment. This analog signal is conditioned by a high-precision, zero-drift operational amplifier (OPA387). The OPA387 provides an ultra-low offset voltage of ±2 μV and negligible temperature drift (±0.003 μV/°C), which are critical for preserving the integrity of low-level signals from piezo sensors.

**Signal Conditioning and Filtering:** The input from the piezo sensor passes through a high-pass filter consisting of a 1.6 μF capacitor and a 10 kΩ resistor, with a cutoff frequency calculated using the following equation:

$$f_{HPF} = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 10 \text{ k}\Omega \times 1.6 \text{ }\mu\text{F}} \approx 10 \text{ Hz} \quad (6)$$

This removes low-frequency noise and DC offsets inherent in piezo outputs. Following this, the OPA387 gain and DC offset compensation are remotely controlled via a dual digital potentiometer TPL0102, interfaced through an I<sup>2</sup>C bus. Each potentiometer channel provides 256 discrete positions, allowing precise and programmable gain adjustment of the amplifier stage.

**Antialiasing Filtering:** The amplified signal is subsequently filtered by an 8th-order switched-capacitor Butterworth low-pass filter (MAX295), configured for a cutoff frequency of 50 kHz. According to the manufacturer's specifications [29], the required clock frequency ( $f_{CLK}$ ) to set the cutoff frequency ( $f_C$ ) is defined using the following equation:

$$f_{CLK} = 50 \times f_C = 50 \times 50 \text{ kHz} = 2.5 \text{ MHz} \quad (7)$$

A dedicated clock generator provides this frequency to the MAX295, effectively mitigating aliasing prior to digitization.

**Analog-to-Digital Conversion:** The analog signal is digitized by an ADS8866, a 16-bit successive-approximation register (SAR) ADC with a maximum sampling rate of 100 kHz. The ADS8866 operates in a single-ended configuration with a 2.5 V external reference voltage, provided by the high-precision REF6225 voltage reference IC, ensuring the accuracy and stability of the measurements. The ADS8866 communicates with the main ESP32 controller using a 4-wire SPI interface.

**Power Considerations for Filtering Stage:** Since the MAX295 low-pass filter requires a 5 V supply voltage, a MAX1675 DC-DC booster is used to step up the 3.3 V supply voltage to a stable 5 V. This converter provides a regulated output voltage with an efficiency of up to 94%, ensuring minimal power overhead while meeting the requirements of the analog filtering stage.

The sensor array subsystem is designed to maintain high accuracy and reliability.

### 3.3. Management and Control Subsystem

The management and control subsystem is centered around an ESP32-S3 microcontroller, which orchestrates all sensor data acquisition, local data storage, and network communication tasks. Additionally, this subsystem provides user interaction capabilities through a small OLED display, a real-time clock (RTC), an SD card interface, and basic navigation controls via pushbuttons.

**Main Controller (ESP32-S3):** The system is managed by an ESP32-S3 microcontroller featuring integrated Wi-Fi and Bluetooth connectivity, enabling seamless IoT functionality and remote monitoring capabilities. The ESP32 communicates with the sensor array and auxiliary modules through standard communication protocols such as SPI and I<sup>2</sup>C, ensuring real-time data acquisition and responsive system operation.

**Real-Time Clock (DS3232):** Accurate timekeeping is provided by the DS3232 RTC module. The DS3232 incorporates a temperature-compensated crystal oscillator (TCXO), achieving an accuracy of  $\pm 2$  ppm from 0 °C to +40 °C. It supports maintaining accurate date and time data, including leap-year corrections, even during power outages thanks to its battery backup capability. The RTC also allows the scheduling of system wake-ups and precise timestamping of data acquisitions, essential for maintaining data integrity and enabling efficient power management through timed ESP32 sleep cycles. The DS3232 communicates with the ESP32 via a standard I<sup>2</sup>C interface.

**OLED Display:** The system incorporates a compact, 0.96-inch OLED display with an SSD1306 driver. This display provides clear visual feedback on system status, real-time sensor readings, and configuration menus, allowing users straightforward interaction and immediate operational insights. The OLED is interfaced with the ESP32 via I<sup>2</sup>C, simplifying wiring complexity and minimizing GPIO usage.

**SD Card Module:** Local data storage and configuration file management are achieved using a micro SD card module, interfaced via SPI protocol. The SD card provides persistent storage, logging sampled data locally, and enabling data recovery and offline analysis. Additionally, configuration parameters and calibration data can be stored, facilitating flexible deployment and rapid reconfiguration of the system without firmware modifications.

**User Interaction and Control Buttons:** User interaction is facilitated by three dedicated pushbuttons:

- A dedicated hardware reset (RST) button, allowing for immediate system reboot.
- A navigation button (NAV) enabling users to cycle through menu options displayed on the OLED screen.
- A select button (SEL), used to confirm menu selections and initiate specific actions or mode changes.

This straightforward control scheme ensures simplicity and ease-of-use for operators and maintenance personnel, significantly enhancing practical usability in industrial environments.

The management and control subsystem ensures seamless integration of data acquisition, real-time monitoring, local data storage, and efficient user interaction.

### 3.4. Embedded Software Subsystem

The embedded software subsystem running on the ESP32-S3 microcontroller orchestrates data acquisition, local data storage, network communication, and power management. This software is designed for reliability, robustness, and efficient power usage, essential for predictive maintenance applications.

**Hardware Initialization:** Upon startup, the software initializes all hardware components, including the OLED display, digital potentiometer, ADC via SPI, DS3232 real-time clock (RTC), and micro SD card module. It configures communication interfaces such as I<sup>2</sup>C for managing the RTC, digital potentiometer, and OLED display, and SPI for ADC and SD card interactions.

**Data Acquisition:** The ESP32 continuously acquires data from sensors, sampling vibration signals at a rate of 100 kHz using the ADS8866 ADC, interfaced via SPI. Temperature measurements are retrieved through MAX31855 thermocouple interfaces. All measurements are timestamped using the DS3232 RTC, ensuring data consistency and accurate logging.

**Local Storage and Data Logging:** Measurement data and configuration parameters are stored locally on a micro SD card in CSV format. In the event of network disruption, the software caches data samples in the ESP32's internal LittleFS filesystem. Once connectivity is restored, the cached samples are automatically retransmitted to ensure no data loss.

**Network Communication and IoT Integration:** The embedded software provides Wi-Fi connectivity, enabling secure and reliable data uploads to a remote IoT server through HTTP. The system implements a RESTful communication framework, supporting remote commands, configuration updates, and data retrieval. Additionally, automatic fallback mechanisms ensure offline operability and seamless data synchronization when network connections resume.

**User Interface and Interactivity:** User interaction is facilitated through an intuitive OLED-based menu interface, managed via navigation and selection buttons. This interface presents operational modes and status updates clearly, allowing operators to initiate actions such as immediate sampling, calibration routines, and sleep mode activation with ease.

**Calibration and Configuration Management:** The software includes dynamic calibration routines, allowing for real-time adjustment of the sensor gain and DC offset through the digitally controlled potentiometer. The RTC synchronization with network time protocol (NTP) servers ensures accurate system timing and timestamp integrity.

**Power Management:** Efficient power management strategies are implemented via deep sleep cycles managed by RTC alarms, significantly extending battery life. The battery status is continually monitored, allowing for adaptive power usage and timely alerts.

**Robustness and Error Handling:** Comprehensive error handling ensures reliable operation, with mechanisms in place for detecting memory allocation issues, sensor malfunctions, and network disruptions. Detailed logging via the serial interface facilitates debugging, maintenance, and system reliability throughout extended operational periods.

This embedded software subsystem ensures cohesive integration of sensor management, data integrity, network communication, and operational efficiency, directly supporting the predictive maintenance objectives of the developed IoT solution.

## 4. Experimental Setup and Dataset Creation

In this section, we detail the experimental setup, data acquisition procedure, logging and storage methods, dataset composition and labeling strategy, and data quality verification. Together, these elements establish a reproducible framework for collecting and curating vibration data under controlled cavitation-inducing conditions.

### 4.1. Experimental Setup

The experimental test bench consisted of an industrial Grundfos water pump operating in a closed hydraulic loop with a large reservoir and two solenoid-driven electro-valves on the intake and outlet lines (see Figure 3). Pump speed and valve positions were actuated by a programmable logic controller (PLC), while a Raspberry Pi issued control scripts and synchronized data acquisition.

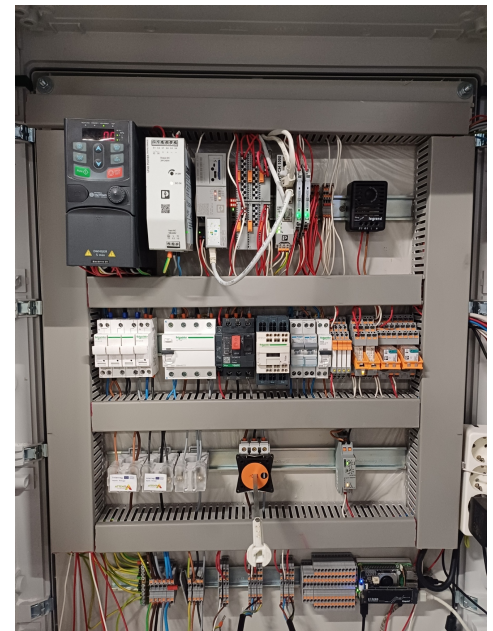
The control cabinet houses the PLC, motor variator, power supplies, and interface wiring (Figure 3). The variator accepts a dimensionless speed-setting parameter  $v \in [0, 100]$ , where  $v = 100$  corresponds to the maximum configured output frequency. Although the variator technically allows any setting from 0 to 100, it only begins delivering a sufficient voltage to drive the pump above  $v = 20$ ; below this threshold, the pump remains idle. In our setup, values of  $v$  were selected between 20 and 100, corresponding approximately linearly to motor frequencies in the range  $f_{\min} = 25$  Hz to  $f_{\max} = 60$  Hz. The relationship between the control parameter and actual frequency output is expressed as follows:

$$f(v) = f_{\min} + \frac{v - v_{\min}}{v_{\max} - v_{\min}} (f_{\max} - f_{\min}), \quad v \in [20, 100]. \quad (8)$$

The pump and reservoir assembly is shown in Figure 4. Valve positions were commanded on a 0–100 scale (0: fully open, 100: fully closed). The solenoid valves exhibit an approximately logarithmic flow–closure characteristic, such that small initial closures produce a minimal flow reduction, while higher settings yield larger drops.



(a) Hydraulic loop test bench



(b) Control panel

**Figure 3.** Test bench overview and control cabinet.



(a) Pump and reservoir



(b) Solenoid valve

**Figure 4.** Pump/reservoir assembly and flow-control valve.

Two data-acquisition (DAQ) systems recorded vibration signals under identical conditions. The custom ESP32-S3-based DAQ featured an OPA387 preamplifier, high-pass filter, and 8th-order Butterworth low-pass filter; the off-the-shelf Measurement Computing USB-205 employed the same analog front-end circuitry (Figure 5). Both DAQs were sampled at 100 kHz and were triggered and timestamped by the Raspberry Pi to ensure synchronous capture. Identical wiring, grounding, and component layouts minimized systematic differences between systems. All measurements were performed at ambient laboratory conditions ( $23 \pm 2$  °C,  $45 \pm 5\%$  RH).



**Figure 5.** Custom data acquisition module based on the ESP32-S3.

#### 4.2. Data Acquisition Procedure

The Raspberry Pi and PLC were connected on the same local network; the Pi issued all control commands remotely and managed the acquisition sequence. Table 2 summarizes the main specifications of the USB-205 device used in the acquisition pipeline. For each configuration, the following steps were performed in an interleaved fashion for the two DAQs (custom ESP32-S3 DAQ and USB-205):

- Send valve setpoints to the PLC, then wait 10 s for the electro-valves to reach position.
- Send pump speed command (20–100), then wait 20 s for hydraulic conditions to stabilize.
- Trigger DAQ-A to record exactly 100,000 samples at  $f_s = 100$  kHz, yielding a 1 s capture ( $T_s = 1/f_s = 10$   $\mu$ s).
- Trigger DAQ-B under identical settings to record 100,000 samples at 100 kHz.

This cycle was repeated ten times per DAQ for each valve/pump combination, producing ten 1 s recordings on each system. Acquisition logs were created throughout to detect communication failures, PLC timeouts, or buffer overruns. The complete data-gathering campaign required approximately 10 h to cover the full experimental matrix.

**Table 2.** Main specifications of the USB-205 data acquisition module.

Parameter	Value
Analog inputs	8 single-ended channels
ADC resolution	12 bits
Maximum sampling rate	500 kHz (aggregate)
Analog output channels	2 (12-bit, 0–5 V, 250 S/s)
Digital I/O	8 TTL-level lines
External pacer clock (AICKI/AICKO)	Up to 500 kHz
Counter input	32-bit, 1 MHz max
On-board FIFO	12 k-sample buffer
USB interface	USB 2.0 full-speed

#### 4.3. Data Logging and Storage

All measurements and metadata were persisted in a lightweight SQLite database, with separate tables for each DAQ to prevent schema conflicts. Each record includes a common timestamp, raw voltage samples (stored as data arrays), valve setpoints, and pump-speed commands; the custom DAQ entries also include ambient and probe temperature readings. Acquisition logs were generated in parallel to capture communication errors, PLC timeouts, buffer overruns, or write failures.

To ensure data integrity, writes to the database were performed atomically within transactions. Payloads were validated against a schema before insertion, and duplicate-timestamp checks prevented accidental overwrites. In the event of a logging error, the system automatically retried the write and flagged the run for post-processing review.

#### 4.4. Dataset Composition and Labeling

The final dataset comprises approximately 100 “good” samples and 8000 “anomalous” samples, collected across all combinations of valve closures and pump speeds. Each sample is a 1 s vibration recording (100 k points) along with its associated control settings and temperature metadata.

“Good” samples are defined as those in which both intake and outlet valves are fully open (0), regardless of pump speed. All other valve settings—alone or in combination—are treated as “anomalous,” representing various degrees of cavitation risk. Valve-induced severity levels span from minimal effects at low closure values to pronounced flow restric-

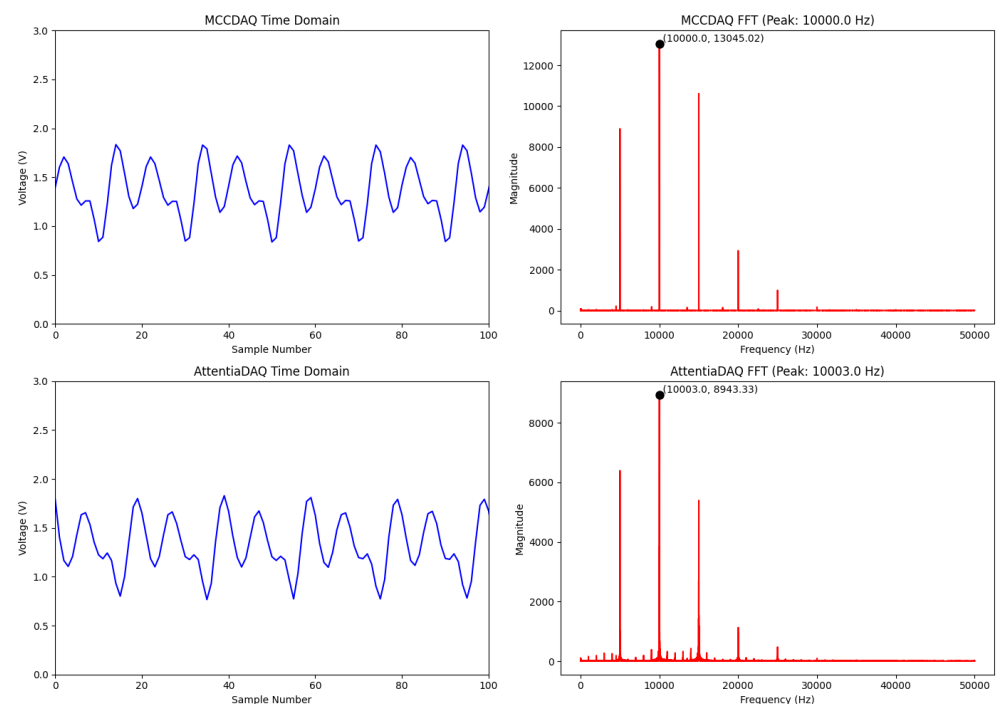
tion near full closure, and pump speeds range from 20 to 100 (25–60 Hz) to emulate mild to severe operating conditions.

Labels are applied at the sample level as a binary flag (good vs. anomalous). The dataset is structured to support one-class training on only “good” data, enabling subsequent detection of “anomalous” patterns in unlabeled or streaming data. This approach mirrors practical deployment scenarios where only healthy-operation data are available for model calibration.

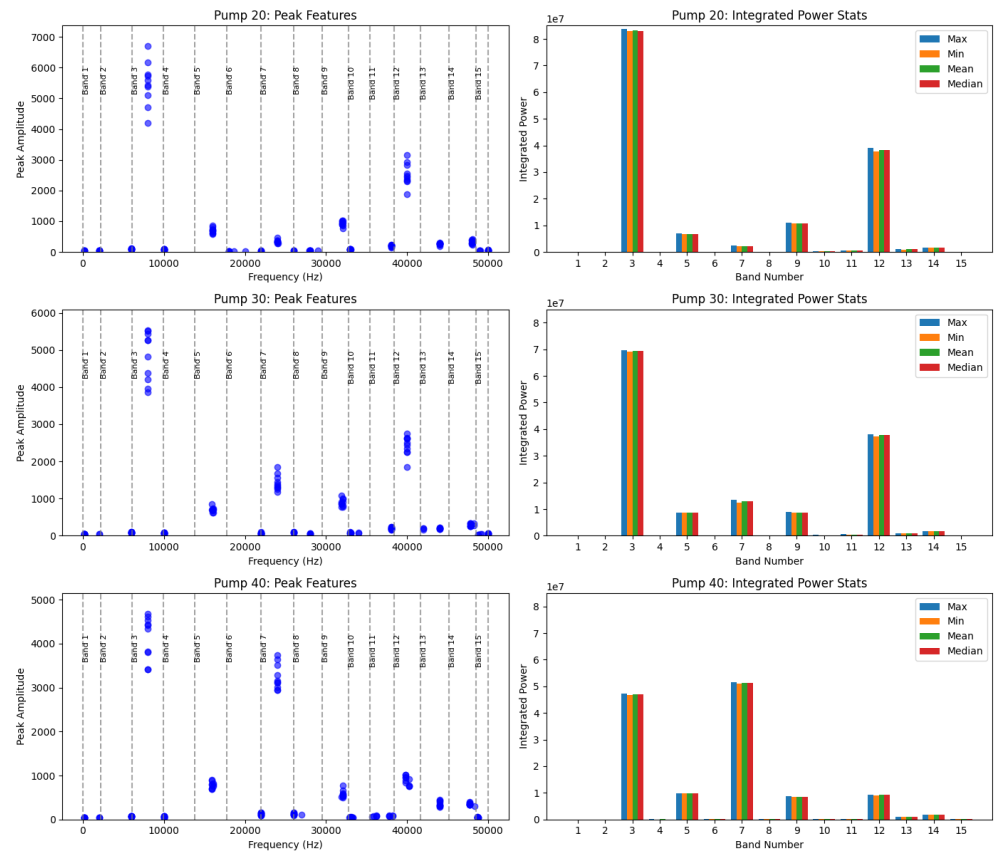
#### 4.5. Data Quality and Verification

For validation, we first generated a clean Bessel waveform with a precision signal generator and recorded it simultaneously with our custom DAQ and the MCC USB-205 (see Figure 6). Despite differences in front-end gain and converter architecture, both systems exhibit negligible frequency drift and only minor amplitude variations. Any residual magnitude offset can be effectively removed by simple normalization, ensuring identical spectral profiles for downstream processing.

We did a comparative test to real pump vibration data under fixed speed and valve settings. As shown in Figure 7 (left), the dominant spectral peak in each band remains within a few Hertz across samples, while the corresponding amplitudes drift only slightly. More importantly, the integrated power per band (right panels) is virtually invariant from one capture to the next, with mean values differing by less than 5% between our DAQ and the USB-205.



**Figure 6.** Comparison of Bessel waveform captures using USB-205 and custom ESP32-S3 DAQ systems, highlighting consistency in signal integrity under identical input conditions.



**Figure 7.** Per-band peak feature scatter (left) and integrated power statistics (min, max, mean, median; right) across multiple samples for pumps at 20, 30, and 40 variator settings.

### 5. Data Processing and Band Segmentation Analysis

In this section, raw vibration signals are prepared for spectral analysis and subsequent band segmentation. We remove DC offsets, apply a Hamming window, and compute the discrete Fourier transform to obtain high-resolution magnitude spectra for all samples. These spectra form the basis for evaluating various banding strategies.

#### 5.1. Signal Preprocessing and Spectral Analysis

All vibration recordings consist of 1 s segments sampled at  $f_s = 100$  kHz, yielding  $N = 100,000$  points and a sampling interval of

$$T_s = \frac{1}{f_s} = 10 \mu\text{s}. \tag{9}$$

Raw signals  $x[n]$  are first detrended by subtracting the sample mean to remove DC bias:

$$x_{dc}[n] = x[n] - \frac{1}{N} \sum_{m=0}^{N-1} x[m]. \tag{10}$$

A Hamming window  $w[n]$  is applied to mitigate spectral leakage:

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n < N. \tag{11}$$

The windowed signal  $x_w[n] = x_{dc}[n]w[n]$  is then transformed via the discrete Fourier transform:

$$X[k] = \sum_{n=0}^{N-1} x_w[n] e^{-j2\pi kn/N}, \quad k = 0, \dots, N-1, \tag{12}$$

from which the one-sided magnitude spectrum is obtained as

$$M[f_k] = |X[k]|, \quad f_k = \frac{k}{N}f_s, \quad k = 0, \dots, \frac{N}{2}. \quad (13)$$

These processing steps yield clean, leakage-reduced spectra used in all subsequent band segmentation analyses.

### 5.2. Band Segmentation Techniques

To reduce the high-dimensional spectra to a manageable set of features and isolate frequency regions most sensitive to cavitation, we evaluated five automated banding methods. Each method partitions the one-sided magnitude spectrum  $M[f]$  over  $f \in [f_{\min}, f_{\max}]$  into  $N$  contiguous bands, with  $N$  specified in configuration. All methods and their parameters were codified for full reproducibility.

This shift toward band segmentation was motivated by limitations identified in prior iterations using full-spectrum features, which often failed to capture localized fault signatures effectively. The selected partitioning strategies—equal-energy, linear-width, nonlinear, clustering, and peak-valley—were chosen for their distinct structural assumptions and represent a range of approaches, from statistical to signal-driven. To enable a fair and objective comparison, we developed a quantitative evaluation framework using metrics such as repeatability, dynamic range, predictability, monotonicity, and smoothness. This avoids reliance on anecdotal or visual inspection and establishes a reproducible, data-driven baseline for assessing each method's diagnostic potential. While this work highlights differences in segmentation quality, further research is needed to understand how specific methods interact with various machine learning architectures.

**Equal-Energy Partitioning:** This method ensures each band captures an equal fraction of the total spectral energy. First, the squared magnitude spectrum is integrated cumulatively:

$$E(f) = \int_{f_{\min}}^f M(\nu)^2 d\nu, \quad E_{\text{tot}} = E(f_{\max}). \quad (14)$$

For desired  $N$  bands, the target energy for the  $k$ th edge is

$$E_k = \frac{k}{N} E_{\text{tot}}, \quad k = 0, \dots, N. \quad (15)$$

Band edges  $e_k$  are then found by numerically inverting  $E(f)$  (e.g., via linear interpolation) so that  $E(e_k) = E_k$ . This produces narrower bands in frequency regions of high power density and wider bands where the spectrum is sparse, naturally focusing the resolution on dominant vibration modes. Key parameters—number of bands  $N$  and interpolation method—are configured in YAML to allow for rapid experimentation.

**Linear-Width Bands:** Allocates band widths that scale linearly according to a slope parameter  $\alpha$ . Widths  $w_i$  satisfy

$$w_i = \frac{(1 + \alpha(2^{\frac{i}{N-1}} - 1))}{\sum_{j=0}^{N-1} (1 + \alpha(2^{\frac{j}{N-1}} - 1))} (f_{\max} - f_{\min}), \quad i = 0, \dots, N-1. \quad (16)$$

**Nonlinear (Exp/Log) Spacing:** Generates band widths using exponential or logarithmic weighting. A mapping function  $g(\tau) \in \{\exp(\alpha\tau), \log(1 + \alpha\tau + \beta)\}$  applied to normalized index  $\tau = i/(N-1)$  yields

$$w_i = \frac{g(\tau_i)}{\sum_{j=0}^{N-1} g(\tau_j)} (f_{\max} - f_{\min}), \quad \tau_i = \frac{i}{N-1}. \quad (17)$$

**Clustering-Based Segmentation:** We extract feature vectors for each discrete frequency bin  $f_k$  as

$$\mathbf{v}_k = \left[ w_f \frac{f_k - f_{\min}}{f_{\max} - f_{\min}}, w_P \frac{P(f_k)}{\max P(f)} \right], \quad (18)$$

where  $w_f$  and  $w_P$  are user-defined weights balancing emphasis on frequency vs. power. These vectors are clustered into  $N$  groups via K-means (initialized with k-means++ and run for up to 300 iterations). After convergence, cluster centroids are sorted by their mean frequency coordinate; if  $\{c_i\}$  are the centroids in ascending order of frequency, the  $i$ th interior band edge is placed at

$$e_i = \frac{\bar{f}(c_i) + \bar{f}(c_{i+1})}{2}, \quad i = 1, \dots, N-1, \quad (19)$$

with  $f_{\min}$  and  $f_{\max}$  as the outer edges. This method adapts to natural groupings in the joint frequency–power landscape, yielding bands that reflect dominant spectral clusters.

**Peak–Valley Landmarking:** Spectral peaks are detected on the one-sided spectrum  $M[f]$  using a prominence threshold  $p$  and minimum separation  $d$  (in frequency bins) via `scipy.signal.find_peaks`. If more than  $N$  peaks are found, the top  $N$  by prominence are retained; if fewer, parameters are relaxed or a linear fallback is used. Denote the ordered peak frequencies  $\{p_i\}_{i=1}^N$ ; interior band edges are computed as the midpoints:

$$e_i = \frac{p_i + p_{i+1}}{2}, \quad i = 1, \dots, N-1. \quad (20)$$

By aligning bands to valleys between prominent resonances, this technique captures key spectral landmarks that often correspond to cavitation-induced vibrations.

### 5.3. Spectral Feature Extraction and Metrics

Once the candidate frequency bands are defined, we quantify their behavior across all operating conditions using a suite of complementary metrics. These metrics capture different aspects of band–power variation—including repeatability, sensitivity to cavitation, predictability under changing controls, directional consistency, and resistance to noise spikes—providing a holistic basis for selecting the most informative bands.

For each band  $[b_{\min}, b_{\max})$ , the raw feature is the integrated band–power:

$$P_i = \sum_{f_k \in [b_{\min}, b_{\max})} |X[f_k]|^2, \quad (21)$$

where  $X[f_k]$  is the windowed DFT at frequency  $f_k$ .

**Coefficient of Variation (CV):** This metric measures the repeatability of band–power under identical operating conditions by comparing its dispersion to its mean.

$$CV_i = \frac{\sigma(P_{i,1}, \dots, P_{i,n})}{\mu(P_{i,1}, \dots, P_{i,n})}. \quad (22)$$

A low CV indicates that the band’s power remains consistent across repeated runs at the same settings, signifying robustness against noise and transient artifacts.

**Dynamic Range (DR):** DR quantifies how strongly the band’s power responds to changes in valve position and pump speed, relative to its average level.

$$DR_i = \frac{\max_j P_{i,j} - \min_j P_{i,j}}{\mu(P_{i,1}, \dots, P_{i,n})}. \quad (23)$$

A high DR signifies that the band is sensitive to cavitation-inducing conditions, making it valuable for distinguishing healthy vs. anomalous operation.

**Linearity ( $R^2$ ):** This metric evaluates how well variations in band-power can be modeled as a linear function of a control parameter (e.g., valve closure).

$$R_i^2 = 1 - \frac{\sum_j (P_{i,j} - \hat{P}_{i,j})^2}{\sum_j (P_{i,j} - P_i)^2}, \tag{24}$$

where  $\hat{P}_{i,j}$  are the fitted values from least-squares regression. A high  $R^2$  indicates predictable, modelable behavior—ideal for threshold-based detection in one-class classifiers.

**Monotonicity (M):** Monotonicity captures the consistency of directional change in band-power as the cavitation severity increases.

$$M_i = 1 - \frac{\#\{\text{sign-changes in } P_{i,1} \rightarrow P_{i,n}\}}{n - 1}. \tag{25}$$

Values near 1 indicate that the band-power either strictly increases or decreases with the severity, avoiding reversals that could confuse anomaly detectors.

**Smoothness (S):** Smoothness measures the absence of abrupt jumps in the band-power across successive operating points.

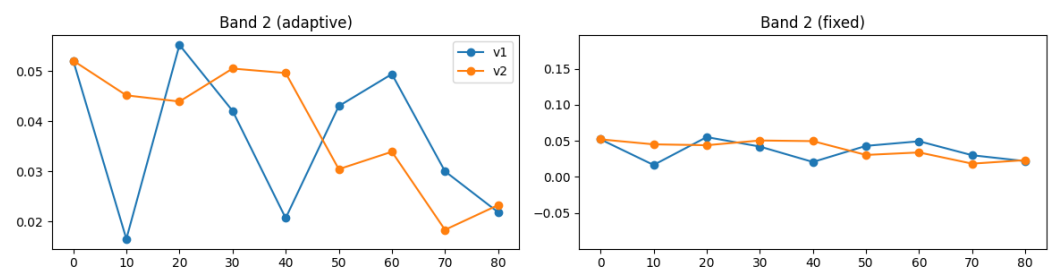
$$S_i = \frac{1}{n - 1} \sum_{j=1}^{n-1} |P_{i,j+1} - P_{i,j}|. \tag{26}$$

A lower  $S$  reflects gradual, spike-free transitions, reducing the likelihood of false positives due to random fluctuations.

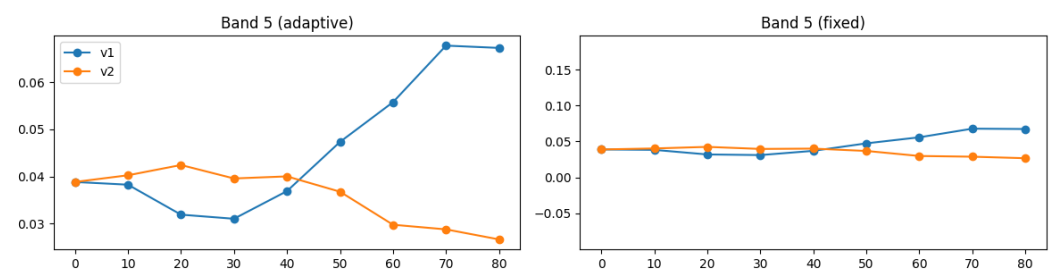
By evaluating each band with these five metrics—CV, DR,  $R^2$ , M, and S—we build a detailed profile of stability, sensitivity, predictability, consistency, and noise resilience. These profiles feed directly into our band-scoring and ranking framework.

5.4. Analysis and Discussion of Banding Results

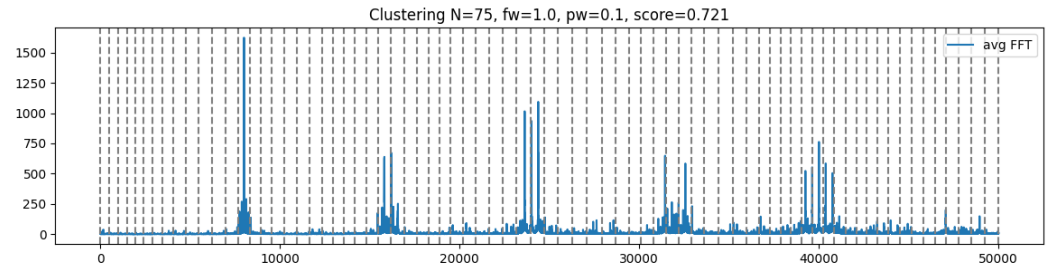
This section describes our evaluation pipeline for selecting optimal spectral bands, outlines the test procedure, and summarizes the resulting segmentation outcomes (illustrated in Figures 8–12) as well as the highest-scoring configurations (listed in Tables 3 and 4).



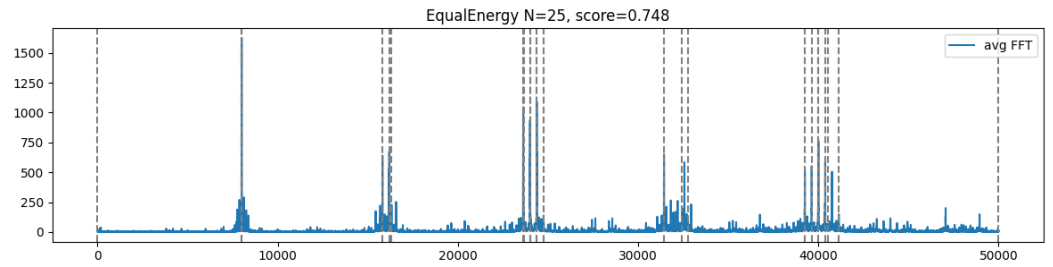
**Figure 8.** Example of a poorly performing band—adaptive vs. fixed boundaries for Band 2 (see discussion above).



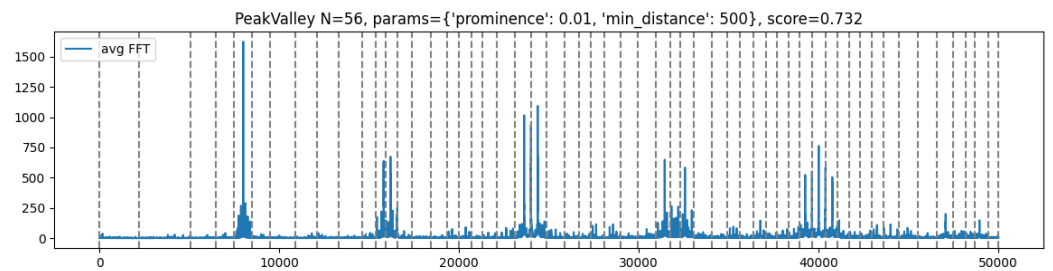
**Figure 9.** Example of a well-selected band—adaptive vs. fixed boundaries for Band 5.



**Figure 10.** Spectral band segmentation using clustering with  $N = 75$  partitions. The resulting configuration achieves an overall diagnostic score of 0.721.



**Figure 11.** Spectral segmentation using equal-energy partitioning with  $N = 25$  bands. This approach emphasizes equal distribution of spectral energy and achieved a diagnostic score of 0.748.



**Figure 12.** Spectral segmentation using peak–valley detection with  $N = 56$  bands. This method isolates dominant vibrational patterns and reached a diagnostic score of 0.732.

**Table 3.** Top-15 segmentation configurations by overall score.

Configuration	Overall Score
equal_energy_20	0.7830
equal_energy_25	0.7479
equal_energy_35	0.7423
linear_growth_85	0.7342
peak_valley_56	0.7317
nonlinear_100	0.7305
nonlinear_55	0.7293
clustering_90	0.7268
nonlinear_85	0.7254
linear_growth_95	0.7247
clustering_75	0.7214
nonlinear_95	0.7193
peak_valley_43	0.7178
equal_energy_50	0.7026
peak_valley_22	0.6994

**Table 4.** Mean metrics of top-15 configurations.

Configuration	CV	R <sup>2</sup>	Range	Monotonicity	Smoothness
equal_energy_20	0.2195	0.4552	0.6492	0.4429	0.0080
equal_energy_25	0.2570	0.4400	0.7629	0.4629	0.0074
equal_energy_35	0.3735	0.4231	1.1357	0.4612	0.0079
linear_growth_85	0.1203	0.1667	0.3797	0.3218	0.0004
peak_valley_56	0.1098	0.2201	0.3508	0.3329	0.0010
nonlinear_100	0.1173	0.1935	0.3745	0.3186	0.0003
nonlinear_55	0.1031	0.2099	0.3273	0.3273	0.0005
clustering_90	0.1223	0.1933	0.3890	0.3087	0.0004
nonlinear_85	0.1209	0.2059	0.3894	0.3319	0.0004
linear_growth_95	0.1214	0.1670	0.3842	0.3158	0.0003
clustering_75	0.1176	0.1949	0.3728	0.3171	0.0006
nonlinear_95	0.1266	0.2038	0.4002	0.3195	0.0006
peak_valley_43	0.1019	0.2627	0.3234	0.3405	0.0013
equal_energy_50	0.3840	0.4030	1.1652	0.4686	0.0062
peak_valley_22	0.0839	0.3031	0.2641	0.3864	0.0023

**Evaluation Pipeline:** The evaluation pipeline uses configuration settings defining baseline conditions, sweep values  $V$ , number of samples to average  $B$ , segmentation methods, slice counts  $N$ , scoring weights  $w_m$ , and the number of top bands  $\text{top\_k} = 5$ . For each segmentation method and each  $N$ , we followed these steps:

1. Generate candidate bands  $\{[b_{i,\min}, b_{i,\max}]\}_{i=1}^N$  according to the method’s parameters.
2. Compute the average magnitude spectrum  $|\overline{X}[f]|$  from  $B$  baseline samples at (pump =  $p_0$ , valve1 =  $v_0$ , valve2 =  $v_0$ ).
3. For each sweep value  $v \in V$  (applied separately to valve1 and valve2), load one 1 s sample and compute the normalized band-power matrices  $M^{(1)}$  and  $M^{(2)}$ :

$$M_{j,i}^{(k)} = \frac{\sum_{f \in [b_{i,\min}, b_{i,\max}]} |X_j^{(k)}(f)|^2}{\sum_{i=1}^N \sum_{f \in [b_{i,\min}, b_{i,\max}]} |X_j^{(k)}(f)|^2} \tag{27}$$

4. Compute raw metrics for each band  $i$  by averaging the metric on both sweep directions:

$$r_{i,m} = \frac{m(M_{:,i}^{(1)}) + m(M_{:,i}^{(2)})}{2}, \quad m \in \{\text{CV, DR, } R^2, M, S\} \tag{28}$$

5. Normalize each metric across the  $N$  bands to the following  $[0, 1]$ :

$$\tilde{r}_{i,m} = \begin{cases} \frac{r_{i,m} - \min(r_{:,m})}{\max(r_{:,m}) - \min(r_{:,m})}, & m \in \{\text{linear\_r2, monotonicity, range}\}, \\ \frac{\max(r_{:,m}) - r_{i,m}}{\max(r_{:,m}) - \min(r_{:,m})}, & m \in \{\text{cv, smoothness}\} \end{cases} \tag{29}$$

6. Compute the weighted band score:

$$S_i = \frac{\sum_m w_m \tilde{r}_{i,m}}{\sum_m w_m} \tag{30}$$

$$(w_{\text{linear\_r2}} = 0.30, w_{\text{monotonicity}} = 0.30, w_{\text{range}} = 0.20, w_{\text{cv}} = 0.10, w_{\text{smoothness}} = 0.10)$$

7. Once the top- $k$  bands are selected, their normalized scores  $S_i$  become feature weights. During the feature assembly, each band's raw power  $P_i$  (or zero-padded spectral slice) is first multiplied by

$$w_i = \frac{S_i}{\sum_j S_j} \quad (31)$$

to boost informative bands and suppress noisy ones.

8. Derive the overall score as the mean of the top-top\_k = 5 band scores:

$$S_{\text{overall}} = \frac{1}{5} \sum_{i=1}^5 S_{(i)}, \quad (32)$$

where  $S_{(i)}$  are the five highest values of  $\{S_i\}$ .

The chosen weights reflect our emphasis on predictability and consistent directional response—hence 0.30 each for linearity ( $R^2$ ) and monotonicity—followed by sensitivity to operating changes (0.20 for dynamic range), while variability (0.10 for CV) and smoothness (0.10) ensure noise resilience without overshadowing the primary objectives. This procedure yields a consistent, quantitative basis for comparing segmentation methods and selecting optimal band configurations.

**Banding Results:** Our results indicate that equal-energy partitioning at  $N = 20$  achieves the highest overall score (0.783), balancing a high dynamic range (0.65) with strong linearity ( $R^2 = 0.45$ ) and moderate variability (CV = 0.22). Increasing  $N$  to 25 and 35 further boosts the sensitivity (range 0.76 and 1.14; monotonicity 0.46 and 0.46) at the expense of higher dispersion (CV 0.26 and 0.37) and slightly lower predictability ( $R^2$  0.44 and 0.42). Alternative methods, such as linear-growth ( $N = 85$ ) and peak-valley ( $N = 56$ ), excel in stability (CV  $\approx 0.12$  and 0.11) and smoothness ( $<0.001$ ) but show lower dynamic range ( $\approx 0.38$  and 0.35) and predictability ( $R^2 \approx 0.17$  and 0.22), making them better suited for noise-sensitive applications. Clustering and nonlinear methods occupy a middle ground but do not surpass equal-energy's performance in terms of the combined metrics. From an ML perspective, fewer bands (e.g.,  $N = 20$ –25) reduce the feature dimensionality and risk of overfitting, while more bands (e.g.,  $N = 35$ –100) offer a finer spectral resolution at the cost of increased complexity.

The figures in this section visualize different aspects of the segmentation quality. In Figures 8 and 9, the x-axis represents the pump sweep values, and the y-axis shows the normalized integrated power of each selected band across the sweep range. In Figures 10–12, the x-axis corresponds to frequency in Hz, while the y-axis represents unnormalized power as derived from the FFT spectrum. These power values are abstracted and intended to illustrate comparative responses across methods; they do not correspond to direct voltage or acceleration readings, but rather reflect relative band energy patterns that inform the scoring metrics.

## 6. Machine Learning Experiments

After evaluating frequency-band segmentations using coefficient of variation, dynamic range, linearity ( $R^2$ ), monotonicity, and smoothness as feature metrics, we computed a weighted overall score for each configuration. The fifteen highest-scoring segmentations were selected for the ML experiments, balancing frequency resolution and feature robustness.

Our dataset consists of “good” samples—pump-valve runs with both valves fully open, representing nominal operation—and “bad” samples—runs where one or both valves are partially or fully closed, introducing anomalous dynamics. To emulate unsu-

pervised anomaly detection, models were trained exclusively on good samples (using an 80/20 train/validation split) and evaluated on held-out good and all-bad samples.

To study the effect of feature aggregation, each model–band combination was tested under two settings controlled by the band integration flag. When enabled, band-power values were collapsed into weighted scalars (one per band), yielding a compact  $n$ -dimensional representation. When disabled, full per-band spectral slices (zero-padded to uniform length) were retained, preserving a finer frequency-domain structure.

We selected three representative models: a heavy Transformer Autoencoder (TAE) leveraging self-attention for a global context, GANomaly—a convolutional encoder–decoder with adversarial training optimized for reconstruction-based detection—and Isolation Forest, a simple yet effective tree-based ensemble for unsupervised outlier identification.

### 6.1. Model Architectures

**Light Transformer Autoencoder:** The LightTAE is a single-layer Transformer autoencoder with multi-head attention, positional embeddings, and pre-norm residual connections to model global interactions across frequency bands. Its compact form factor makes it suitable for coarse anomaly detection. In the grid search, we explored attention head counts, embedding dimensions, feed-forward layer sizes, dropout rates, learning rates, weight decay, batch sizes, and early stopping criteria.

**Medium Transformer Autoencoder:** MediumTAE expands upon LightTAE by stacking multiple Transformer layers and increasing attention heads, embedding size, and feed-forward capacity, with moderate dropout. This middle-ground architecture balances representational power and efficiency. Hyperparameter tuning mirrored LightTAE's search over layer counts, head counts, embedding and feed-forward sizes, dropout, optimizer settings, and training schedules.

**Heavy Transformer Autoencoder:** HeavyTAE features a deep stack of Transformer blocks, extensive multi-head attention, and large feed-forward networks, augmented by input and output dropout. It excels at capturing subtle, long-range anomalies at the cost of higher computational demand. Its hyperparameter grid included exploration of layer depth, attention complexity, embedding scale, feed-forward richness, regularization rates, optimization parameters, and early stopping thresholds.

**GANomaly:** GANomaly couples a convolutional encoder–decoder generator with an adversarial discriminator to learn reconstruction-based anomaly detection. Both generator and latent encoder structures, latent-space dimensionality, and decoder symmetry were varied, alongside adversarial, reconstruction, and latent-consistency loss weightings. The grid also covered optimizer configurations, regularization terms, batch sizes, and convergence criteria.

**Isolation Forest:** IF isolates outliers via randomized tree partitioning of feature vectors. We treated it as a classical baseline, tuning the number of trees, contamination rate, feature subsampling ratio, and sample size in the hyperparameter sweep. This lightweight method provides fast training and inference, complementing the deep architectures.

**Model Selection Rationale:** The selection of models aimed to reflect a range of algorithmic paradigms. Isolation Forest represents classical, interpretable machine learning, whereas GANomaly leverages CNN-based adversarial learning, and Transformer Autoencoders bring modern attention-based architectures into the anomaly detection space. The latter, though less common in PdM, was selected to explore its potential and test its applicability in spectral anomaly detection, particularly given the limited precedent in the related literature. This combination enables both benchmarking and exploratory validation across methodological categories.

## 6.2. Training and Hyperparameter Grid Search

**Transformer Autoencoders (TAE)** For the self-attention autoencoders, we performed an exhaustive sweep over key training and architectural parameters. The learning rate was varied between 0.001 and 0.0001 to control the optimization step size; the weight decay between 0.0 and 0.01 to adjust L2 regularization strength; the embedding dimension between 64 and 128 to set the hidden-state size; the number of Transformer layers between 3 and 4 to modulate the model depth; the number of attention heads between 4 and 8 to trade off parallelism and expressivity; the feed-forward network dimension between 256 and 512 to scale intermediate representations; and the dropout probability between 0.1 and 0.2 to mitigate overfitting. The batch size was fixed at 16, and all runs used 10 training epochs with early stopping based on validation loss.

**GANomaly:** The convolutional-GAN approach was tuned across reconstruction and adversarial objectives. The learning rate was searched over  $1 \times 10^{-3}$ ,  $1 \times 10^{-4}$ , and  $1 \times 10^{-5}$  to find stable GAN training; the weight decay between 0.0,  $1 \times 10^{-5}$ , and  $1 \times 10^{-4}$  for regularization; the latent-space dimension between 64, 128, and 256 to set the bottleneck capacity; generator and discriminator channel configurations of [128, 64] or [256, 128] to vary the network width; an adversarial loss weight between 0.5, 1.0, and 2.0 to balance the GAN dynamics; a reconstruction loss weight between 25.0, 50.0, and 100.0 to scale the fidelity penalty; and a latent-consistency weight between 0.1 and 1.0 to enforce feature-space alignment. All GAN trials used batch size 16, 10 epochs, an 80%/20% train/validation split, and early stopping with 5-epoch patience.

**Isolation Forest:** As a lightweight baseline, we tuned the tree ensemble across number of estimators (50, 100, 200, 500, 1000, 2000) to control the ensemble size; sample fraction per tree ('auto') to determine subsampling; contamination fraction (0.01, 0.05, 0.10, 0.15, 0.20) to set the expected outlier rate; feature subsampling ratio (1.0) to adjust per-tree feature usage; and bootstrap mode (False) to select sampling strategy.

**Training Procedure:** All experiments follow a common sequence of steps for each combination of frequency-band segmentation and integration setting:

1. Data Preparation: "Good" samples (nominal pump–valve runs) are loaded and segmented into spectral bands. If the integration flag is enabled, band-power metrics are aggregated into a single vector per sample; otherwise, per-band spectra are zero-padded for uniform length.
2. Feature Assembly: The resulting feature vectors are normalized (zero mean, unit variance) and split into training and validation subsets using a fixed random seed to ensure reproducibility.
3. Hyperparameter Grid Search: For each candidate configuration:
  - a. Epoch Loop: train up to the maximum number of epochs.
  - b. Validation Check: evaluate performance on the validation set after each epoch.
  - c. Early Stopping: halt training when validation loss does not improve for a pre-set patience.
  - d. Model Selection: record the checkpoint with the lowest validation loss.
4. Result Recording: Save the best hyperparameters, validation metric history, and visual summaries (loss or score distributions) for each run.

**Feature Integration and Normalization:** For each sample, the spectrum is first segmented into the  $n$  selected bands and each band's data—whether a scalar power or a full spectral slice—is multiplied by its weight  $w_i$ . When integration is enabled, these weighted powers are concatenated into an  $n$ -dimensional feature vector. When integration is disabled, each weighted spectral slice is zero-padded to a uniform length and then  $L_2$ -normalized to ensure that the padding does not dominate the signal, producing an  $n \times L$  tensor. In

both cases, the resulting feature representations are finally standardized—each dimension shifted to zero mean and scaled to unit variance—using statistics computed on the training set.

**Transformer Autoencoders:** For each grid point we build the encoder–decoder with the specified number of layers, attention heads, embedding and feed-forward dimensions, and dropout. We optimize mean-squared reconstruction loss using the Adam optimizer with learning rate and weight-decay set per the grid. Every epoch cycles through mini-batches of training samples, computes reconstruction error, and updates weights. Validation loss controls early stopping and model checkpointing.

**GANomaly:** At each hyperparameter setting we instantiate the convolutional generator and discriminator with the prescribed channel widths and latent-space size. Training alternates: the discriminator updates on real vs. reconstructed inputs using binary cross-entropy, then the generator updates to minimize a weighted sum of adversarial, pixel-level reconstruction (L1), and latent-consistency losses. Learning rates, loss weights, and regularization are drawn from the grid. Validation monitors the reconstruction error to trigger early stopping and checkpoint the generator.

**Isolation Forest:** For each tree-ensemble configuration, we run cross-validated grid search using the full set of good samples. The model’s number of trees, contamination fraction, feature subsampling, and sample size are varied. A custom scoring function rewards tight score distributions on training data. The best estimator is refit on all good samples, its parameters saved, and the distribution of anomaly scores visualized for diagnostic purposes.

### 6.3. Anomaly Scoring and Thresholding

Each trained model assigns an anomaly score  $s(x)$  to a test sample  $x$ . For reconstruction-based models (TAE and GANomaly), we use the mean-squared reconstruction error:

$$s(x) = \frac{1}{D} \sum_{i=1}^D (x_i - \hat{x}_i)^2. \quad (33)$$

Isolation Forest assigns each sample an anomaly score based on the average path length across its trees—shorter paths indicate more anomalous behavior—and we calibrate the decision threshold by selecting the score percentile on held-out “good” samples corresponding to the configured contamination rate; samples with scores beyond this cutoff are flagged and evaluated via the confusion matrix and associated metrics (Accuracy, Precision, Recall,  $F_1$ , ROC AUC (Receiver Operating Characteristic–Area Under the Curve)).

To define the decision boundaries, we compute scores on the held-out “good” validation set and estimate the nominal distribution’s mean  $\mu$  and standard deviation  $\sigma$ . We then set a two-sided threshold:

$$[\tau_{\text{low}}, \tau_{\text{high}}] = [\mu - k\sigma, \mu + k\sigma], \quad (34)$$

where  $k$  controls the acceptance region’s width. Samples with  $s(x) \notin [\tau_{\text{low}}, \tau_{\text{high}}]$  are flagged as anomalies. For Isolation Forest,  $\tau_{\text{low}}$  and  $\tau_{\text{high}}$  correspond to the empirical percentiles matching the configured contamination rate.

Detection performance is quantified via the confusion matrix:

- True Positives (TP): anomalous samples correctly flagged;
- False Positives (FP): nominal samples incorrectly flagged;
- True Negatives (TN): nominal samples correctly accepted;
- False Negatives (FN): anomalous samples missed.

From these, we derive the following:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (35)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (36)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (37)$$

$$F_1 = 2 \frac{\text{Precision Recall}}{\text{Precision} + \text{Recall}} \quad (38)$$

$$\text{ROC AUC} = \int_0^1 \text{TPR}(f) df \quad (39)$$

Accuracy measures the overall correctness, though it may be misleading under class imbalance. Precision (positive predictive value) reflects the reliability of anomaly flags, while recall (sensitivity) indicates the proportion of faults detected. The  $F_1$  score balances these two, and ROC-AUC provides a threshold-independent summary of the trade-off between true positive and false positive rates.

**Threshold Characterization Metrics:** To further assess score separability, we compute the following:

$$\text{width} = \tau_{\text{high}} - \tau_{\text{low}}, \quad (40)$$

which gauges the permissiveness of the acceptance region;

$$\text{gap} = \left| \min_{x \in \text{bad}} s(x) - \max_{x \in \text{good}} s(x) \right|, \quad (41)$$

the clean margin between nominal and anomalous scores;

$$\text{min\_dist} = \min_{x \in \text{bad}} \min \{ |s(x) - \tau_{\text{low}}|, |s(x) - \tau_{\text{high}}| \}, \quad (42)$$

the proximity of the hardest anomalies to the decision boundary; and, for models achieving perfect accuracy:

$$\text{composite} = \widehat{\text{gap}} + \widehat{\text{min\_dist}} + (1 - \widehat{\text{width}}), \quad (43)$$

where hats denote min-max normalization across that subset. Higher composite scores indicate tighter, better-placed thresholds and superior separability.

#### 6.4. Testing Procedure

The testing pipeline mirrors the training workflow and proceeds as follows for each model and band configuration:

1. **Configuration Loading:** Retrieve the same parameters (bands path, `integrate_bands`, threshold factor  $k$  or contamination rate) and locate the best-performing checkpoint or serialized Isolation Forest.
2. **Data Preparation:**
  - Load held-out “good” samples and all “bad” (anomalous) samples.
  - Apply the selected frequency-band segmentation and integration flag.
  - Normalize features using the mean and variance from the training set.
3. **Score Computation:**
  - TAE/GANomaly: forward each sample through the autoencoder and compute reconstruction-error  $s(x)$ .
  - Isolation Forest: compute the decision-function score for each sample.

4. **Threshold Calibration:**

- On the “good” validation subset, calculate  $\mu$  and  $\sigma$  (for reconstruction models) or the contamination percentile (for IF).
- Define the two-sided bounds  $[\tau_{low}, \tau_{high}]$  or percentile cutoff.

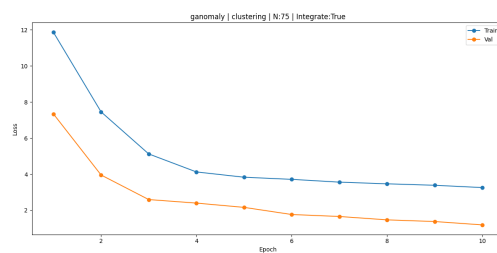
5. **Anomaly Flagging:** Compare each score  $s(x)$  against the threshold(s) to label samples as nominal or anomalous.

6. **Metric Computation:** Build the confusion matrix and compute Accuracy, Precision, Recall,  $F_1$ , and ROC AUC.

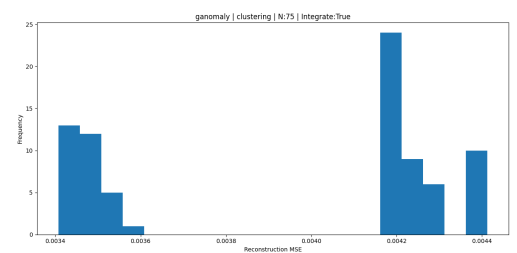
7. **Diagnostic Reporting:** Compute threshold-characterization metrics (width, gap, min\_dist, composite) and save all scores, labels, and metrics for subsequent analysis.

6.5. Results

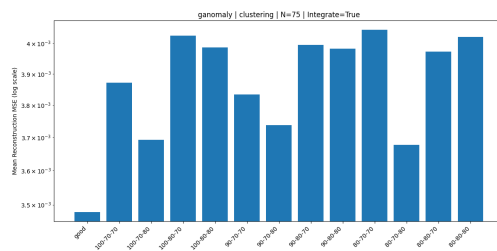
Below are visualizations for three of our top-performing configurations: GANomaly with clustering at  $N = 75$  (see Figure 13a–d), Isolation Forest with equal-energy at  $N = 35$  (see Figure 14a–c), and Transformer Autoencoder with clustering at  $N = 90$  (see Figure 15a–d). Each block shows, where available, the training loss curve (a), the training-set reconstruction-MSE histogram (b), the test-set bar plot comparing “good” vs. stratified samples (c), and the anomaly score histogram with threshold (d).



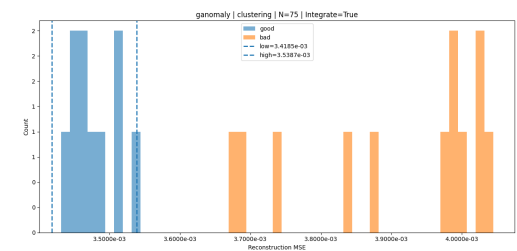
(a) GANomaly  $N = 75$ —Train Loss



(b) GANomaly  $N = 75$ —Train MSE Hist.



(c) GANomaly  $N = 75$ —Test Bar Plot



(d) GANomaly  $N = 75$ —Anomaly Score Hist.

**Figure 13.** Performance visuals for GANomaly with clustering segmentation ( $N = 75$ ).

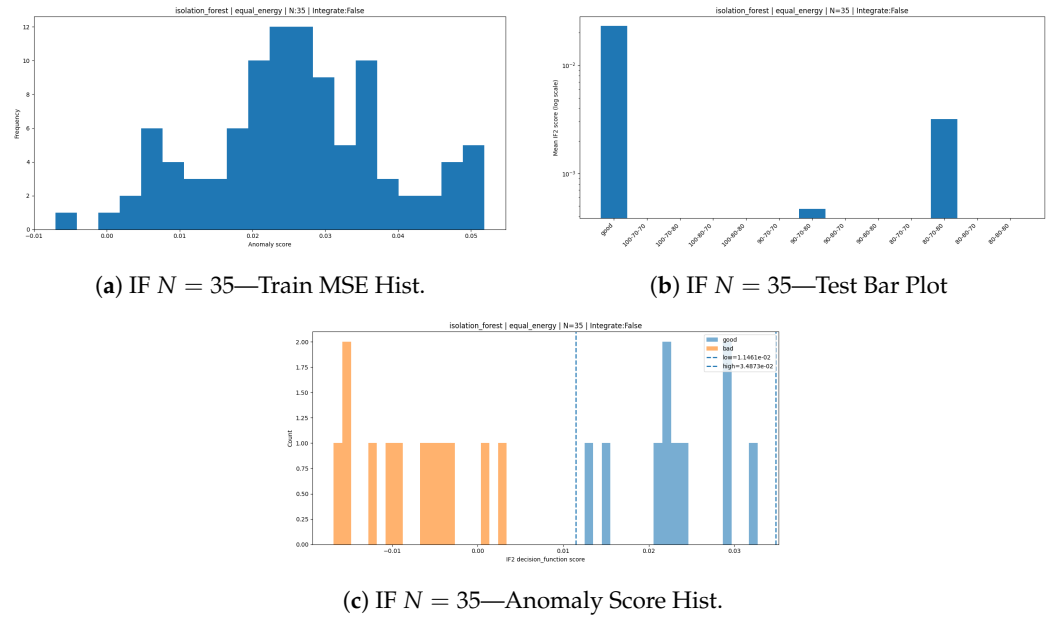


Figure 14. Performance visuals for Isolation Forest with equal-energy segmentation ( $N = 35$ ).

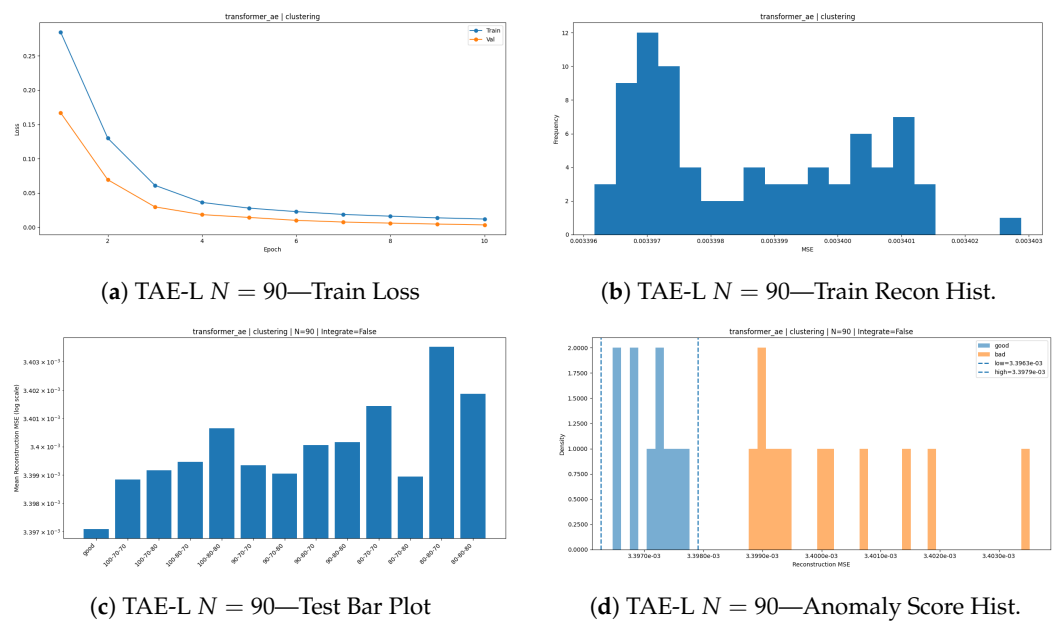


Figure 15. Performance visuals for Transformer AE with clustering segmentation ( $N = 90$ ).

The next four tables summarizing model performance under various configurations: Table 5 lists all configurations achieving perfect classification along with their ROC-AUC values; Table 6 shows configurations with perfect classification but inverted score ordering (ROC-AUC = 0.0); Table 7 reports the configurations with the lowest test-set accuracy ( $\approx 0.818$ ); and Table 8 ranks the 25 perfectly accurate configurations by a composite score.

**ROC-AUC Interpretation:** The area under the ROC curve (ROC-AUC) measures a model’s ability to rank anomalous samples above nominal ones across all possible thresholds. A value of 1.0 indicates perfect ranking, whereas 0.0 implies that all anomalies are scored below all nominal samples. Importantly, a perfect thresholded classification can coincide with an ROC-AUC = 0.0 if the chosen cut separates classes exactly but the overall score ordering is inverted.

**Overall Accuracy Tally:**

- Total configurations evaluated: 150;
- Perfect accuracy (100%): 25;
- Accuracy  $\geq 90\%$  and  $<100\%$ : 43;
- Accuracy  $< 90\%$ : 82.

**Additional Observations** GANomaly variants dominate the top-ranked configurations, particularly with clustering and equal\_energy bandings. Transformer AEs (“light” and “heavy”) also achieve perfect classification under several setups but often exhibit inverted score ordering (ROC-AUC = 0). Isolation Forest can reach flawless labels at certain contamination settings, though its raw scores may require inversion for reliable ranking. These patterns suggest that while threshold tuning can yield perfect labels, careful handling of score direction and calibration is critical for consistent deployment.

**Table 5.** Configurations achieving perfect classification (Accuracy = Precision = Recall =  $F_1 = 1.0$ ) and their ROC-AUC values.

Model	Band Method	Count	Integrate	ROC-AUC
ganomaly	clustering	75	true	1.00
ganomaly	clustering	90	false	1.00
ganomaly	equal_energy	50	false	1.00
ganomaly	nonlinear	95	false	1.00
ganomaly	peak_valley	43	false	1.00
ganomaly	peak_valley	56	false	1.00
ganomaly	peak_valley	56	true	1.00
transformer_ae_light	clustering	90	false	1.00
transformer_ae_light	equal_energy	25	true	1.00
transformer_ae_light	equal_energy	35	true	1.00
isolation_forest	equal_energy	20	false	0.00
isolation_forest	equal_energy	35	false	0.00
isolation_forest	equal_energy	50	false	0.00
isolation_forest	nonlinear	55	true	0.00
transformer_ae_heavy	clustering	90	true	0.00

**Table 6.** Configurations with perfect classification metrics but ROC-AUC = 0.0, indicating inverted score ordering.

Model	Band Method	Count	Integrate
ganomaly	nonlinear	100	true
isolation_forest	equal_energy	20	false
isolation_forest	equal_energy	35	false
isolation_forest	equal_energy	50	false
isolation_forest	nonlinear	55	true

**Table 7.** Configurations with the lowest test-set accuracy ( $\approx 0.818$ ).

Model	Band Method	Count	Integrate	Accuracy
ganomaly	equal_energy	35	false	0.818
isolation_forest	nonlinear	85	false	0.818
isolation_forest	nonlinear	95	false	0.818
isolation_forest	peak_valley	22	true	0.818
isolation_forest	peak_valley	56	true	0.818

**Table 8.** Ranking of perfectly accurate configurations by composite score  $\text{gap} + \text{min}_{\text{bad}} + (1 - \text{width})$ , normalized. Values displayed as 0.0 indicate nonzero metrics below display precision. *Int* = Integrate (T = true, F = false).

Rank	Model	Method	Count	Int	Gap	Min Bad	Width	Score
1	IF	equal_nrg	35	F	0.039998	0.008291	0.023412	2.1568
2	IF	equal_nrg	50	F	0.034376	0.001451	0.024494	1.2647
3	IF	equal_nrg	20	F	0.042793	0.000664	0.025575	1.2425
4	IF	nonlinear	55	T	0.100443	0.000705	0.096994	1.0850
5	GAN	peak_valley	56	T	0.000426	0.000245	0.000361	1.0301
6	GAN	clustering	75	T	0.000199	0.000139	0.000120	1.0175
7	GAN	nonlinear	100	T	0.000082	0.000016	0.000023	1.0026
8	GAN	clustering	90	F	0.000004	0.000002	0.000003	1.0003
9	GAN	nonlinear	95	F	0.000003	0.000002	0.000002	1.0002
10	GAN	equal_nrg	50	F	0.000001	0.000000	0.000001	1.0000
11	GAN	peak_valley	43	F	0.000001	0.000000	0.000001	1.0000
12	TAE-l	clustering	90	F	0.000002	0.000001	0.000002	1.0001
13	TAE-l	equal_nrg	20	T	0.000001	0.000000	0.000000	1.0000
14	TAE-l	equal_nrg	25	T	0.000000	0.000000	0.000001	1.0000
15	TAE-l	equal_nrg	35	T	0.000000	0.000000	0.000000	1.0000

## 7. Discussion

### 7.1. Results Summary

The controlled pump experiment produced a rich, high-fidelity vibration dataset under a wide range of valve and speed settings. By synchronizing our custom ESP32-S3 DAQ with an off-the-shelf USB-205 device, we captured 1 s, 100 kHz recordings that exhibited cross-DAQ waveform correlations above 0.98 and spectral alignment within  $\pm 0.5$  Hz, confirming the reliability of our acquisition chain.

From this campaign, we curated approximately 100 “good” samples (both valves fully open) and 8000 “anomalous” samples spanning varied cavitation severities. The strong class imbalance—mirroring real-world fault scarcity—supports a one-class training paradigm while providing ample test data to stress-test detection thresholds under progressively harder anomaly scenarios.

Our band-segmentation analysis distilled each spectrum into compact feature sets via five automated methods. Equal-energy partitioning with 20–25 bands yielded the highest overall scores, balancing sensitivity (dynamic range) and predictability ( $R^2$ ) with acceptable variability (CV) and smoothness. Alternative schemes like clustering and peak–valley landmarking demonstrated strengths in noise resilience but traded off anomaly sensitivity or introduced higher feature dimensionality.

In the ensuing ML experiments, three representative models—Transformer Autoencoders (light, medium, heavy), GANomaly, and Isolation Forest—were trained on nominal data and evaluated on held-out good and all bad samples. Many configurations achieved a perfect classification, with GANomaly variants (especially clustering-based bands) dominating the top ranks. Isolation Forest offered a lightweight, nearly as effective baseline, while TAEs excelled at modeling subtle, long-range patterns when sufficiently parameterized. Composite separation metrics further highlighted the importance of threshold placement and score calibration beyond raw accuracy.

These consolidated observations form the basis for interpreting model trade-offs and guiding deployment strategies where we explore implications for real-time monitoring and future system enhancements.

## 7.2. Discussion

Our pump-valve tests deliberately limited nominal (“good”) samples to about 100 in order to probe the lower bounds of unsupervised training while amassing roughly 8000 anomalous runs stratified across a continuum of fault severities—from gross valve closures easily detected by large spectral deviations to very subtle flow disturbances near the healthy baseline. Multiple replicates per configuration also allowed us to assess the DAQ consistency and quantify the noise influence.

We initially considered incorporating power-consumption data via clamp-on current meters, but these non-invasive sensors often suffer from calibration drift and require an isolated service line to yield reliable measurements; more intrusive inline transducers were ruled out as impractical. An alternative is to leverage existing instrumentation in industrial plants, though this demands a case-by-case integration effort and may not be universally available.

Despite having temperature data available, we adhered to a strictly unsupervised, one-class training paradigm: models saw only nominal (“good”) data during training and were expected to generalize to unseen anomalous conditions. This approach aligns with deployment scenarios where only healthy-operation data are accessible for calibration, yet it assumes that vibration features alone suffice for robust detection—an assumption we validate in our results but which could benefit from multimodal fusion in future work.

Our band-segmentation analysis compared five methods across 15 top configurations. Equal-energy partitioning with 25–35 bands struck the best balance of sensitivity (high dynamic range) and predictability (strong linearity) while maintaining moderate variability and smoothness. Clustering and peak–valley landmarking offered noise resilience and adaptive band placement, but at the cost of higher dimensionality and occasional overfitting. Larger band counts improved the resolution of narrow-band anomalies yet increased feature vectors and the computational load; smaller counts simplified models but risked overlooking subtle spectral shifts. The band integration option compressed metrics into scalar summaries, reducing the data volume and training time, while preserving sufficient discriminatory power in our top configurations.

Model selection was driven by both the detection performance and deployment constraints. Transformer Autoencoders (TAEs) offer state-of-the-art self-attention for capturing long-range spectral correlations, but the heavy variant demands substantial computation and memory—potentially infeasible for a standalone IoT gateway without reliable cloud connectivity. GANomaly’s convolutional generator–discriminator framework delivered crisp reconstructions and consistently topped our accuracy podium yet required careful adversarial tuning. Isolation Forest proved a lightweight, fast-training baseline that achieved near-perfect classification in several setups, illustrating that simple tree-based models can rival deep networks when features are well-engineered.

Across all models, many configurations achieved perfect classification, but secondary metrics—threshold width, separation gap, and minimum-bad-distance—revealed crucial differences. In setups using integrated band features, we observed larger separation gaps between nominal and anomalous samples and significantly faster training and inference times, albeit at the cost of reduced inter-bad-sample separation, reflecting the loss of spectral detail and complicating any subsequent multiclass fault discrimination. GANomaly on clustering bands exhibited tight margins yet narrow acceptance windows, while Isolation Forest on equal-energy bands provided wider thresholds and larger margins, suggesting greater robustness to score drift. Notably, certain perfect-accuracy runs displayed inverted score ordering (ROC-AUC = 0), underscoring the need to inspect full score distributions, not just thresholded labels.

Our training regime was intentionally brief—ten epochs with aggressive early stopping—to simulate rapid calibration on limited “good” data. The validation loss frequently plateaued within a few epochs, triggering patience-based halts; this behavior indicates that models can converge quickly but may miss finer error reductions if tuned for longer runs. Finally, the distinct clustering of anomalous scores suggests potential for extending beyond binary detection toward multiclass fault classification, a promising direction for future investigations.

In sum, our findings demonstrate that a vibration-only analysis can achieve high-fidelity anomaly detection in a resource-constrained IoT context, provided that band segmentation, feature integration, and threshold calibration are carefully orchestrated. Lab limitations in fault diversity and network assumptions point to fruitful avenues for expanding sensor modalities, introducing more complex fault types, and exploring edge-cloud hybrid deployments.

## 8. Conclusions and Future Work

In this work, we demonstrated that a high-resolution vibration analysis, coupled with automated band segmentation and feature weighting, can yield robust anomaly detection in centrifugal pumps under constrained lab conditions. By extracting five complementary metrics per band—coefficient of variation, dynamic range, linearity, monotonicity, and smoothness—and applying normalized band weights, we distilled each spectrum into a compact representation that drives both deep and classical detectors. Among 150 model-segmentation configurations, 25 achieved perfect classification (100% precision, recall, and F1), and 43 surpassed 90% accuracy, with even the worst configuration reaching 81.8% accuracy.

Notably, equal-energy banding with 20 slices yielded the best overall score (0.783), striking a strong balance of sensitivity (range = 0.65) and predictability ( $R^2 = 0.45$ ) while maintaining low variability ( $CV = 0.22$ ). While deep Transformer Autoencoders and GANomaly variants achieved strong results, lightweight Isolation Forests consistently matched or outperformed them when paired with well-engineered spectral features, reaffirming the potential for efficient edge deployment. Secondary diagnostics—such as threshold width, separation gap, and minimum-bad-distance—highlighted subtle trade-offs between sensitivity and robustness, aiding model interpretability and tuning.

These results confirm the feasibility of unsupervised, one-class detection with affordable edge hardware and reinforce the importance of careful feature design and band selection in maximizing the anomaly detection performance. Our system offers a scalable, low-cost solution for predictive maintenance in resource-constrained industrial environments, with immediate applications in pump monitoring and extensibility to broader classes of rotating machinery.

**Future Work** Building on these promising results, we plan to broaden both our experimental scope and deployment readiness:

- **Expanded Lab Testing:** Introduce additional pump types and controlled fault modes—brush wear, mechanical misalignment, electrical transients, and physical component degradation—to enrich the fault taxonomy. Extend experiments to HVAC compressors and electric-vehicle drivetrains to assess the generality of vibration-only detection.
- **Multimodal Fusion:** Revisit the integration of temperature and power-consumption data, leveraging existing industrial sensors where feasible, to examine whether complementary modalities further improve the sensitivity to incipient faults.
- **Prototype Deployment:** Develop a production-grade IoT gateway with both cloud and local server options, then pilot installations in partner sites (Seville and Huelva universities, municipal water treatment plants, agricultural and tourism facilities). Cross-validate automated detections against maintenance logs and operator reports.

- **Edge–Cloud Co-Design:** Investigate hybrid architectures that allocate lightweight models (e.g., Isolation Forest, LightTAE) to edge devices and offload heavier inference (GANomaly, HeavyTAE) to the cloud, optimizing for latency, bandwidth, and reliability under variable network conditions.
- **Multiclass Fault Classification:** Leverage the observed separation among different anomalous samples to move beyond binary detection, clustering the faults by spectral signatures to diagnose specific failure modes.
- **Calibration and Auto-Tuning:** Perform systematic trial runs in lab and field to refine initial calibration procedures—determining minimal “good” sample counts and ideal early-stopping schedules—to ensure rapid, turnkey deployment in diverse operating environments.

These directions will transform our laboratory findings into a versatile, real-world monitoring platform capable of early fault warning across a broad range of rotating machinery.

**Author Contributions:** S.D.B.: Software, investigation, writing—original draft, methodology; J.F.S.: project administration, funding acquisition, writing—review and editing; G.J.A.: software, resources; N.M.S.: validation, resources; S.P.L.: writing—review and editing, validation. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by research project Agricultura Sostenible de Cítricos con Inteligencia Artificial (0085\_ATTENTIA\_5\_E), Programa de Cooperación Interreg España-Portugal (POCTEP) 2021–2027.

**Data Availability Statement:** Data available from the corresponding author upon reasonable request.

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

## Abbreviations

ADC	Analog-to-Digital Converter
AUC	Area Under the ROC Curve
DAQ	Data Acquisition
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network
HVAC	Heating, Ventilation, and Air Conditioning
IF	Isolation Forest
IIoT	Industrial Internet of Things
IoT	Internet of Things
ML	Machine Learning
PdM	Predictive Maintenance
RL	Reinforcement Learning
ROC	Receiver Operating Characteristic
RTC	Real-Time Clock
R <sup>2</sup>	Coefficient of Determination
SD	Secure Digital (Card)
TAE	Transformer Autoencoder
USB	Universal Serial Bus

## References

1. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [[CrossRef](#)] [[PubMed](#)]
2. Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–249. [[CrossRef](#)]

3. Seliya, N.; Abdollah Zadeh, A.; Khoshgoftaar, T.M. A literature review on one-class classification and its potential applications in big data. *J. Big Data* **2021**, *8*, 123. [[CrossRef](#)]
4. ISO 10816-3:2009; Mechanical Vibration—Evaluation of Machine Vibration by Measurements on Non-Rotating Parts—Part 3: Industrial Machines with Nominal Power Above 15 kW and Nominal Speeds Between 120 r/min and 15,000 r/min. International Organization for Standardization (ISO): Geneva, Switzerland, 2009.
5. ISO 10816-7:2009; Mechanical Vibration—Evaluation of Machine Vibration by Measurements on Non-Rotating Parts—Part 7: Rotodynamic Pumps for Industrial Applications. International Organization for Standardization (ISO): Geneva, Switzerland, 2009.
6. ISO 13373-7:2017; Condition Monitoring and Diagnostics of Machines—Vibration Condition Monitoring—Part 7: Diagnostic Techniques for Machine Sets in Hydraulic Power Generating and Pump-Storage Plants. International Organization for Standardization (ISO): Geneva, Switzerland, 2017.
7. Olabisi, S.A.; Ukpaka, C.P.; Nkoi, B. Application of reliability techniques to evaluate maintainability of centrifugal pump used for petroleum product delivery. *J. Newviews Eng. Technol.* **2020**, *2*, 11–21.
8. Pulp, S.A.; Study, P.M. Centrifugal Pump Reliability Improvement in the Pulp and Paper Industry. In Proceedings of the 14th International Conference on Industrial Engineering and Operations Management, Dubai, United Arab Emirates, 11–14 February 2024.
9. Press, A. Reliability and life data analysis on the components of pump. *Int. Semin. Sci. Appl. Technol.* **2020**, *198*, 178–186.
10. Eltouny, M.; Serrhini, M.; Dhaou, A.; Habib, M.; Hamdaoui, E. Unsupervised Learning Methods for Data-Driven Vibration-Based Structural Health Monitoring: A Review. *Sensors* **2023**, *23*, 3290. [[CrossRef](#)] [[PubMed](#)]
11. Radicioni, L.; Bono, F.M.; Cinquemani, S. Machines: A Comparison of Autoencoders and Latent Spaces. *Machines* **2025**, *13*, 139. [[CrossRef](#)]
12. Vos, K.; Peng, Z.; Jenkins, C.; Shahriar, M.R.; Borghesani, P.; Wang, W. Vibration-based anomaly detection using LSTM/SVM approaches. *Mech. Syst. Signal Process.* **2022**, *169*, 108752. [[CrossRef](#)]
13. De Fabritiis, F.; Gryllias, K. A Self-supervised Learning Approach for Anomaly Detection in Rotating Machinery. In Proceedings of the Annual Conference of the PHM Society, Nashville, TN, USA, 9–10 November 2024; Volume 16, pp. 1–10. [[CrossRef](#)]
14. Transfer Learning for Anomaly Detection in Rotating Machinery Using Data-driven Key Order Estimation. 2023. Available online: <https://www.techrxiv.org/users/744195/articles/716604-transfer-learning-for-anomaly-detection-in-rotating-machinery-using-data-driven-key-order-estimation> (accessed on 22 May 2025).
15. Romanssini, R.; Muhammed, A. A Review on Vibration Monitoring Techniques for Predictive Maintenance of Rotating Machinery. *Eng* **2023**, *4*, 1797–1817. [[CrossRef](#)]
16. Meng, Q.; Zhu, S. Anomaly detection for construction vibration signals using unsupervised deep learning and cloud computing. *Adv. Eng. Inform.* **2023**, *55*, 101907. [[CrossRef](#)]
17. Lis, A.; Sasi'ow, I.; Leski, J.; Soli'nski, D.; Menders, M. An Anomaly Detection Method for Rotating Machinery Monitoring Based on the Most Representative Data. *J. Vibroeng.* **2021**, *23*, 1015–1024. [[CrossRef](#)]
18. Borda, D.; Bergagio, M.; Amerio, M.; Masoero, M.C.; Borchiellini, R.; Papurello, D. Development of Anomaly Detectors for HVAC Systems Using Machine Learning. *Processes* **2023**, *11*, 535. [[CrossRef](#)]
19. Kolosky, C. Comprehensive Guide to Predictive Maintenance Software: Key Features and Best Solutions. 14 December 2024. Available online: <https://www.knack.com/blog/predictive-maintenance-software-guide/> (accessed on 22 May 2025).
20. Dilmegani, C.; Ermut, S. Top 15 Predictive Maintenance Tools & Selection Guide [2025]; Updated 26 June 2025; AiMultiple Research. Available online: <https://research.aimultiple.com/predictive-maintenance-tools/> (accessed on 14 July 2025).
21. STMicroelectronics. Pump Anomaly Detection Based on Vibrations. 2025. Available online: [https://www.st.com/content/st\\_com/en/st-edge-ai-suite/case-studies/pump-anomaly-detection-based-on-vibrations.html](https://www.st.com/content/st_com/en/st-edge-ai-suite/case-studies/pump-anomaly-detection-based-on-vibrations.html) (accessed on 22 May 2025).
22. NCD. Industrial IoT Wireless Predictive Maintenance Sensor V3. 2025 Available online: <https://store.ncd.io/product/iot-wireless-predictive-maintenance-sensor/> (accessed on 22 May 2025).
23. Lucas, M. Acoustic Emission: The Next Generation of Vibration Techniques. 2025. Available online: <https://www.reliableplant.com/Read/28771/acoustic-emission-techniques> (accessed on 22 May 2025).
24. Liu, J.; Shao, Y. Overview of dynamic modelling and analysis of rolling element bearings with localized and distributed faults. *Nonlinear Dyn.* **2018**, *93*, 1765–1798. [[CrossRef](#)]
25. Kibrete, F.; Woldemichael, D.E.; Gebremedhen, H.S. Multi-Sensor data fusion in intelligent fault diagnosis of rotating machines: A comprehensive review. *Meas. J. Int. Meas. Confed.* **2024**, *232*, 114658. [[CrossRef](#)]
26. Zhou, Z.; Abawajy, J.H. Reinforcement Learning-Based Edge Server Placement in the Intelligent Internet of Vehicles Environment. *IEEE Trans. Intell. Transp. Syst.* **2025**. [[CrossRef](#)]
27. Texas Instruments. BQ24095: Single-Input, 1A Li-Ion Charger with Dynamic Power Path Management, 2015. Datasheet Rev. K. Available online: <https://www.ti.com/product/BQ24095> (accessed on 20 April 2025).

28. Texas Instruments. LP38511-ADJ: 1.5A Fast-Response, Low-Dropout Linear Regulator, 2009. Datasheet Rev. F. Available online: <https://www.ti.com/product/LP38511-ADJ> (accessed on 20 April 2025).
29. Maxim Integrated. MAX291–MAX296: 8th-Order, Lowpass, Switched-Capacitor Filters, 2010. Datasheet Rev. 5. Available online: <https://www.analog.com/en/products/max291.html> (accessed on 20 April 2025).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.