University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Distributed Acoustic-Vibration System for Industrial Predictive Maintenance

Maximilian Huwyler
Zurich, Switzerland
Student ID: 13-925-557

Supervisor: Prof. Dr. Bruno Rodrigues, Katharina O. E. Müller,
Prof. Dr. Burkhard Stiller
Date of Submission: April 6, 2026

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

Pumped-storage hydropower plants play a critical role in the energy transition by providing flexible grid storage; yet, their turbines are subject to significant mechanical stress due to frequent switching between generation and pumping modes. Unplanned equipment failures in such facilities carry substantial economic consequences, motivating the need for automated fault detection and localization systems. While acoustic anomaly detection has been established as a viable approach for industrial condition monitoring, its application to hydropower environments is complicated by strong background noise and variable operating conditions. Furthermore, existing systems address only whether an anomaly has occurred without identifying where within the machine room it originates.

This thesis investigates whether combining acoustic and vibration sensing within a low cost distributed multimodal anomaly detection and localization pipeline improves upon audio-only approaches. Five wireless sensor nodes, each integrating a microphone and an inertial measurement unit, are deployed across a small-scale experimental setup simulating a pumped-storage hydropower machine room. The system relies on off the shelf hardware without precise time synchronization, keeping both cost and deployment complexity low. Three reconstruction-based autoencoder models are trained on normal operating data: an audio-only model, a vibration-only model, and a multimodal model combining both modalities through feature-level fusion. Anomalies are induced by dropping a ball at nine predefined positions across three intensity levels, covering a range of spatial and energetic anomaly conditions.

The results show that vibration-based detection substantially outperforms audio across all intensity levels, with the performance gap most pronounced at low intensity, where the audio model performs near chance level. Multimodal fusion provides marginal additional gains over vibration alone, most notably at low anomaly intensity. For localization, audio unexpectedly outperforms both vibration and multimodal models, a result attributed to the selective nature of audio detection and the inconsistent propagation of structural vibrations through the heterogeneous experimental setup. Overall, the results demonstrate that reconstruction error based localization is a feasible alternative to wired high precision time synchronization approaches, relying instead on low cost wireless hardware. However, localization accuracy in the current prototype remains modest and warrants further investigation.

Zusammenfassung

Pumpspeicherkraftwerke spielen eine entscheidende Rolle in der Energiewende, indem sie flexible Netzkapazität bereitstellen. Gleichzeitig sind ihre Turbinen aufgrund des häufigen Wechsels zwischen Erzeugungs- und Pumpbetrieb erheblichem mechanischem Stress ausgesetzt, und ungeplante Ausfälle haben erhebliche wirtschaftliche Folgen. Obwohl sich die akustische Anomalieerkennung als praktikabler Ansatz für die industrielle Zustandsüberwachung etabliert hat, wird ihre Anwendung in Wasserkraftumgebungen durch starke Hintergrundgeräusche und variable Betriebsbedingungen erschwert. Zudem adressieren bestehende Systeme nur, ob eine Anomalie aufgetreten ist, ohne deren Ursprungsort im Maschinenraum zu identifizieren.

Diese Arbeit untersucht, ob die Kombination von akustischer und Vibrationssensorik in einer kostengünstigen, verteilten multimodalen Pipeline gegenüber rein audiobasierten Ansätzen Vorteile bietet. Fünf Sensorknoten, die jeweils ein Mikrofon und eine Inertialmesseinheit integrieren, werden in einem kleinmaßstäblichen Versuchsaufbau eingesetzt, der einen Pumpspeicher-Maschinenraum simuliert. Das System verwendet handelsübliche Hardware ohne präzise Zeitsynchronisation. Drei rekonstruktionsbasierte Autoencoder-Modelle werden auf normalen Betriebsdaten trainiert: ein reines Audiomodell, ein reines Vibrationsmodell und ein multimodales Modell mit Fusion auf Repräsentationsebene. Anomalien werden durch das Fallenlassen eines Balls an neun Positionen auf drei Intensitätsstufen induziert.

Die Ergebnisse zeigen, dass vibrationsbasierte Erkennung die audiobasierte auf allen Intensitätsstufen deutlich übertrifft, wobei der Unterschied bei niedriger Intensität am ausgeprägtesten ist. Multimodale Fusion bietet geringfügige zusätzliche Verbesserungen, am deutlichsten bei niedriger Intensität. Bei der Lokalisierung übertrifft Audio überraschenderweise beide anderen Modelle, was auf die selektive Natur der Audioerkennung und die inkonsistente Vibrationsausbreitung im heterogenen Versuchsaufbau zurückgeführt wird. Insgesamt zeigen die Ergebnisse, dass rekonstruktionsfehlerbasierte Lokalisierung eine praktikable Alternative zu Systemen mit hochpräziser Zeitsynchronisation darstellt, wobei die Lokalisierungsgenauigkeit im aktuellen Prototyp bescheiden bleibt und weiterer Untersuchungsbedarf besteht.

Acknowledgments

I would like to thank the Communication Systems Group (CSG) at the Department of Informatics of the University of Zurich for enabling me to conduct this thesis project and for providing the necessary infrastructure and support that made this work possible.

I am particularly grateful to my supervisors, Prof. Dr. Bruno Rodrigues, Katharina O. E. Müller, and Prof. Dr. Burkhard Stiller, for their guidance, feedback, and continued support over the course of this work.

Finally, I would like to thank my family and friends, whose support and companionship outside of the university gave me the energy and balance needed to see this work through.

Contents

Chapter 1

Introduction

Section 1.1 motivates the thesis by establishing the importance of condition monitoring in pumped-storage hydropower plants and the limitations of existing approaches. Section 1.2 formulates the research questions, and Section 1.3 outlines the contributions of the work, followed by an overview of the thesis structure in Section 1.4.

1.1 Motivation

Hydropower is Switzerland's most important domestic source of renewable energy, accounting for approximately 59.5% of domestic electricity production across 704 active plants. Among these, pumped-storage facilities contribute approximately 4.1% of total hydropower production [1]. In the broader Alpine context, pumped-storage hydropower plays a central role in the energy transition by providing flexible, low-carbon storage capacity that complements the intermittent nature of wind and solar power [2]. Unlike conventional hydropower plants, pumped-storage facilities must continuously alternate between generating electricity during periods of high demand and pumping water back to the upper reservoir during periods of surplus. This repeated switching subjects turbines and generators to cyclic mechanical loading and dynamic stress, accelerating component fatigue and wear over time [3].

The consequences of equipment failures in such facilities can be severe. On 3 July 2009, a lightning induced emergency shutdown at the Rodundwerk II pumped-storage plant in Vorarlberg, Austria, caused a seven ton pole to detach from the rotor, destroying both the generator and the turbine [4]. The repair required the complete replacement of the machine set and the entire control system. Beyond the direct repair costs, the financial impact of such outages extends to lost production revenue. The Rodundwerk II plant operates at a full-load capacity of 295 MW, and at prevailing wholesale electricity prices, Khamaisi et al. estimated that each hour of unplanned downtime alone costs over 28,000 Euros [5].

Predictive maintenance offers a proactive alternative to reactive repair, enabling faults to be identified and addressed before they escalate into costly failures [6]. Looking at

the broader industrial picture, unplanned downtime costs the world's 500 largest companies approximately $1.4 trillion annually, equivalent to 11% of their combined revenues. Siemens estimates that broad adoption of AI-driven machine health monitoring across Fortune Global 500 industrial organizations could save 2.1 million hours of downtime annually, deliver $388 billion in productivity gains, and reduce maintenance costs by $233 billion through a 40% reduction [7].

Anomaly detection and localization are core components of predictive maintenance, enabling faults to be identified and located before they escalate into costly failures. Acoustic anomaly detection is a common approach for monitoring industrial machinery [8–11]. Khamaisi et al. [5] applied this approach directly to the Rodundwerk II pumped-storage plant, benchmarking several machine learning models on real-world recordings. Their work highlights the particular challenges of hydropower environments, where strong background noise and shifting operating modes between pumping and generation complicate reliable detection. Furthermore, their system focuses exclusively on detecting whether an anomaly has occurred, without addressing where within the machine room it originates.

Vibration sensing offers a physically distinct perspective on the same underlying events and has equally been applied to anomaly detection in industrial machinery [12–14]. This thesis investigates what vibration sensing can contribute to an anomaly detection and localization pipeline when combined with audio, exploring whether the addition of structural vibration data improves detection reliability, enables or enhances spatial localization of fault sources, or both. This question is investigated in the context of the machine room of a pumped storage hydropower plant, using a controlled small scale experimental setup designed to replicate its essential acoustic and mechanical characteristics. Figure 1.1 shows the machine room of the Rodundwerk II plant, illustrating the scale and complexity of the environment that motivates this work.

Figure 1.1: Machine Room Rodundwerk 2 [5]

1.2 Thesis Goals

Building on the work of Khamaisi et al. [5], who demonstrated the viability of acoustic anomaly detection in a pumped-storage hydropower setting, this thesis extends the approach in two directions: the addition of vibration sensing as a complementary modality and the introduction of spatial localization of anomaly sources. Three research questions guide the work.

The first question asks whether vibration sensing improves anomaly detection compared to audio alone. While audio based anomaly detection has been established as a viable approach, it faces well-documented challenges in noisy industrial environments. Vibration signals propagate through machinery structures independently of airborne noise, potentially offering a more robust detection signal under such conditions.

The second question asks whether combining both modalities within a unified multimodal autoencoder framework yields additional detection gains over either modality alone. Feature level fusion of audio and vibration has shown promise in comparable settings [15], but its benefits in a hydropower context remain unexplored.

The third question asks whether a low cost distributed sensor deployment can estimate the spatial origin of anomalies within a machine room. Achieving the precise time synchronization required by classical acoustic localization methods is not feasible with off the shelf wireless embedded hardware. This thesis therefore investigates whether the combined responses of spatially distributed sensor nodes provide sufficient information to localize fault sources without relying on specialized wired equipment.

1.3 Contributions

This thesis makes the following contributions. On the system side, a complete end to end distributed multimodal anomaly detection and localization pipeline is designed and implemented, spanning embedded sensor firmware, wireless data acquisition, preprocessing, model training, inference, and spatial localization. A controlled small scale experimental setup is constructed to simulate the essential characteristics of a pumped storage hydropower machine room, enabling reproducible evaluation of the system under varying anomaly positions and intensities.

On the scientific side, a comparative evaluation of audio only, vibration only, and multimodal autoencoder models is conducted across three anomaly intensity levels. The results provide evidence that, under the tested conditions, vibration achieves better detection results than audio, and that multimodal fusion provides marginal additional gains, most notably at low anomaly intensity. Reconstruction error based spatial localization is evaluated in a distributed multimodal sensor setting in a simulated machine room context, comparing four localization methods across modalities, drop positions, and intensity levels. The analysis reveals clear trade-offs between methods that perform well at sensor positions and those better suited to intermediate locations. The results furthermore provide evidence that reconstruction error based localization is a feasible low cost alternative

to classical time based localization in wireless sensor deployments, though localization accuracy in the current prototype remains modest and leaves room for improvement.

1.4 Thesis Structure

The theoretical and technical foundations of the system are established in Chapter 2, covering time-frequency representations of acoustic and vibration signals, autoencoder-based anomaly detection, source localization techniques, and the communication protocols used in the distributed deployment. A review of related work on acoustic and vibration-based anomaly detection, multimodal fusion, and reconstruction error based fault localization concludes the chapter. The system design is presented in Chapter 3, motivating the key architectural and algorithmic decisions with reference to the related work. Chapter 4 describes the implementation of selected pipeline components and provides a practical walkthrough of how the system is set up and operated end to end. The experimental setup, evaluation methodology, and detection and localization results are presented in Chapter 5, followed by a discussion and an analysis of the limitations of the work. Finally, Chapter 6 summarizes the findings, draws conclusions with respect to the research questions, and outlines directions for future work.

Chapter 2

Fundamentals

The theoretical foundations of the system are introduced in Section 2.1, covering time-frequency representations, autoencoder-based anomaly detection, and source localization. Section 2.2 describes the communication and synchronization protocols that form the infrastructure layer of the distributed deployment. Section 2.3 surveys existing work on acoustic and vibration-based anomaly detection, multi-modal fusion, and reconstruction-error-based fault localization and positions the present work relative to the state of the art.

2.1 Theoretical Background

This section covers the core concepts on which the system is built: time-frequency representations of acoustic and vibration signals, autoencoder-based anomaly detection, and source localization techniques.

Mel Spectrogram

A log-mel spectrogram (LMS) is a time-frequency representation of an audio signal that reflects how humans perceive sound. Its computation proceeds in three stages. First, the raw audio signal is divided into short overlapping frames and the Short-Time Fourier Transform (STFT) is applied to each frame to obtain its frequency content. Second, the resulting spectrum is filtered using a bank of triangular filters arranged according to the mel scale [16, 17]. The mel scale is a perceptual frequency scale that compresses high frequencies more than low frequencies, mimicking the fact that human hearing is more sensitive to pitch differences at lower frequencies. Third, the logarithm of the filter outputs is taken, yielding the log-mel spectrogram [17].

Compared to a raw linear frequency spectrogram, the mel filter bank reduces the number of frequency bins while focusing on the ranges where human hearing is most sensitive. Compared to Mel-Frequency Cepstral Coefficients (MFCCs), the log-mel spectrogram

skips the additional Discrete Cosine Transform (DCT) step. In MFCC computation, the DCT is applied to the log filter bank outputs to decorrelate the coefficients, as the filter bank energies are highly correlated with each other due to the overlapping triangular filters [17]. The log-mel spectrogram retains this correlation structure, making it well suited as input to convolutional neural networks, which can process it as a two-dimensional image and learn patterns across both time and frequency [18].

The log-mel spectrogram takes the form of a two-dimensional image in which the horizontal axis represents time, the vertical axis represents mel-scaled frequency, and the color intensity at each point encodes the log energy present in that frequency band at that moment. Sustained tonal components, such as a motor running at a constant speed, appear as horizontal bands at the corresponding frequency. Transient events, such as impacts or clicks, appear as brief vertical streaks spanning multiple frequency bands. Gradual changes in machine behavior, such as a shift in rotational speed, manifest as horizontal bands drifting up or down over time. Figure 2.1 illustrates two example log-mel spectrograms: one capturing steady background noise and the other showing the same signal with several superimposed impacts.

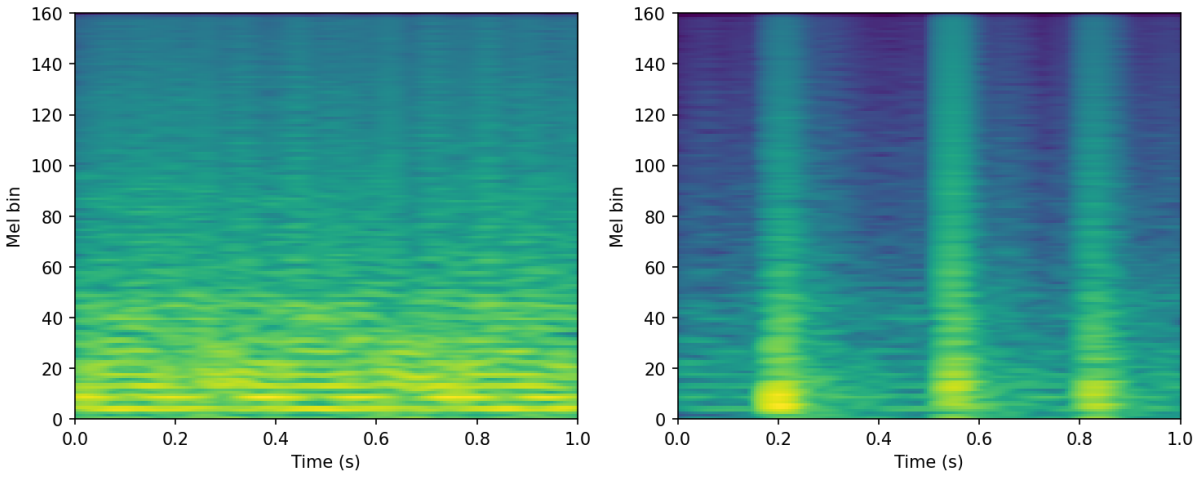

Figure 2.1: Log-Mel Spectrogram of a Steady Vibration and Several Impacts

Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) is a time-frequency analysis technique that decomposes a signal into a set of scaled and translated versions of a localized mother wavelet function [19, 20]. The CWT addresses a fundamental limitation of both the Fourier transform and the Short-Time Fourier Transform (STFT). The standard Fourier transform provides a global frequency description but discards all temporal localization, making it poorly suited for non-stationary signals whose frequency content changes over time. The STFT improves on this by applying a fixed-length window before transforming, yielding simultaneous time and frequency information. However, its time-frequency resolution is fixed for all frequencies. The CWT, instead, uses a variable window: at high frequencies, the window narrows, providing fine temporal resolution; at low frequencies,

the window widens, providing fine frequency resolution [19]. This frequency dependent resolution makes the CWT well suited for signals that contain both short transient events and slowly varying components, as is commonly the case in industrial machinery sounds.

The result of applying the CWT to a signal is a two-dimensional scalogram, in which the horizontal axis represents time, the vertical axis represents scale (inversely related to frequency), and the color intensity at each point encodes the magnitude of the wavelet coefficients [19]. High-magnitude regions indicate time intervals where the signal strongly resembles the wavelet at the corresponding scale, revealing where transient features and oscillatory components are concentrated in time. Figure 2.2 illustrates two example CWT scalograms: one capturing steady background vibration and one showing the same signal with several superimposed impacts.

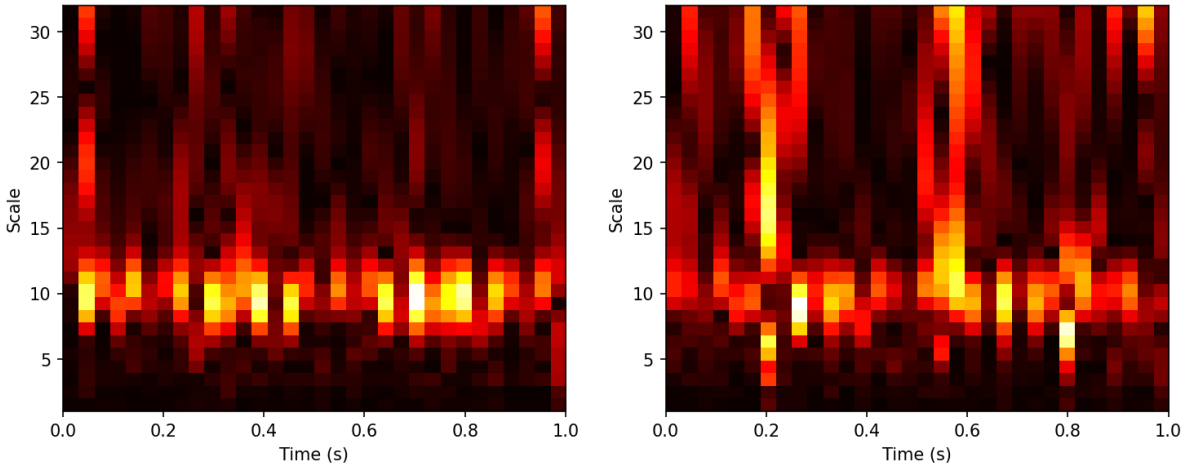

Figure 2.2: CWT of a Steady Vibration and Several Impacts

Autoencoders

An autoencoder (AE) is a neural network trained to reconstruct its input at the output layer, routing the data through a bottleneck of lower dimensionality in between [21, 22]. It consists of two components: an encoder, which compresses the input into a compact latent representation, and a decoder, which attempts to reconstruct the original input from that representation (See Figure 2.3). The bottleneck forces the network to discard redundant information and retain only the most essential structure in the data.

Training is performed in an unsupervised manner by minimizing the reconstruction error between the input and the output, commonly measured as the mean squared error (MSE) or mean absolute error (MAE). No labels are required, the model receives only examples of normal data and learns to capture the statistical regularities of that distribution [22].

Training is performed in an unsupervised manner by minimizing the reconstruction error between the input and the output, commonly measured as the mean squared error (MSE) or mean absolute error (MAE). No labels are required — the model receives only examples of normal data and learns to capture the statistical regularities of that distribution [22].

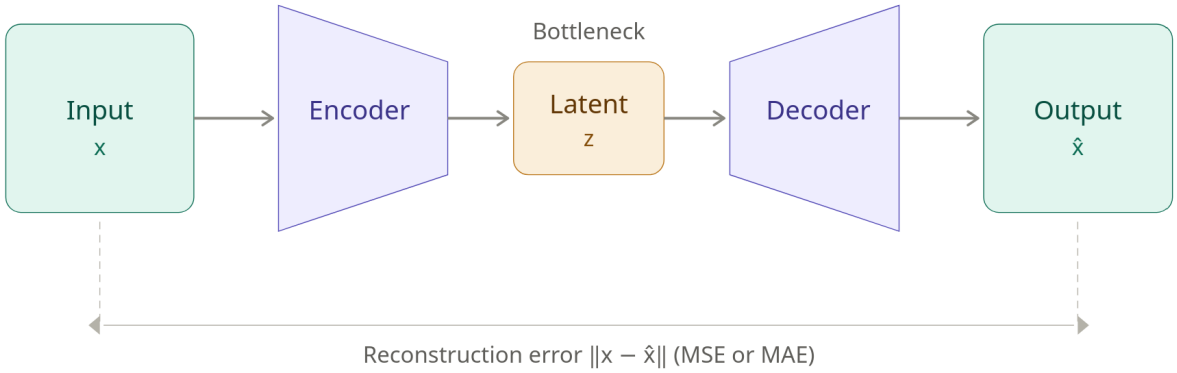

Figure 2.3: Autoencoder Architecture

Figure 2.4 illustrates this principle: the top row shows a normal window where the autoencoder produces a faithful reconstruction closely matching the input, while the bottom row shows an anomalous window where the reconstruction deviates noticeably, resulting in a higher reconstruction error.

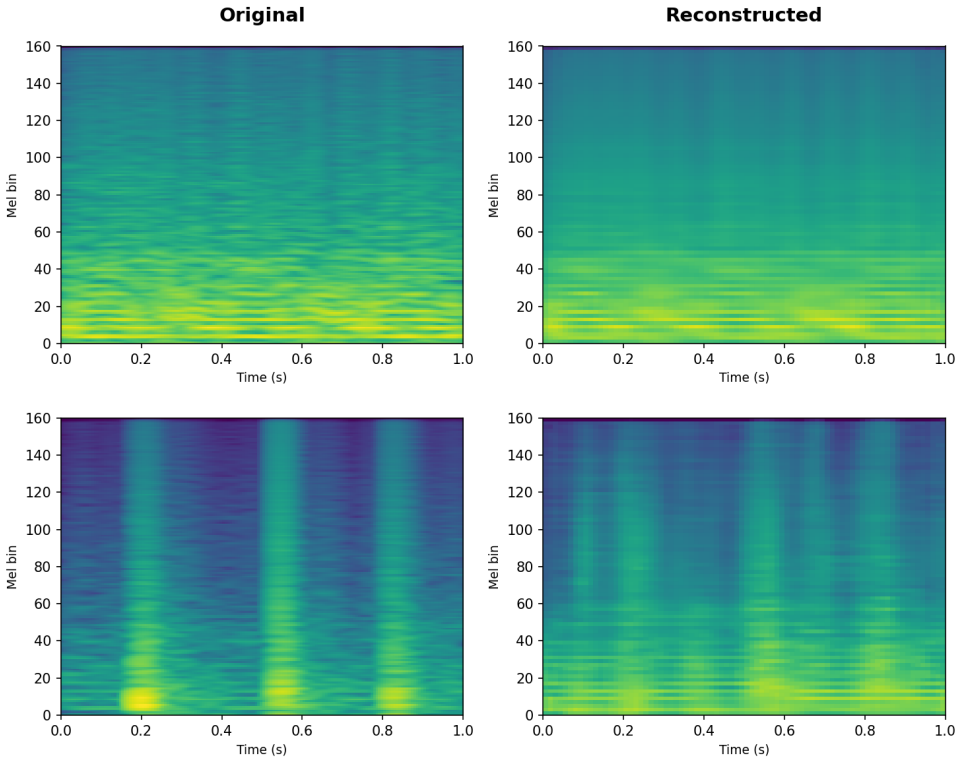

Figure 2.4: Log-mel Spectrogram Reconstruction for a Normal (top) and an Anomalous Window (Bottom)

When multiple sensor modalities are available, their complementary information can be combined through multi-modal fusion. In feature-level fusion, each modality is processed by a dedicated encoder branch that learns a modality-specific latent representation; the resulting representations are then merged before being passed to a shared decoder, allowing the model to capture cross-modal interactions in the latent space [15].

Localization

Time Difference of Arrival (TDOA) is a source localization technique that estimates the position of a signal source from the differences in arrival times of the emitted signal at spatially distributed receivers. A signal originating from a source reaches different receivers at different times, depending on their distances from the source. For each pair of receivers, the measured time difference constrains the source to a curved surface in space defined by the two receiver positions. The intersection of such surfaces from multiple pairs yields a position estimate. The time difference for each pair is estimated from the recorded signals using the generalized cross-correlation method, which identifies the time lag at which the similarity between the two channels is maximized [23]. Accurate TDOA estimation requires that all sensor nodes share a precise common time reference, as clock offsets between nodes translate directly into localization errors.

Both acoustic waves and structural vibrations lose energy as they propagate away from their source, a phenomenon known as attenuation. This decay occurs through three distinct mechanisms. First, geometric spreading causes the wave energy to be distributed over an increasingly large area as the wavefront expands, reducing the amplitude even in a perfectly lossless medium [24]. For a point source radiating in three dimensions, the amplitude decays proportionally to $1/r$, where r is the distance from the source. Second, material damping converts a portion of the mechanical wave energy into heat through internal friction within the medium, causing an additional exponential decay with distance [24, 25]. Third, scattering redistributes energy in unintended directions when the wave encounters inhomogeneities such as material boundaries, surface irregularities, or structural discontinuities [25].

Despite their differences, both airborne sound and structure-borne vibration share the fundamental property that signal amplitude decreases with distance from the source [24, 25]. As a practical consequence, a sensor positioned closer to an anomaly source will record a stronger signal than one positioned farther away, providing the physical basis for spatially localizing anomalies from the relative response magnitudes observed across a distributed sensor array.

Spatial localization can also be performed by interpreting the per-node reconstruction error as a proxy for the distance between the anomaly source and each sensor. Under the assumption that both acoustic and structural signals attenuate with distance, a node closer to the source will produce a higher reconstruction error than a more distant one, as the anomalous signal deviates more strongly from the learned normal representation [26, 27]. The estimated source position is then derived by aggregating the known sensor positions, weighted according to their respective reconstruction errors.

The centroid method computes the estimated source position as the weighted average of all sensor node coordinates, where the weight of each node is given by its reconstruction error. This approach is adapted from the weighted centroid localization principle introduced for wireless sensor networks [28], applied here to reconstruction error rather than received signal strength. A natural extension is the distance-weighted centroid, in which the reconstruction error of each node is additionally weighted by its proximity to the other nodes. Concretely, a node with high reconstruction error, supported by neighboring

nodes with similarly elevated errors, receives a higher weight, as the spatial agreement between nearby nodes provides stronger evidence that the source is located in that region. Conversely, an isolated node with high reconstruction error but no agreement from its neighbors receives a reduced weight, as its response may reflect indirect signal propagation rather than true proximity to the source. This modification aims to produce a more spatially robust estimate by anchoring the centroid towards regions where multiple nearby nodes agree on anomalous activity, following the principle of inverse distance weighting well established in spatial interpolation [29].

A simpler alternative is the maximum reconstruction error method, in which the estimated source position is taken directly as the coordinates of the node with the highest reconstruction error. This approach assumes that the closest node to the anomaly source will always produce the strongest response and forgoes any spatial aggregation in favor of a single discrete estimate. While computationally straightforward, this method is inherently limited to predicting one of the fixed sensor positions, making it incapable of estimating source locations between nodes.

2.2 Technical Background

This section describes the communication and synchronization protocols that form the infrastructure layer of the distributed sensor system, enabling reliable data transmission along the pipeline and ensuring that recordings across the network share a consistent time reference.

Network Time Protocol

Keeping clocks synchronized across networked devices is a fundamental requirement for distributed systems. The Network Time Protocol (NTP) [30] is the standard mechanism used to achieve this over IP networks. Under ideal conditions on fast local area networks, NTPv4 can achieve sub-millisecond accuracy, though this is not guaranteed in practice and depends heavily on network conditions and the hardware used [30].

NTP organizes time sources into a hierarchical structure called the stratum [30, 31]. At the top (stratum 0) sit high-precision reference clocks, such as GPS receivers or atomic clocks. Servers synchronized directly to these are stratum 1; servers synchronized to stratum 1 servers are stratum 2, and so on. Devices closer to the root of the hierarchy generally provide more accurate time [31]. A widely used public time service is the NTP Pool Project [32], a large virtual cluster of volunteer-operated servers that provides stratum 2 and stratum 3 time to clients worldwide.

To estimate the correct time, an NTP client exchanges timestamped packets with one or more servers. From the round-trip delay and the offset between the client and server clocks, NTP computes a best estimate of the true time and gradually steers the local clock toward it, avoiding abrupt jumps that could disrupt running processes [30].

Message Queuing Telemetry Transport

Message Queuing Telemetry Transport (MQTT) is a lightweight, publish-subscribe messaging protocol designed for constrained environments, prioritizing minimal bandwidth usage, low power consumption, and a small code footprint, making it well-suited for IoT and machine-to-machine communication [33].

The protocol operates on a broker-based architecture [33]. Rather than communicating directly with one another, clients connect to a central server called a `broker`, which is responsible for receiving and routing all messages. A client that wishes to send data `publishes` a message to a named `topic`. Any other client that has `subscribed` to that topic receives the message automatically. Publisher and subscriber are fully decoupled — neither needs to know of the other's existence (see Figure 2.5).

Topics are organized as hierarchical strings using forward-slash delimiters, such as `building/floor2/sensor3/temperature`, allowing subscribers to express interest in individual data points or entire subtrees. MQTT also defines three Quality of Service levels governing delivery guaranties: at most once, at least once, and exactly once, allowing a trade-off between reliability and overhead to be made depending on the application requirements [33].

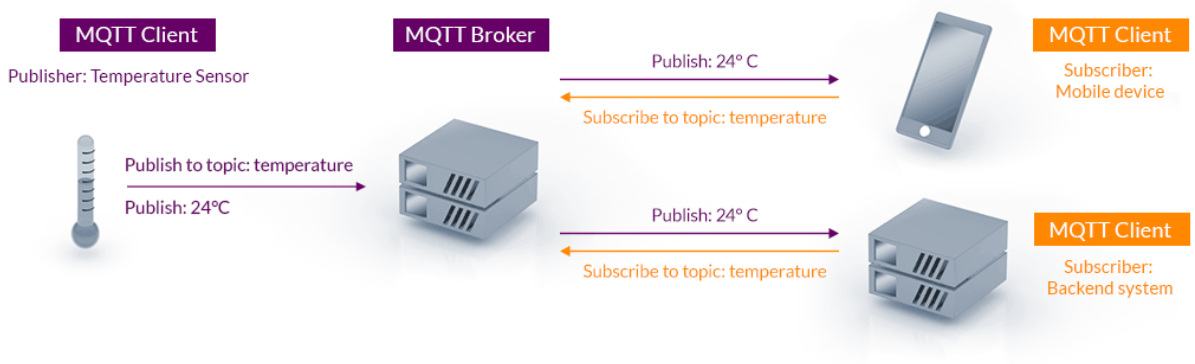

Figure 2.5: MQTT Architecture [34]

WebSockets

WebSocket is a communication protocol that provides a full-duplex channel over a single TCP connection. It addresses the limitations of HTTP for real-time, bidirectional communication. HTTP follows a strict request-response model, where the server can only send data in response to a client request [35]. This makes it poorly suited for applications requiring continuous data exchange, such as live monitoring or sensor streaming.

To establish a WebSocket connection, the client initiates an HTTP request with an `Upgrade` header, requesting a protocol switch. If the server supports WebSocket, it responds with HTTP status code 101 (Switching Protocols); after this, the HTTP protocol is no longer used, and the connection transitions to the WebSocket binary frame-based protocol. From this point, both client and server may send messages to each other at any time without waiting for a request over the same persistent connection. WebSocket is designed

to operate over the standard HTTP ports 80 and 443, making it compatible with existing infrastructure such as proxies and firewalls [35].

2.3 Related Work

This section reviews existing work on acoustic and vibration based anomaly detection, multimodal sensor fusion, and reconstruction error based fault localization, positioning the present work relative to the state of the art.

Anomaly Detection

Liu et al. [8] propose a self-supervised anomalous sound detection method motivated by a key limitation of the log-mel spectrogram: its mel filter bank suppresses high-frequency components where anomaly-relevant information may reside, leading to unstable detection performance across machines of the same type.

To address this, the method combines two complementary features extracted from the raw audio waveform. The first is a standard log-mel spectrogram. The second is a learned temporal feature (TF) extracted by a lightweight convolutional network designed to capture information that the mel spectrogram discards. The two features are concatenated and passed to MobileFaceNet (MFN) [36], a lightweight classifier trained in a self-supervised manner using machine identity labels.

The method is evaluated on the DCASE 2020 Challenge Task 2 benchmark, covering six types of industrial machines. Beyond the standard area under the receiver operating characteristic curve (ROC-AUC), the authors report the minimum area under the curve across individual machines of the same type, which measures detection consistency rather than average performance. The proposed method outperforms the previous state-of-the-art by 14.54% on this metric, and an ablation study confirms that the combination of both features is essential, neither alone achieves comparable stability [8].

Subsequent work by Zhang et al. [9] extended this approach by introducing a multi-head self-attention mechanism (MHSA) applied to the log-mel spectrogram, enabling the model to automatically identify and emphasize the most informative frequency components for a given machine type, further confirming that frequency-selective feature extraction is beneficial for anomalous sound detection.

A recurring critique of log-mel spectrograms as input features is raised by Kong et al. [10], who argue that mel filtering reduces sensitivity to high-frequency components and that windowing in the short-time Fourier transform limits low-frequency resolution, causing a loss of spectral detail relevant to anomaly detection. As an alternative, they use raw spectral features computed via the Discrete Fourier Transform (DFT), which retain finer frequency resolution across the full spectrum.

To exploit these features, the framework offers two network architectures trained using machine identity labels and ArcFace loss. ASDNet is a lightweight eight-layer one-dimensional convolutional network designed for edge deployment, while SpecMFN combines a spectral processing network with MobileFaceNet for higher-precision detection. Both architectures support multiple anomaly scoring strategies, with K-Means clustering yielding the best overall results. Evaluated on the DCASE 2020 Challenge Task 2 benchmark, ASDNet achieves an average area under the receiver operating characteristic curve of 94.42% and SpecMFN 94.36%, both substantially outperforming a log-mel baseline, supporting the argument that spectral features carry information that mel filtering discards [10].

Targeting the scalability limitations of per-machine binary classifiers, Peng et al. [11] propose a supervised multiclass framework for industrial acoustic anomaly detection that trains a single model to jointly identify machine type and operational state across eight classes on the MIMII dataset [37]. Rather than the log-mel spectrogram representations common in the acoustic anomaly detection literature, the authors introduce Smoothed Pseudo Wigner–Ville Distribution-based Mel-Frequency Cepstral Coefficients (SPWVD-MFCCs) as the input feature, arguing that the higher time-frequency resolution of the SPWVD better captures transient, non-stationary fault signatures under varying noise conditions compared to STFT-based decompositions. The extracted features are temporally aggregated via sliding window averaging before being passed to a long short-term memory (LSTM) convolutional neural network (CNN) classifier, which combines convolutional layers for local spectral feature extraction with LSTM layers for modeling temporal dependencies across the aggregated sequence.

Performance is evaluated using classification accuracy and macro F1-score under both matched and cross-noise conditions, where the model is trained on recordings with moderate background noise and tested on progressively noisier ones to simulate real deployment variability. The signal-to-noise ratio, which measures the ratio of the machine sound level to the background noise level, is varied across three levels in the MIMII dataset [37], with lower values corresponding to louder and more disruptive factory noise. The framework achieves strong classification performance under moderate noise and retains meaningful accuracy even under the loudest noise condition tested, demonstrating that SPWVD-MFCCs provide greater robustness to background interference than conventional STFT-based features [11].

Khamaisi et al. [5] present a comparative study of acoustic anomaly detection methods applied directly to a pumped-storage hydropower plant, specifically the Rodundwerk II facility in Vorarlberg, Austria (see Figure 1.1). This setting is of direct relevance to the present work, as hydropower machinery operates under acoustic conditions that differ substantially from the generic industrial benchmarks commonly used in the anomaly detection literature. The authors highlight that existing models trained on datasets such as MIMII [37] often fail to generalize to the complex acoustic environment of hydropower plants, which is characterized by strong background noise, signal reflections from concrete structures, and highly variable operating modes as turbines switch between generation and pumping cycles.

The preprocessing pipeline is carefully designed to handle these challenges. Raw audio is

recorded at 16 kHz and subjected to noise reduction filtering before feature extraction, followed by root mean square normalization to ensure consistent amplitude levels across recordings. A Short-Time Fourier Transform is then applied with a 1024-sample window and a hop length of 512, from which a 128-band log-mel spectrogram is derived and converted to decibels. The resulting spectrogram is min-max normalized to the unit interval before being segmented into overlapping frames with a 20% hop overlap, ensuring that short transient events are not lost at frame boundaries.

Three unsupervised models are benchmarked on two datasets collected on-site: one with synthetically induced anomalies produced by controlled hammer impacts, and one recorded under real operational conditions, containing a naturally occurring transient event during a turbine mode transition. The models compared are a long short-term memory autoencoder (LSTM-AE), a One-Class Support Vector Machine (OC-SVM), and K-Means clustering. While all three models achieve strong separation between normal and anomalous recordings with ROC AUC scores above 0.962, the F1-scores on the real-world dataset remain modest: the highest is 0.360, obtained by the LSTM autoencoder, while the One-Class SVM and K-Means score 0.261 and 0.314, respectively (see Table 2.1) [5].

Method	Train Time (s)	ROC AUC	Precision	Recall	F1-Score	Inference Time (s)
K-Means	5.25	0.962	0.192	0.861	0.314	0.043
OC-SVM	55.54	0.998	0.150	1.000	0.261	12.17
LSTM-AE	145.25	0.997	0.219	1.000	0.360	0.867

Table 2.1: Model Comparison Hydropower Storage Plant Setting [5]

Targeting fault detection in wind turbine gearboxes, Lee et al. [12] propose an unsupervised anomaly detection framework based on vibration signals, with a particular focus on the role of preprocessing in improving detection performance. Vibration data were collected from a wind farm in northern Sweden at a sampling rate of 12.8 kHz, covering 46 consecutive months of operation from a single wind turbine with two confirmed bearing failures.

The preprocessing pipeline constitutes the central contribution of the paper. Raw vibration signals are first decomposed using the Wavelet Packet Transform (WPT), which isolates fault-relevant characteristics across specific frequency sub-bands. A high-pass filter (HPF) is then applied to suppress low-frequency background noise and emphasize high-frequency components associated with mechanical faults such as bearing wear, followed by principal component analysis (PCA) for dimensionality reduction. Three progressive configurations are evaluated: PCA only, HPF with PCA, and the full WPT, HPF, and PCA pipeline. The same model architecture is used throughout, allowing the contribution of each preprocessing step to be assessed independently.

The model is a standard LSTM-AE trained exclusively on normal data, using the mean squared reconstruction error as the anomaly score and a threshold derived from the normal training distribution to flag anomalies at inference time. Performance is reported using ROC-AUC: the PCA-only baseline achieves 0.941, the HPF-PCA configuration drops to 0.862, suggesting that high-pass filtering without prior frequency decomposition can suppress useful signal content, while the full WPT-HPF-PCA pipeline achieves 0.974.

The authors conclude that targeted frequency-domain preprocessing, rather than model complexity, is the primary driver of detection performance [12].

Radicioni, Bono, and Cinquemani [13] investigate unsupervised anomaly detection in a degrading conveyor chain system using vibration signals from a piezoelectric accelerometer at 64 Hz. Rather than the standard STFT, the Fourier-based Synchrosqueezing Transform (FSST) is applied to produce sharper and sparser time-frequency spectrograms by reassigning spectral energy to the instantaneous frequency of each component, making dominant features more clearly visible. The resulting spectrograms are segmented into fixed-size image tensors and normalized with respect to motor velocity before being fed to the model.

Four autoencoder variants are compared: a Convolutional Autoencoder (CAE) and a Variational Autoencoder (VAE), each with either a dense or spatial latent space, all trained exclusively on healthy data and evaluated using reconstruction error as the anomaly score. The CAE with a spatial latent space clearly outperforms the other variants, achieving clean separation between healthy and degraded rounds with only 233k parameters compared to 42M for its dense counterpart. The VAE performs poorly under reconstruction-based metrics, which the authors attribute to a fundamental mismatch between reconstruction error and the probabilistic structure of the variational latent space [13].

Targeting long-term condition monitoring of on-load tap changers in real power infrastructure, Dabaghi-Zarandi et al. [14] deploy accelerometers on three single-phase autotransformers in the Hydro-Québec network, continuously recording vibration signals since 2016. The preprocessing pipeline extracts the signal envelope using the Hilbert-Huang Transform (HHT) and a low-pass filter (LPF), applies time realignment to remove temperature-induced temporal shifts between envelopes, normalizes each envelope to the unit interval, and reshapes it from a sample vector into a matrix to enable convolutional processing.

The model is a Convolutional Variational Autoencoder (CVAE) with a deliberately constrained two-dimensional latent space, trained separately for each transformer family. The low-dimensional latent space serves a dual purpose: it enables direct visualization of each envelope as a point on a 2D scatter plot, allowing engineers to track temporal drift and family boundaries, while also supporting anomaly detection through reconstruction error. The anomaly threshold is initialized as the maximum reconstruction error on the training set and subsequently optimized by sweeping percentages of this default value. When evaluated on held-out data from the same transformer family, the optimal threshold flags only 2-5% of samples as anomalous. When applied to data from the other two transformer families, detection rates reach 99-100%, demonstrating that the model learns a tight and family-specific representation of normal operation [14].

Wang et al. [38] propose HybridCube, an early fault detection framework for power transformers that fuses infrared thermal imaging and acoustic signals from spatially distributed sensors. Rather than operating on raw signals directly, each modality is first converted into a three-dimensional spatial representation: infrared images captured from three orthogonal views are projected into a volumetric temperature field using weighted view fusion, while acoustic signals from distributed sensors are interpolated into a continuous 3D pressure field. The two resulting spatial cubes are then fused at the data level into a unified Hybrid Cube using information entropy theory, specifically by maximizing mutual

information between the modalities while minimizing conditional entropy, ensuring the fused representation retains complementary information from both sources.

The Hybrid Cube is subsequently processed by a 3D-CNN encoder that extracts local features from spatial sub-cubes, aggregated via a sliding window into time-series feature vectors. A ConvLSTM-based decoder then predicts future infrared image sequences, and a Mask R-CNN module identifies anomalous regions across the three views, with results fused via a logic check to produce fault type, location, and estimated occurrence time. The framework was validated in a case study at the Three Gorges Power Plant, where it successfully identified early-stage faults that traditional single-modality methods missed [38].

Addressing the practical challenges of irregular signal acquisition and heterogeneous sensor fusion in industrial predictive maintenance, Zhang et al. [39] propose MentorPDM, a curriculum learning framework for multi-modal predictive maintenance. Operating on six modalities, namely vibration, force, phase current, speed, torque, and temperature, collected from bearing test rigs, each signal is first transformed into the frequency domain via the Fast Fourier Transform (FFT). A graph-augmented pretraining module then constructs a K-Nearest Neighbor (KNN) induced graph from time segments in the spectral domain and applies Graph Neural Network (GNN) message passing with a temporal contrastive learning (TCL) objective to capture dependencies within each modality. A bi-level curriculum learning module subsequently fuses the modality representations by dynamically assigning attention weights to both modalities and individual samples based on downstream task loss. Evaluated on the Paderborn bearing dataset [40] for fault classification and remaining useful life estimation, MentorPDM achieves 60.2% accuracy and 59.6% macro F1, consistently outperforming CNN, LSTM, and Transformer baselines, with performance improving further when using all modalities jointly compared to any single modality alone.

MultiSenseNet [41] is a supervised multi-modal architecture for binary machine failure risk classification. The model processes scalar sensor readings, including temperature, vibration frequency, power usage, and humidity, through dedicated per-modality CNN blocks, followed by a shared LSTM for temporal modeling, a multi-head transformer attention mechanism for feature importance weighting, and a GNN that encodes inter-sensor relationships based on machine type. All branches are fused via concatenation into fully connected layers with a sigmoid output. Evaluated on two small tabular datasets of roughly 1,000 samples each, MultiSenseNet achieves accuracies of 94.7% and 95.2%, respectively, outperforming standalone convolutional, recurrent, graph neural networks, and Transformer baselines.

Khan, Yüksel, and Kirchner [15] propose a multi-modal autoencoder-based anomaly detection system for minor vehicle damage detection, targeting use cases such as car sharing and fleet management. The system combines an IMU accelerometer and a microphone integrated into a compact device mounted on the vehicle's windshield. The accelerometer captures mechanical vibration responses to impact events along all three spatial axes, while the microphone provides complementary acoustic information.

Acceleration signals are first passed through a low-pass filter with a cutoff at 218 Hz before being transformed into 32×32 time-frequency representations using the CWT with a Morlet wavelet, with one scalogram computed per accelerometer axis. Audio signals

are processed through a bank of three band-pass filters with different frequency ranges before spectrogram computation, yielding a single-channel image of size 160×160. Both representations are fed directly into their respective encoder branches of the autoencoder architecture.

To integrate both modalities, the authors evaluate five feature-level mid-fusion architectures, each processing the two modalities through separate encoder branches before merging the resulting latent representations. The variants differ in how the fusion is performed: simple concatenation (MAA1), convolutional refinement of the concatenated representation (MAA2), additional max-pooling after convolution (MAA3), an attention mechanism applied to the joint representation (MAtten), and an attention bottleneck architecture (MBotF). This progression allows the authors to assess the trade-off between fusion complexity and detection performance.

The pooling-based variant MAA3 (see Figure 3.3) achieves the best overall performance, reaching a ROC-AUC of 0.92 on the audio modality, compared to 0.84 for the unimodal audio baseline and 0.80 for the unimodal acceleration baseline. Among the multi-modal variants, performance improves progressively from simple concatenation (MAA1, 0.74) through convolutional fusion (MAA2, 0.82) to pooling-based fusion (MAA3, 0.92), while the more complex attention-based and bottleneck variants (MAtten, MBotF, both 0.62) underperform compared to even the unimodal baselines. These results suggest that moderate fusion complexity, sufficient to capture cross-modal dependencies without over-compressing the joint representation, is key to effective multi-modal integration. Across all architectures, audio features consistently yield higher anomaly discrimination than acceleration features and benefit more from multi-modal fusion [15].

Localization

Holly et al. [26] propose an autoencoder-based end-to-end workflow for anomaly detection and fault localization in large industrial cooling systems, addressing the challenges of multivariate time series data with high dimensionality, inconsistent sensor recordings, and the absence of labeled training data.

The system operates on data from 34 sensors covering a diverse set of physical quantities, including temperature, vibration, and engine speed. Raw sensor signals are irregularly sampled and preprocessed through a pipeline that includes correlation-based signal selection, resampling to a fixed 60-second interval, forward-fill imputation for missing values, and min-max normalization. A sliding window of size 10 is applied to produce fixed-size input feature vectors, which are fed into an LSTM-AE trained exclusively on normal operating data. Anomaly detection is performed by thresholding the total reconstruction error across all sensor signals, where the threshold is derived from the mean and standard deviation of the training reconstruction errors and tuned using receiver operating characteristic curves.

Fault localization is achieved by decomposing the total reconstruction error into per-sensor individual reconstruction errors, directly inspired by saliency map techniques from computer vision. When the total reconstruction error exceeds the anomaly threshold, the

sensors contributing most to the error are identified by iteratively selecting the signal with the highest individual reconstruction error. These signals are then mapped to physical subsystems via an expert-defined look-up table, enabling root cause analysis without any supervised learning. The look-up table encodes domain knowledge about which sensors are typically associated with which failure types and was compiled by domain experts with extensive maintenance experience.

Evaluated on eight months of real-world cooling system data using four-fold cross validation, the workflow achieves an F1-score of 0.56 against automatically generated threshold-based labels. Notably, the autoencoder yields a substantially higher consistency score of 0.92 compared to 0.62 for the automatically generated labels, indicating that while the threshold-based labels detect anomalies earlier, the autoencoder recognizes them in a considerably more stable and persistent manner [26].

Li et al. [27] propose MIDAS, a mechanics-informed framework for unsupervised structural damage detection and localization using distributed strain sensors. The system is designed as a passive, deploy-and-forget solution that requires no labeled data and no prior knowledge of damage scenarios, learning exclusively from the structural response of the intact structure during normal operation.

Rather than operating on raw time-series strain data, the method first compresses the sensor signals into a compact statistical representation by fitting a Gaussian distribution to the cumulative exceedance times of the strain signal at several predefined thresholds. These compressed features serve as input to a fully-connected autoencoder trained to reconstruct the normal structural response. The reconstruction error on the training data establishes a reference distribution representing the intact state, and a shift in this distribution at inference time serves as the anomaly indicator. The central contribution is the augmentation of the standard MSE loss with a mechanics-informed term that penalizes pairwise reconstruction inconsistencies between sensors, weighted by their relative strain magnitudes in the undamaged state.

Localization is achieved by computing a per-sensor damage score from the relative change in reconstruction error norms between the undamaged reference and the test data. Because sensors positioned closer to a damage source experience a stronger deviation from the learned normal behavior, they accumulate higher damage scores than those farther away. These scores are spatially interpolated across the sensor array to produce a heatmap, with the peak identifying the estimated damage location. When only a small number of sensors are available, a weighted centroid of sensor positions is computed instead, where each sensor's position is weighted by its damage score. The Gaussian compression parameters are kept separate in this computation, which additionally enables damage type differentiation between crack-type damage and stiffness reductions such as loose connections. Validated on gusset plate and beam-column structures, the mechanics-informed autoencoder is compared against a standard autoencoder, an online principal component analysis baseline, Isolation Forest, One-Class Support Vector Machine, and a lightweight online anomaly detector, outperforming all baselines across accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve, with up to 35% better localization accuracy than the standard autoencoder for medium-sized cracks [27].

Discussion

The work of Liu et al. [8] and Zhang et al. [9] demonstrates that the log-mel spectrogram may discard high-frequency anomaly-relevant information. However, as the anomalies targeted in this work fall within the perceptually relevant frequency range captured by the mel filter bank [17], the log-mel spectrogram constitutes a sufficient input representation for our system. Furthermore, the consistent use of the area under the receiver operating characteristic curve across the anomalous sound detection literature [8, 9, 37] motivates its adoption as the primary evaluation metric in this work.

The empirical evidence provided by Kong et al. [10], showing that raw spectral features outperform log-mel features on the same benchmark, further underlines this trade-off. While their approach achieves strong detection performance, replacing the mel filter bank with full-spectrum DFT features introduces additional computational complexity that is not justified for the present work, where the characteristic fault signatures fall well within the frequency range already captured by the mel filter bank [17]

While Peng et al. [11] demonstrate clear noise robustness gains with SPWVD-MFCCs, the present work retains log-mel spectrograms given their established performance on acoustic anomaly detection benchmarks and lower computational cost, which is relevant in a multi-node deployment. Their recurrent architecture is likewise not adopted, as anomalies are treated as independent acoustic events that do not require modeling of temporal dependencies across windows. The cross-noise evaluation of Peng et al. [11] does, however, highlight an important practical consideration: industrial machines operate in acoustically complex environments, motivating the inclusion of simulated background noise in the experimental evaluation of the present system.

The findings of Khamaisi et al. [5] directly motivate several design decisions in the present work. Their preprocessing pipeline, particularly the use of normalization and overlapping frame segmentation, is adopted here as it addresses practical challenges equally relevant in the machine room setting considered in this thesis. Their work further confirms that log-mel spectrograms combined with a reconstruction-based autoencoder provide a solid foundation for acoustic anomaly detection in hydropower machinery, justifying the same core architectural choice made here. The evaluation methodology is equally adopted, as the combination of ROC AUC, precision, recall, and F1-score provides a comprehensive view of model performance that captures not only detection quality but also practical deployment feasibility. Given the high operational costs associated with real hydropower infrastructure, replicating such an environment for controlled experimentation is infeasible, which motivated the construction of a dedicated test environment for this work. Building on their acoustic-based approach, the present work extends toward a multi-modal and multi-node deployment, combining acoustic and vibration sensors distributed across the experimental setup to enhance both anomaly detection and spatial localization of fault sources.

While Lee et al. [12] demonstrate that targeted frequency-domain preprocessing is the primary driver of detection performance, their specific pipeline is not directly transferable to the present work. Their high-pass filtering targets fault signatures only accessible at their 12.8 kHz sampling rate, whereas the vibration sensors used here operate at 1 kHz. By

the Nyquist theorem, this limits the representable frequency range to 500 Hz, rendering high-frequency filtering redundant. Instead of the WPT, the present work employs the CWT, providing a richer and more continuous time-frequency decomposition without requiring prior identification of a fault-relevant sub-band. The successful application of a reconstruction-based autoencoder to vibration data supports its adoption here together with audio in a multimodal setting. The authors of this work based their reconstruction error threshold on the distribution of the normal data. This approach is also adopted using mean and standard deviation of the normal operating data.

The results of Radicioni, Bono, and Cinquemani [13] motivate several design decisions in the present work: the poor performance of VAE under reconstruction-based metrics justifies the exclusive use of a standard CAE, and their low vibration sampling rate motivates the use of higher-frequency sensors here to capture a broader range of fault-relevant frequency content without using FSST. Furthermore, the superiority of the spatial latent space over the dense variant reinforces confidence in the spatial latent space adopted in the autoencoder architecture of the present work.

The threshold selection strategy of Dabaghi-Zarandi et al. [14], which initializes the anomaly threshold as the maximum reconstruction error observed in the normal training data, presents a fully unsupervised and label-free approach to thresholding that is directly applicable to the present work, but it was abandoned because of a preference for a mean and standard deviation thresholding approach.

Wang et al. [38] demonstrate the viability of data-level fusion across modalities; however, constructing three-dimensional spatial representations for each modality introduces considerable complexity that is hard to justify when assessing the contribution of vibration signals because two-dimensional time-frequency representations are sufficient for this purpose. Although data-level fusion is an appealing direction, the present work opts for a simpler feature-level fusion approach. Nevertheless, their use of convolutional architectures for processing spatiotemporal sensor data reinforces that a CAE is a good choice for the present system.

The framework proposed by Zhang et al. [39] assumes irregular and asynchronous signal acquisition across modalities, a challenge that does not arise in the present work, where regular synchronous sampling is assumed. Furthermore, the considerable architectural complexity of MentorPDM, which combines graph construction, GNN pretraining, contrastive learning, and bi-level curriculum optimization, makes it difficult to justify for a system where the primary goal is unsupervised anomaly detection rather than supervised fault classification. For these reasons, the approach was not pursued further in the present work. Nevertheless, the consistent performance gains observed when fusing multiple sensor modalities confirm that combining different sensor types is a worthwhile direction, supporting the multi-sensor design adopted in this thesis.

The architecture and task formulation for MultiSenseNet [41] differ substantially from the present work, as the model operates on low-dimensional scalar readings under a supervised classification objective, neither of which applies to the hydropower setting considered here. Nonetheless, the sensor fusion ablation provides a clear empirical argument that combining complementary modalities consistently outperforms any single sensor, corroborating the

findings of Zhang et al. [39] and supporting the multi-modal deployment pursued in this work.

The architecture and fusion design of Khan, Yüksel, and Kirchner [15] are directly relevant to the present work. The system combines acoustic and vibration modalities processed as time-frequency representations within an unsupervised autoencoder trained on normal data and scored via reconstruction error. The use of CWT scalograms for vibration inspired the same choice in the present work. Comparing several fusion variants shows that the pooling-based mid-fusion architecture MAA3, which concatenates modality-specific latent representations before applying convolution and pooling, outperforms both simpler and more complex alternatives, suggesting that moderate fusion complexity is sufficient for effective cross-modal integration. The MAA3 architecture is therefore integrated into the anomaly detection and localization pipeline developed in this work and evaluated in a small-scale experimental setup simulating a pumped-storage hydropower machine room.

Table 2.2 provides a structured overview of all reviewed anomaly detection works, summarizing their domain, input modality, preprocessing, and model architecture.

Study	Year	Domain	Multi-Modal	Sound	Vibration	Data Preprocessing	Models
[8]	2022	Machinery	✗	✓	✗	TF, LMS	MFN
[9]	2023	Machinery	✗	✓	✗	TF, LMS	MFN + MHSA
[10]	2025	Machinery	✗	✓	✗	DFT	SpecMFN, ASDNet
[11]	2025	Machinery	✗	✓	✗	SPWVD-MFCCs	CNN-LSTM
[5]	2025	Hydropower	✗	✓	✗	STFT, LMS	K-Means, OC-SVM, LSTM-AE
[12]	2024	Wind Turbines	✗	✗	✓	WPT, PCA, HPF	LSTM-AE
[13]	2025	Machinery	✗	✗	✓	FSST	CAE, VAE
[14]	2025	Transformers	✗	✗	✓	HHT, LPF	CVAE
[38]	2024	Transformers	✓	✓	✗	IP	LSTM-AE, Mask R-CNN
[39]	2025	Machinery	✓	✗	✓	FFT	KNN, GNN, TCL
[41]	2025	Machinery	✓	✗	✓	-	CNN, LSTM, AM ,GNN
[15]	2025	Automobiles	✓	✓	✓	BPF, LPF, CWT	AE

Table 2.2: Overview of Related Work in Anomaly Detection

The work of Holly et al. [26] demonstrates that per-sensor reconstruction error can be decomposed to localize faults within a large industrial system without any supervised training, directly inspiring the spatial localization approach pursued in the present work. The same threshold derivation from the mean and standard deviation of training reconstruction errors is adopted in the present work and referred to as Gaussian-based thresholding. However, their system operates on multivariate time-series sensor readings rather than time-frequency representations, and the look-up table mapping sensors to subsystems relies on expert domain knowledge that is not available in our setting.

Building on a related principle, Li et al. [27] show that reconstruction errors from an autoencoder trained on normal data can serve not only as anomaly scores but also as the basis for fault localization by identifying which sensors contribute most to the total error. While MIDAS operates on static structures rather than rotating machinery, the core idea of spatially distributing reconstruction error across a sensor array to estimate fault location directly informs the localization strategy pursued in the present work.

The reviewed works collectively establish several building blocks for the system developed in this thesis. Reconstruction-based autoencoders have proven effective for unsupervised

anomaly detection across both acoustic and vibration modalities [5, 12, 13], and log-mel spectrograms and CWT scalograms have emerged as suitable time-frequency representations for each modality, respectively [5, 15]. Feature-level fusion of complementary sensor modalities consistently improves detection performance over single-modal baselines [15, 39, 41]. Reconstruction error has furthermore been shown to support fault localization by decomposing per-sensor contributions across a sensor array [26, 27], with the Gaussian-based thresholding strategy of Holly et al. [26] adopted directly in the present work. However, while acoustic and vibration modalities have been explored in isolation across various industrial settings, their combination within a unified autoencoder framework has not been applied to hydropower machinery. Furthermore, none of the reviewed works address the spatial localization of fault sources within a machine room using reconstruction error across a multi-node sensor deployment. The present work addresses both gaps by combining acoustic and vibration sensors in a hydropower setting, enabling simultaneous anomaly detection and fault localization through a shared reconstruction-based framework evaluated in a dedicated test environment.

Chapter 3

Design

This chapter presents the design of the multimodal anomaly detection and localization system, starting with an overview of the full pipeline in Section 3.1. The design of the sensor nodes is discussed in Section 3.2, followed by the server side data acquisition system in Section 3.3 and the preprocessing pipeline in Section 3.4. Section 3.5 covers the model design, including the autoencoder architecture, training setup, and thresholding strategy. The localization approach is presented in Section 3.6.

3.1 Pipeline Overview

The system follows an end-to-end pipeline spanning two stages: data acquisition and analysis, as shown in Figure 3.1. The pipeline takes raw sensor data from multiple distributed nodes as input and produces an anomaly localization estimate as output.

At the acquisition stage, each sensor node consists of a microcontroller connected to two sensing modalities: an audio sensor and a vibration sensor. During data acquisition, the microcontroller synchronizes its clock with an NTP server, ensuring that all nodes share a consistent timestamp base, which is essential for cross-device alignment during preprocessing and is required by the anomaly detection and localization module. The audio sensor streams data to a web server that aggregates incoming streams from all nodes in the network. The vibration sensor data is published to a message broker following a publish-subscribe pattern, likewise collecting data from all deployed nodes. Dedicated subscribers consume the incoming streams and persist the data to a shared file system, which acts as the handoff point between the acquisition and analysis stages.

At the analysis stage, a preprocessing module reads the raw recordings from the file system and transforms them into fixed-size time-frequency representations suitable for neural network input. Audio recordings are converted to log-mel spectrograms and vibration recordings to continuous wavelet transform scalograms. A data loader then feeds the preprocessed features into the autoencoder, handling batching, normalization, and shuffling during both training and inference. The autoencoder processes the input features and produces a reconstruction for each window, from which an anomaly detection

module computes a reconstruction error score and compares it against a given threshold derived from the normal training distribution. When an anomaly is detected, the per-node reconstruction errors across all sensor nodes are passed to the localization module, which estimates the spatial position of the anomaly source by exploiting the fact that nodes closer to the source produce stronger reconstruction errors, as the anomalous signal deviates more strongly from the learned normal representation at shorter distances.

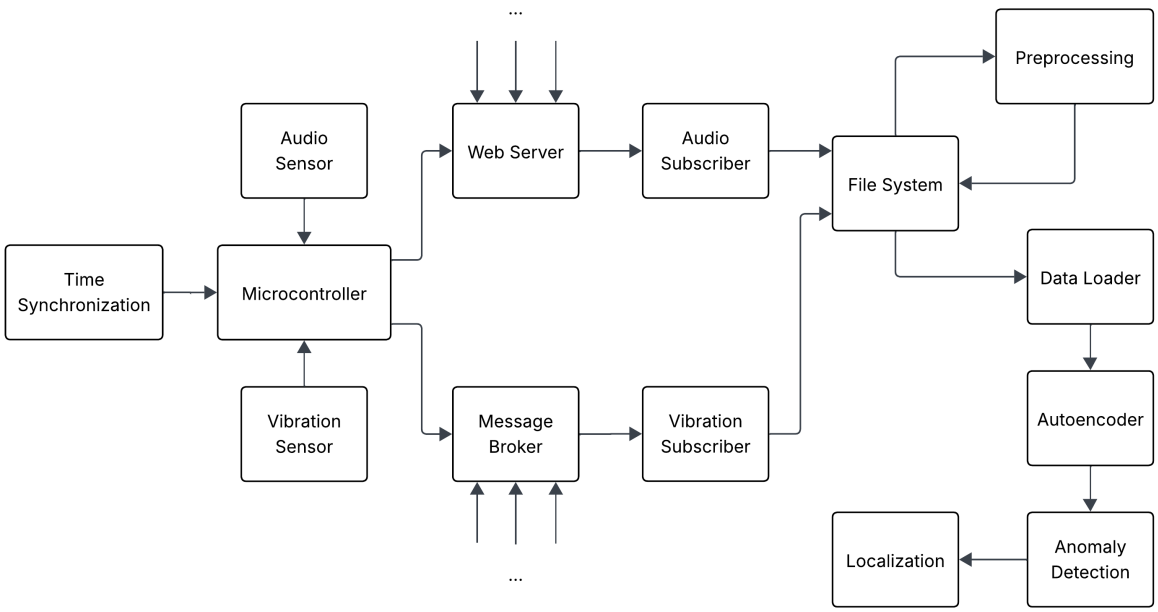

Figure 3.1: Pipeline Overview

3.2 Sensor Nodes

The choice to equip each sensor node with two complementary sensing modalities is motivated by the breadth of the related work. Acoustic anomaly detection has established itself as a primary approach in the literature, with a substantial body of work demonstrating that audio signals carry rich information about deviations from normal operating conditions [5, 8–11, 38]. Vibration signals have also been shown to support reliable anomaly detection, either as a standalone modality or in combination with operational parameters [12–14, 39, 41]. Several works have further demonstrated that combining multiple modalities can yield additional detection gains over either modality alone, as different sensor types capture complementary aspects of the same underlying physical event [15, 38, 39, 41]. The work of Khan, Yüksel, and Kirchner [15] is of particular relevance, as it combines audio and vibration modalities within a reconstruction-based autoencoder framework, the same class of model used across much of the related work [5, 12–15, 38], and demonstrates that their fusion through a feature-level multimodal architecture yields promising results in a comparable sensor setting. This directly motivated the adoption of the same dual-modality approach in the present work.

Pairing both sensors on a single microcontroller ensures that the two modalities are always spatially coincident and observe the same physical location. This matters for the localization approach, which relies on comparing reconstruction errors across nodes. If audio and vibration sensors were deployed as separate independent nodes, their positions would need to be tracked individually, increasing deployment complexity. Co-location further ensures temporal synchronization between the two modalities at the hardware level, as both sensors are read and timestamped by the same microcontroller, eliminating any inter-modal clock offset without requiring additional alignment mechanisms.

Since each sensor node operates as an independent embedded system with its own local clock, cross-device alignment of sensor windows requires a shared time reference. The localization approach compares reconstruction errors across all nodes for the same time window, which is only meaningful if all nodes are observing the same physical moment. Each node therefore synchronizes its clock to a shared time server using NTP [30], ensuring that all nodes maintain a consistent timestamp base throughout a recording session.

Hardware

Deploying sensor nodes as wireless embedded systems rather than wired sensors connected to a central acquisition unit significantly reduces hardware complexity. Wiring multiple sensors across even a small room requires expertise in electrical installation that goes beyond the scope of this work. Even configuring the sensors and connecting them to the microcontrollers via short cables required considerable effort. A wireless approach avoids this entirely, with each node being independent and requiring only a power connection. It also provides flexibility in sensor placement, as nodes can be repositioned without any rewiring, which is particularly valuable during the iterative development of the experimental setup.

The hardware design of each sensor node is driven by the goal of building a low cost prototype that can be assembled and modified without specialist expertise. A microcontroller was chosen as the central compute and transmission unit over more powerful single board computers, as it is sufficient for sensor reading and wireless transmission and is substantially cheaper. Each node integrates two sensors: a digital omnidirectional microphone for audio capture and an inertial measurement unit (IMU) for vibration capture. The microphone was selected for its low cost and omnidirectional pickup pattern, which ensures that acoustic events are captured regardless of their direction relative to the sensor. For vibration sensing, a piezoelectric sensor was initially considered, but an IMU was chosen instead, as it provides both accelerometer and gyroscope data at a comparable cost, yielding more information about the physical state of the node for the same price.

Both sensors are connected to the microcontroller via extension boards with screw terminal connectors rather than through soldering. This keeps the assembly modular and accessible, as cables can be exchanged or reconfigured without any specialist tools or permanent modifications, while the screw terminals provide sufficient mechanical robustness for the experimental setting. The resulting hardware design is low cost, easy to assemble, and flexible enough to support iterative changes during the development of the prototype.Furthermore, it enables the investigation of whether a low cost distributed sensor

deployment with limited time synchronization precision is capable of localizing anomaly sources within an experimental setup representing a hydropower machine room.

Firmware

The firmware design is centered around the challenge of acquiring two sensor streams simultaneously at different data rates without either stream interfering with the other. In an initial design, both sensors were read sequentially on a single core, alternating between inertial measurement unit and microphone reads within the same loop. This approach proved insufficient, as the sequential execution limited the IMU sampling rate to approximately 100 samples per second. To reach the target sampling rate of 1 kHz, the dual core architecture of the microcontroller was exploited by assigning the two modalities to separate cores. The inertial measurement unit runs on one core as a dedicated sampling task, ensuring a consistent sampling rate. Audio acquisition and all network communication run on the other core. This separation enables both modalities to be sampled at their target rates.

Data is passed between the two cores through a shared queue that acts as a buffer, ensuring that all data transmission and network handling is concentrated on a single core. The two modalities are transmitted over separate protocols chosen to match their respective data characteristics. Vibration data is published in discrete batches over a lightweight message broker protocol. Audio is transmitted as a continuous stream over a persistent connection protocol, reflecting its higher and more consistent throughput requirements. Both streams are timestamped at the moment of acquisition using the NTP synchronized clock of the microcontroller, allowing the server to align the two streams after reception without any additional synchronization mechanism.

3.3 Data Acquisition

In the server side data acquisition system, each component runs in a separate container orchestrated via a container orchestration tool, allowing the full stack to be started with a single command and eliminating local installation dependencies. The current prototype supports five sensor nodes. The architecture is designed with scalability in mind, and the codebase can be extended to support additional nodes with minimal changes; however, scaling to larger node counts would eventually be constrained by the hardware resources available to the host machine. Distributing the subscriber components across multiple machines and merging the data after preprocessing would be a natural extension to address this limitation, but was not explored in the present work.

Vibration data from all sensor nodes is routed through a message broker following a publish-subscribe pattern. Each sensor node publishes its IMU batches to a device specific topic, and a dedicated vibration subscriber consumes all incoming messages. The vibration subscriber listens on all device specific topics and processes incoming messages as they arrive. Each message contains a batch of raw IMU samples, which are unpacked and

converted to physical units before being written to disk. The subscriber maintains a per device write buffer that is flushed to disk in batches, reducing the number of individual write operations. Each recording session produces one CSV file per device, containing the accelerometer and gyroscope readings alongside their timestamps and transmission latency.

Audio data follows a different path. All sensor nodes establish persistent connections to a WebSocket proxy, which accepts connections on a single well-known port and forwards them internally to the audio subscriber. This keeps the network interface simple from the perspective of the sensor nodes, which need not be aware of the internal service structure.

The audio subscriber required a more sophisticated design than the vibration subscriber due to the substantially higher throughput of the audio stream. An initial implementation using synchronous blocking disk writes, similar to the approach used for vibration data, resulted in exponentially growing latency as the event loop blocked on each write operation, eventually causing the subscriber to fall behind the incoming stream. To address this, the audio subscriber was redesigned around two decoupled components, as shown in Figure 3.2. The main thread maintains a per device ring buffer that absorbs incoming audio frames without blocking. A dedicated file writer thread consumes write operations from a thread safe queue, ensuring that disk IO never blocks the subscription event loop. Flushing is triggered periodically, either after a fixed number of incoming messages or after a fixed time interval, whichever occurs first. A final flush is performed on disconnect or shutdown to ensure no data is lost. Each recording session produces two output files per device: a raw PCM audio file and a CSV timestamp log recording the timing metadata for every incoming frame.

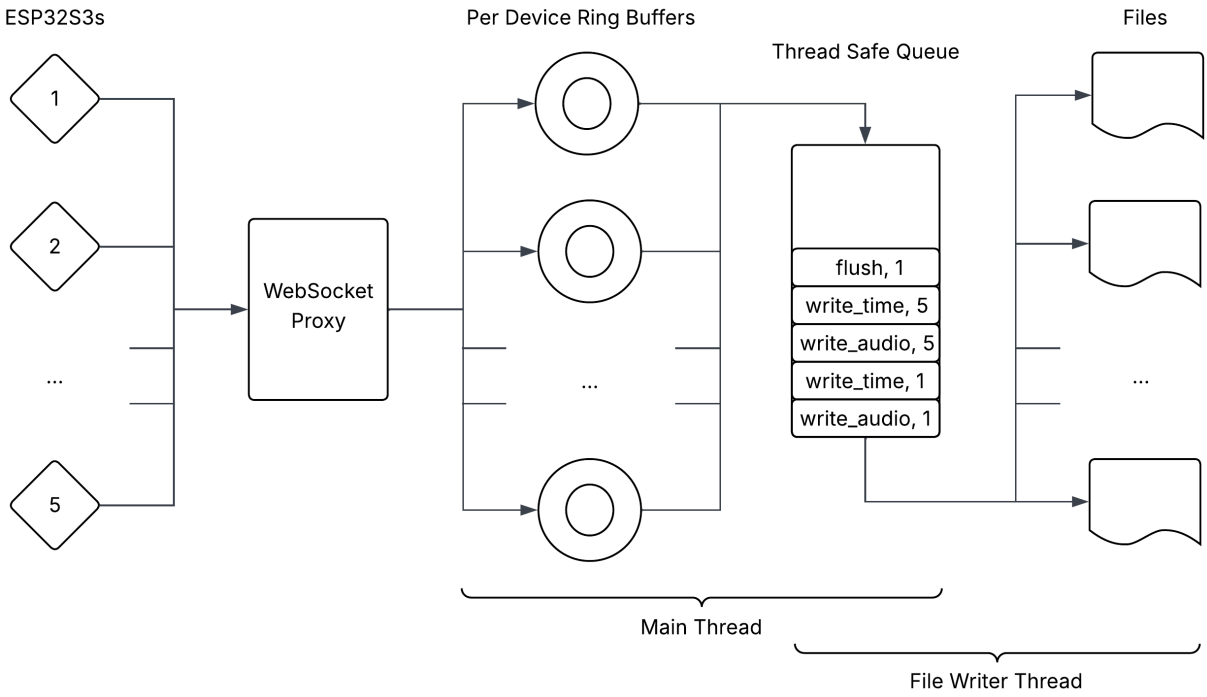

Figure 3.2: Audio Acquisition

The file system acts as the handoff point between the acquisition and analysis stages.

Both the vibration and audio subscribers write their output to a shared directory, from which the preprocessing module reads to prepare the data for both training and inference.

3.4 Preprocessing

Raw recordings from all sensor nodes are processed jointly, using the local timestamps as a common reference for cross-modal and cross-device alignment. A one second window size is chosen as it provides a sufficient duration to capture anomalous events such as impacts and their subsequent decay. A 75% overlap between consecutive windows is chosen for two reasons. First, it multiplies the number of available training samples by a factor of four compared to non-overlapping windows. Second, it ensures that transient events are captured in multiple windows regardless of where they fall within a window, and produces smoother reconstruction error trajectories over time compared to non-overlapping or low-overlap strategies. Windows for which any device or modality fails to meet a minimum data coverage threshold are discarded, ensuring that only complete and reliable windows are used for training and inference. This straightforward approach was chosen to avoid the complexity of handling irregular data acquisition. Furthermore, the aggressive overlap between consecutive windows means that even if two windows are discarded, the remaining windows still overlap with each other, preserving temporal continuity across the dataset. Per device normalization statistics are computed across all valid windows of each recording session and stored alongside the preprocessed data, accounting for differences in baseline signal levels across devices due to variations in sensor sensitivity and mounting conditions, removing the need for prior calibration of individual sensors.

Audio Preprocessing

Audio is recorded at 44.1 kHz on the sensor nodes. This sampling rate was not explicitly optimized during the initial development, as the relevant frequency ranges for the target application were not yet established. Subsequent analysis showed that the characteristic frequency content of both machine hum and impact events is well captured within the 0 to 8 kHz range, which, by the Nyquist theorem, is fully representable at a sampling rate of 16 kHz. The audio is therefore downsampled during preprocessing rather than modifying the sensor node firmware. The log-mel spectrogram is chosen as the audio feature representation, following the established practice in the acoustic anomaly detection literature [5, 8, 9]. The output shape of 160×160 is adopted from Khan, Yüksel, and Kirchner [15] to match their autoencoder architecture. The preprocessing parameters are selected to achieve the target 160×160 output shape. An FFT window of 2048 samples at 16 kHz, corresponding to approximately 128 ms of temporal context per frame, is a conventional choice widely found in the audio processing literature. A hop length of 100 samples is required to produce exactly 160 time frames per one second window, which results in approximately 95% overlap between consecutive frames. This aggressive overlap ensures that short transient events such as impacts appear prominently across multiple frames rather than being missed between them. Setting the number of mel bands to 160 matches the frequency dimension to the time dimension, yielding the target output.

Vibration Preprocessing

The CWT with a Morlet wavelet is chosen as the vibration feature representation, directly following Khan, Yüksel, and Kirchner [15], whose multimodal autoencoder architecture is used in this work. The IMU provides both accelerometer and gyroscope data. During development, both are retained in the preprocessing pipeline as it is initially unclear which proves more informative. The accelerometer is ultimately selected, consistent with the work of the aforementioned authors.

Each one second vibration window contains approximately 1000 samples per axis at a sampling rate of 1 kHz. The CWT is computed on the full window using 32 scales and a Morlet wavelet. Computing on the full window rather than on shorter segments reduces edge effects at the signal boundaries. The resulting time-frequency representation is then downsampled to 32 time bins per scale row, yielding a 32×32 scalogram per axis. The three axes are stacked along the channel dimension, producing the final output that matches the acceleration input dimensions of Khan, Yüksel, and Kirchner [15].

3.5 Anomaly Detection

Anomaly detection in this work is framed as an unsupervised learning problem. Collecting sufficient labeled anomaly data is time consuming and prone to annotation errors, and the required amount of labeled data is difficult to estimate in advance. As the dataset grows, manual labeling scales poorly, and automating the annotation process introduces its own complexity. Normal operating data, by contrast, requires no annotation and is readily available in a controlled experimental setup. Autoencoders using reconstruction errors as anomaly scores have been shown to be effective for unsupervised anomaly detection in related work [5, 12–14, 38]. Beyond binary detection, spatial differences in reconstruction error across distributed sensors can be exploited for localization [26, 27]. The autoencoder architecture used in this work is adopted from Khan, Yüksel, and Kirchner [15], which proposed a multi-modal convolutional autoencoder for small vehicle damage detection and demonstrated convincing results in a feature-level modality fusion setting. The architecture consists of separate convolutional encoders for each input modality, a shared latent bottleneck that fuses the encoded representations, and separate decoders that reconstruct each modality independently.

Model Architecture

The architecture consists of separate convolutional encoders for each modality, a shared latent bottleneck that fuses the encoded representations through feature level concatenation, and separate decoders that reconstruct each modality independently. The audio encoder takes a log-mel spectrogram of shape $160 \times 160 \times 1$ as input, while the vibration encoder takes a CWT scalogram of shape $32 \times 32 \times 3$. Both encoders repeatedly apply Conv2D with ReLU activation followed by MaxPooling, progressively compressing the spatial resolution until both produce a 10×10 spatial representation. The two

encoded representations are concatenated and passed through additional Conv2D and MaxPooling layers to produce a shared latent bottleneck of shape $5 \times 5 \times 256$. Two independent decoders reconstruct the audio and vibration outputs to their original input shapes using Conv2D and MaxPooling operations. The pooling-based variant MAA3 is selected as it achieves the best overall detection performance among the fusion variants evaluated in the original work. Max-pooling ensures that only the most relevant features are retained while reducing computational complexity, making it especially valuable for high-dimensional sensor data [15]. Figure 3.3 illustrates the full architecture.

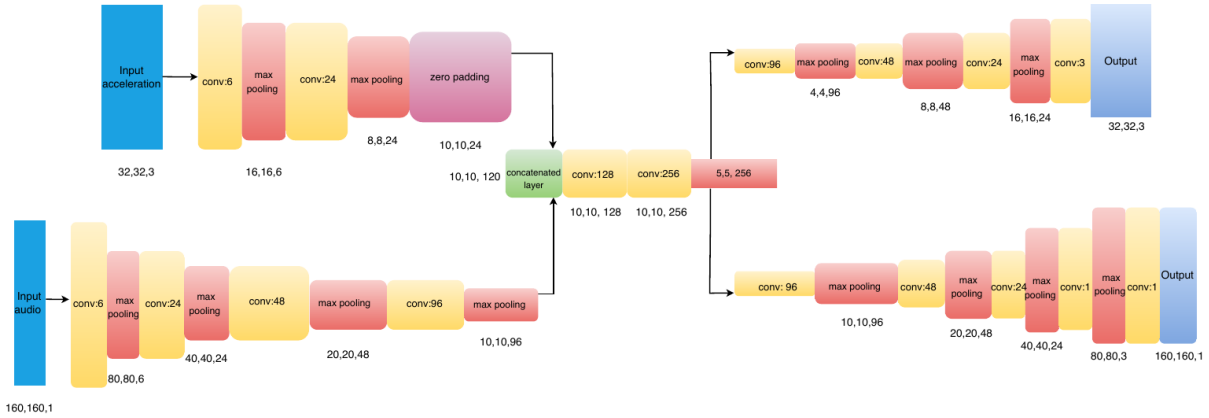

Figure 3.3: MAA3 Model Architecture [15]

In addition to the multimodal architecture, two unimodal variants are implemented: an audio only and a vibration only autoencoder. Both follow the same encoder-decoder design as their respective branches in the multimodal model, and the latent bottleneck structure remains unchanged. All three models are trained under identical conditions on the same normal operating data. This enables a direct comparison of anomaly detection performance across modalities, addressing whether vibration sensing improves detection compared to audio alone, and whether multimodal fusion yields additional gains over either modality individually. Since all three models also produce reconstruction errors that are used as the basis for localization, the same comparison can be extended to localization performance, allowing the contribution of each modality to be assessed across both tasks.

Loss Function and Training

The model is trained to minimize a weighted sum of mean squared error (MSE) reconstruction losses across both modalities:

$$\mathcal{L} = w_a \cdot \text{MSE}(\hat{x}_a, x_a) + w_v \cdot \text{MSE}(\hat{x}_v, x_v)$$

where x_a and x_v denote the audio and vibration inputs, and $\hat{x}_a$, $\hat{x}_v$ their reconstructions. Both weights are set to 1.0 by default. Khan, Yüksel, and Kirchner [15] evaluated several loss functions on the same architecture, finding that Log-Cosh slightly outperformed MSE (ROC-AUC 0.778 vs. 0.739) due to its robustness to outliers. Nevertheless, MSE was retained in this work for its simplicity and straightforward interpretability. The models

are optimized using Adam, and the best model checkpoint based on validation loss is saved throughout training. The dataset is split 80/20 into training and validation sets, and the model can be trained jointly on multiple recording sessions

Data Loader

The data loader is designed to efficiently serve training and inference across potentially large datasets spanning multiple recording sessions. At startup, it scans all preprocessed NPZ files and builds a flat index of every valid window-device pair across all sessions, recording which file, which window, and which device each entry belongs to. This separates the question of what data exists from the question of how to load it, keeping the indexing step lightweight regardless of dataset size.

Two loading modes are provided, motivated by the competing constraints encountered during development. Memory was initially the primary concern, leading to a streaming mode in which arrays are read from disk per batch, grouped by source file to minimize redundant file openings, with normalization applied on the fly. As training scaled up, speed became the bottleneck, motivating a preload mode in which all arrays are loaded into RAM at startup, stacked into two large contiguous matrices for audio and acceleration, respectively, and normalized once. Batch access then reduces to a single numpy index operation, eliminating all file IO during training. The preload mode is preferred when memory allows, and the streaming mode serves as a fallback for memory constrained environments.

Normalization is applied per device and per modality, using statistics computed during preprocessing and stored alongside the preprocessed data. If these statistics are not available, the data loader falls back to computing them on the fly by sampling 10% of the data using Welford's algorithm [42]. The resulting statistics are saved alongside the trained model and reused at inference time, ensuring that the same normalization is applied consistently across training and inference.

Thresholding

After training, the autoencoder is run on all normal training windows to collect the reconstruction error distribution for each device. A Gaussian is fitted to this distribution by computing the mean and standard deviation of the reconstruction errors, and the anomaly threshold is set at a fixed number of standard deviations above the mean, following the approach of Holly et al. [26]. At inference time, any window whose reconstruction error exceeds this threshold is flagged as an anomaly. This approach requires only normal operating data for both training and threshold calibration.

Although input normalization ensures that the features fed to the model share the same scale across devices, the resulting reconstruction error distributions still differ between devices. Each device occupies a different physical position and is mounted on a different surface, exposing it to a different acoustic and vibration environment. As a result,

the autoencoder may reconstruct some devices more accurately than others, producing systematically lower or higher reconstruction errors depending on the node. Applying a single global threshold would therefore bias detection towards or against certain devices. Computing the threshold independently for each device ensures that the anomaly criterion is calibrated to each node's individual reconstruction error distribution under normal conditions.

At inference time, each window is first normalized using the per device statistics computed during training and saved alongside the model, ensuring that the same normalization is applied consistently across training and inference. The normalized window is then passed through the autoencoder, and the MSE between the model output and the original input is computed separately for each modality. The two modality errors are then summed to produce a single scalar per window per device.

3.6 Localization

Time Difference of Arrival is the natural approach for acoustic source localization and is described in Section 2.1. However, TDOA requires very precise time synchronization between nodes, as the speed of sound means that even small timing errors translate directly into large localization errors. Achieving the required sub-millisecond precision would demand dedicated hardware time synchronization over wired connections, which is not feasible in this prototype given the hardware constraints and the expertise required. NTP over a local network can provide synchronization in the low millisecond range under ideal conditions [31]. Since the current system works with wireless connections and ideal conditions cannot be assumed, TDOA based localization is not reliable. Wiring the nodes together to achieve better synchronization was ruled out for the same reasons discussed in the sensor node design.

Reconstruction error based localization, by contrast, only requires that all nodes observe approximately the same time window rather than precise arrival time differences. The window level alignment provided by NTP synchronized timestamps is sufficient for this purpose, making it a practical and accessible alternative to TDOA in a low cost wireless deployment. This approach is motivated by Holly et al. [26] and Li et al. [27], who demonstrated that per-node reconstruction errors can serve as a spatial proxy for fault localization without requiring precise timing.

Localization requires the physical positions of all sensor nodes to be known, reconstruction errors per device to be available for the same time window across all nodes, and a shared time reference to ensure all devices observe the same physical moment. Before the reconstruction errors are used to compute the spatial estimate, they are normalized per device using the mean and standard deviation of the reconstruction errors observed during training. This ensures that a high error at a given device reflects a genuine deviation from its normal behavior rather than a generally elevated baseline, making the errors comparable across devices and allowing the spatial weighting to reflect proximity to the anomaly source rather than differences in absolute error scale.

Four localization methods are evaluated and compared, as described in Section 2.1: the centroid method, the centroid all method, which includes all devices regardless of whether they exceed the anomaly threshold, the distance weighted centroid, and the maximum reconstruction error method. Comparing multiple methods allows the trade-offs between continuous spatial estimation and discrete device snapping, and between using only confidently anomalous devices versus all devices, to be assessed empirically.

The localization findings are visualized as spatial heatmaps showing the distribution of estimated source positions across all drops at a given location, with the true source position marked for reference, as shown in Figure 3.4. This visualization makes it straightforward to assess whether the estimated positions cluster around the true source or are pulled towards nearby devices, and to compare the spatial behavior of different methods and modalities side by side.

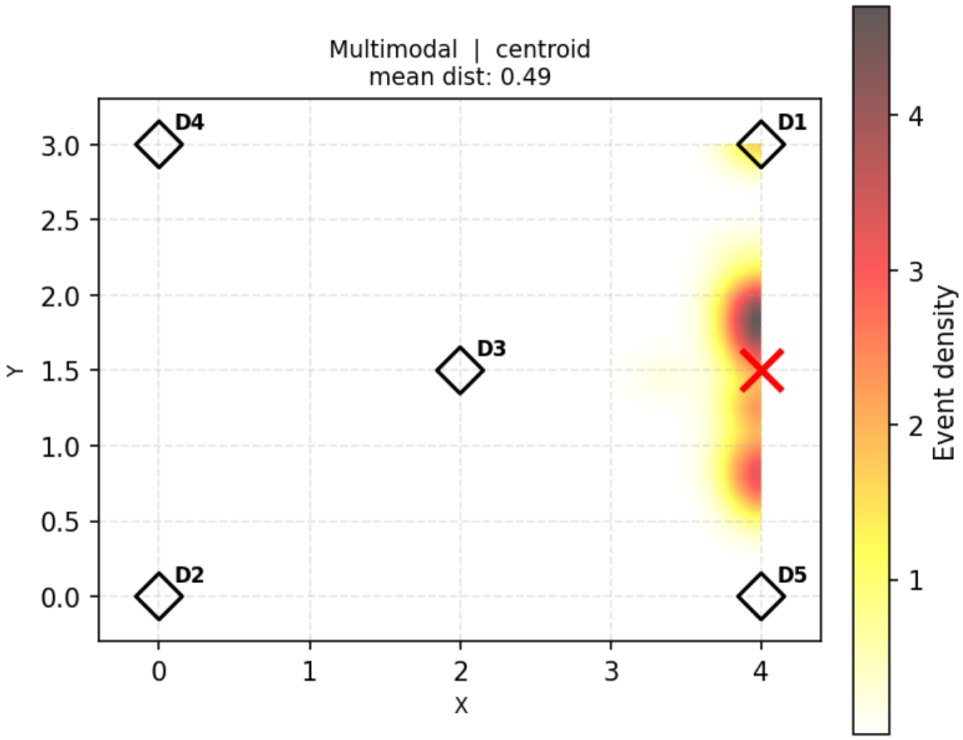

Figure 3.4: Anomaly Localization Heat Map

Since the same model is used for both anomaly detection and localization, the per-node reconstruction errors that drive the detection decision are the same errors used to estimate the source position. While it would, in principle, be possible to use dedicated models with different reconstruction characteristics for each stage, this was not pursued here in order to allow a comparison of the full modality pipelines end to end.

To summarize the anomaly detection and localization pipeline at inference time: a new recording is first preprocessed into fixed-size windows of log-mel spectrograms and continuous wavelet packet scalograms for each device. Each window is normalized using the per device statistics computed during training and fed through the trained autoencoder, which produces a reconstruction for each window. The reconstruction error is computed

as a scalar per window per device and compared against the per device threshold derived from the training distribution. Windows exceeding the threshold are flagged as anomalous. For each flagged window, the per device reconstruction errors are normalized using the training distribution and passed to the localization module, which estimates the spatial position of the anomaly source using one of the four localization methods. The result is a sequence of flagged windows, each associated with a spatial estimate of where the anomaly originated within the monitored environment.

Chapter 4

Implementation

The full implementation is available as an open source repository at `https://github.com/maximilianhuwyler/DAVSIPM`. The repository is organized into two top level directories. The `ESP32/` directory contains the Arduino firmware for the sensor nodes, including configuration and flashing instructions in `ESP32/README.md`. The `Pipeline/` directory contains the full server side data pipeline, divided into four self-contained modules: `data_acquisition/`, `preprocessing/`, `autoencoder/`, and `localization/`. Each module includes its own Dockerfile, shell scripts for common operations, and a README file with detailed usage instructions. The following sections describe the implementation of each component, with the README files serving as the primary reference for running the system.

4.1 Sensor Nodes

The following describes the concrete hardware components and firmware implementation of a sensor node responsible for data collection.

Hardware

Each sensor node consists of an INMP441 omnidirectional MEMS microphone and an MPU-6050 inertial measurement unit mounted on an ESP32-S3 microcontroller. The INMP441 is a high-performance, low-power digital microphone with an I2S interface, offering a flat frequency response suitable for capturing acoustic events [43]. The MPU-6050 combines a 3-axis gyroscope and a 3-axis accelerometer in a single package, communicating via I2C and providing high-resolution vibration measurements [44]. The ESP32-S3 is a low-cost, low-power system-on-chip featuring a dual-core 32-bit processor and Wi-Fi connectivity, making it well suited for distributed embedded sensor applications [45]. Both sensors are wired to the ESP32-S3 via extension boards, and the node is powered through a USB-C connection (see Figure 4.1). The firmware is written in C++ using the Arduino IDE [46] and flashed onto the device via USB-C.

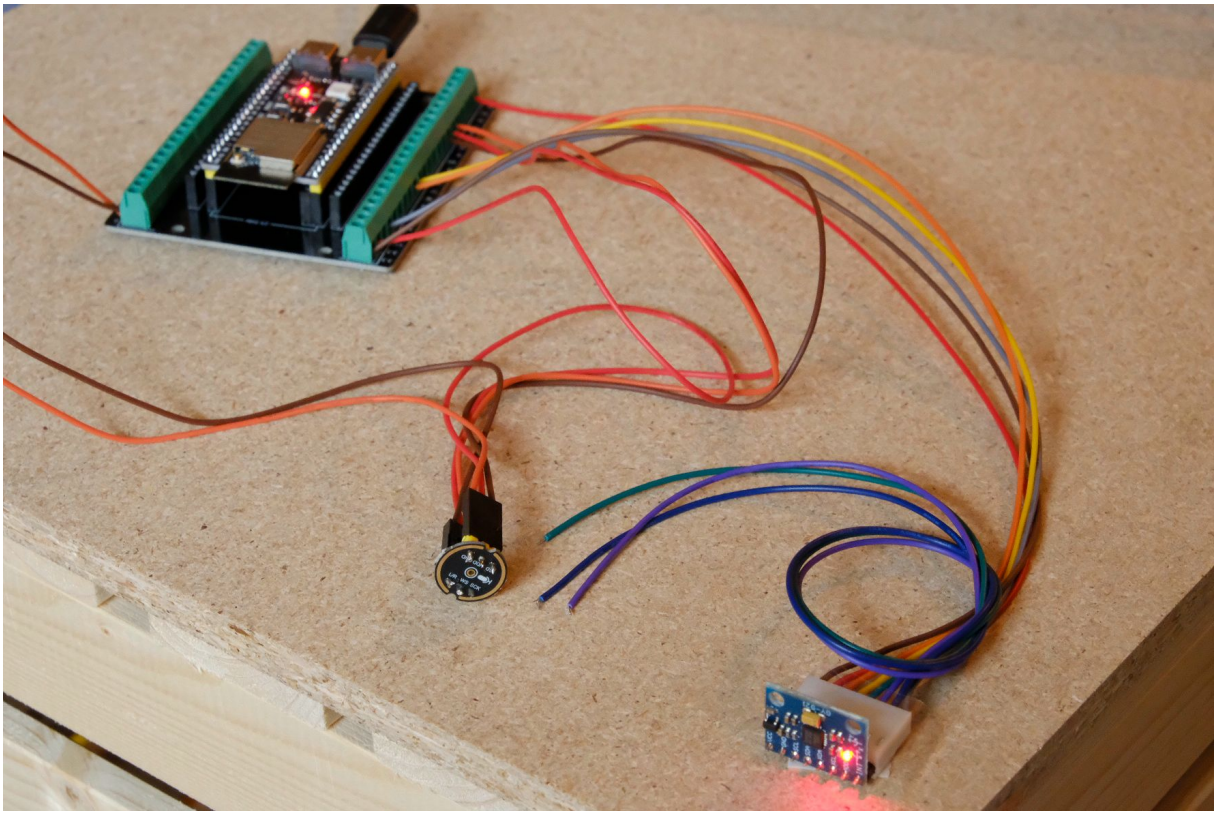

Figure 4.1: Implemented Sensor Node

Firmware

The ESP32-S3 features a dual-core processor, which is exploited using FreeRTOS [47] to handle the two sensors in parallel without interference between the time-critical sampling tasks and network communication. The MPU-6050 sampling runs as a dedicated FreeRTOS task pinned to Core 0 using the function `xQueueCreate()`. Core 1 runs the Arduino `loop()` function, which handles audio acquisition and all network communication. Data is exchanged between the two cores through a FreeRTOS queue, which acts as a thread-safe buffer, decoupling the sampling rate from the transmission rate. The queue holds up to 20 batches, providing approximately one second of buffer capacity. If the queue fills up, the oldest batch is dropped to make room for incoming data, preventing the sampling task from blocking.

The MPU-6050 is sampled at 1 kHz on Core 0 using a dedicated FreeRTOS task. To ensure a consistent sampling rate, `vTaskDelayUntil` is used to wake the task at fixed absolute intervals, regardless of the time spent executing within the loop. Each iteration reads 14 bytes from the MPU-6050 via I2C, covering three axes of acceleration, three axes of angular velocity, and temperature. Each sample is immediately prepended with a 64-bit microsecond timestamp and written into a batch buffer using `memcpy`. Once ten samples are accumulated, the batch is pushed onto the inter-core FreeRTOS queue, where it is picked up by Core 1 for transmission.

At the start of each main loop iteration, the NTP clock synchronization is checked. If more than 30 seconds have elapsed since the last sync, `syncTime()` is called to resynchronize

the system clock with the home router, which acts as the local NTP server. This ensures that the microsecond timestamps prepended to every data packet remain accurate and consistent across all sensor nodes throughout a recording session.

Any MPU-6050 batches that have accumulated in the FreeRTOS queue are then drained and published via MQTT. This is done before the audio read to minimize queue buildup on the vibration side. The loop then calls `i2s_read()`, which blocks until 256 audio samples are available from the DMA buffer, corresponding to approximately 5.8 ms of audio at 44.1 kHz. This blocking call effectively paces the loop, determines how frequently the queue is drained, and how often MQTT and WebSocket connections are serviced. Once the audio samples are read, they are prepended with a 64-bit microsecond timestamp and transmitted as a binary WebSocket frame using `wsClient.sendBinary()`. Both the MQTT and WebSocket connections include reconnection logic so that a temporary network disruption does not permanently halt data transmission.

4.2 Data Acquisition

The server-side data acquisition system is orchestrated using Docker Compose [48], with each component running in an isolated container on a shared internal network. The stack consists of four services: a Mosquitto MQTT broker [34], an MPU-6050 subscriber, an Nginx [49] reverse proxy, and an INMP441 audio subscriber. Together, they form a continuous data collection pipeline: the broker and proxy handle incoming connections from the sensor nodes, while the two subscribers receive, decode, and persist the data to disk. The entire system is started with a single command, `docker compose up`. Recording is terminated with `./stop_recording.sh`, which shuts down all containers, corrects a systematic one-second timestamp offset in the MPU-6050 output files, and prints a per-device data quality report. The following sections describe each component.

Mosquitto MQTT broker

The Mosquitto broker [34] serves as the message broker for the vibration data stream. It receives binary data packets published by the ESP32 sensor nodes on device-specific topics of the form `mpu6050/<device_id>` and forwards them to the MPU-6050 subscriber.

MPU-6050 Subscriber

The MPU-6050 subscriber listens on five device-specific MQTT topics of the form `mpu6050/<device_id>`, one for each sensor node. The `on_message()` callback is triggered for each incoming message, extracting the device ID from the topic string and unpacking the binary batch payload. Each sample consists of an 8-byte timestamp followed by 14 bytes of raw register data, which are converted to physical units. Converted samples are accumulated in a per-device `csv_buffer` and flushed to disk in batches of 200 samples via `save_sensor_data_csv()`. On shutdown, `flush_buffers()` writes any remaining data.

Each recording session produces one `.csv` file per device, containing the converted physical values alongside timestamps and transmission latency.

Nginx WebSocket Proxy

Nginx [49] acts as a reverse proxy in front of the audio subscriber, accepting incoming WebSocket connections from the ESP32 nodes on port 443 and forwarding them to the INMP441 subscriber running internally on port 8080. This allows all sensor nodes to connect to a single well-known port without being aware of the internal service structure.

INMP441 Subscriber

The INMP441 subscriber is an asynchronous Python program built on `asyncio` and the `websockets` library. When an ESP32 node connects, `handle_esp32_connection()` is called once and remains alive for the duration of that connection. It passes the WebSocket path to `extract_device_id()` to identify which device is connecting, using the path format `/inmp441/<device_id>`. The server handles all five nodes concurrently within the same event loop, switching between connections.

Each incoming binary frame is passed to `process_audio_data_optimized()`, which unpacks the leading 8-byte timestamp and converts the remaining bytes into a 16-bit PCM numpy array. The samples are appended to a per-device `deque` ring buffer and flushed to disk every 50 messages or every five seconds via the `FileWriterThread`.

Each recording session produces two output files per device: a `.raw` file containing a flat binary stream of 16-bit signed PCM samples appended continuously as audio arrives, and a `.csv` timestamp log recording the ESP32 timestamp and the number of samples for every incoming frame. These two files together allow the preprocessing stage to reconstruct the precise timing of each audio frame and map it to the correct position within a time window.

4.3 Preprocessing

For each one second window, the CSV timestamp log is searched to identify which WebSocket messages fall within the window by binary searching on the ESP32 timestamp. A coverage check ensures that the total number of samples across all identified messages reaches at least 95% of the expected one second worth of audio before proceeding. The corresponding audio samples are read from the raw file using the byte offset recorded for each message and placed into a silence buffer at the temporal position determined by the ESP32 timestamp relative to the window start. Messages that partially overlap the window boundary are clipped to fit, and dropped packets leave silence in the buffer. The buffer is then downsampled and converted to a log-mel spectrogram using `librosa` [50].

For vibration, a binary search on the IMU timestamps selects all samples within the window. A window is discarded if fewer than 95% of the expected samples are present or if all axes read exactly zero, indicating a sensor error. The CWT is then computed on the full window for each accelerometer axis using `PyWavelets` [51] and resampled to the target size using linear interpolation. A window is only retained if all modalities pass their validity checks across all devices. Normalization statistics are computed separately for audio and vibration in a streaming pass over all valid windows using Chan's parallel algorithm [52] and saved alongside the features for use during training and inference.

Several utility scripts are provided to assure the quality of the recordings after preprocessing. `visualize_samples.py` renders the LMS and CWT scalograms for all devices and axes in a selected window. Each row corresponds to one device and shows the LMS alongside the accelerometer and gyroscope CWT scalograms for all three axes, allowing a quick visual check of feature quality and cross-device consistency for any given moment in a recording. `visualize_windows.py` produces a per-device window coverage graph, making it straightforward to identify gaps in the data caused by dropped packets or connectivity issues. `compare_normalization_stats.py` compares normalization statistics across sessions to check for distribution shifts between recordings and devices.

4.4 Anomaly Detection

Three autoencoder classes are implemented in Keras [53] (via TensorFlow [54]): `autoencoder.py` for the multimodal model, `audio_autoencoder.py` for the audio-only variant, and `vibration_autoencoder.py` for the vibration-only variant. All three follow the same encoder-decoder structure described in Section 3.5 and are saved as `.keras` files upon training.

Training is invoked via `run_training.sh`, which accepts a mode argument (`audio`, `vibration`, or `multimodal`), one or more session timestamps, and optional overrides for epochs, batch size, and learning rate. The default configuration uses 50 epochs, a batch size of 32, and a learning rate of 10^{-3}. `ModelCheckpoint` saves the best model checkpoint based on validation loss throughout training. All three variants can be trained in sequence using `run_training_all_modes.sh`. Once training is complete, `run_error_statistics.sh` runs the trained model on the normal training sessions and computes per-device reconstruction error statistics, including mean, standard deviation, minimum, maximum, and sample count, saving them to `error_stats.json`. This file defines the normal baseline and is used for anomaly thresholding during inference and localization.

Inference on a new session is run via `run_inference.sh`, which takes a model name and session timestamp as arguments. The autoencoder reconstructs each window for each device, and the reconstruction error is saved to a compressed NPZ file. The anomaly threshold is not applied during inference itself but at a later stage during visualization and localization, where the NPZ file containing the new session's reconstruction errors is loaded together with the `error_stats.json` containing the per-device mean and standard

deviation computed from the training data. The threshold of mean plus k standard deviations is then applied on the fly, allowing the threshold to be adjusted without rerunning inference.

Two visualization scripts support result inspection. `run_plot_inference.sh` plots the reconstruction error per window per device as a time series, with the threshold overlaid, serving as a sanity check during development (see Figure4.2). `run_visualize_reconstruction.sh` renders the original and reconstructed LMS and CWT scalogram side by side for specific windows and devices. ROC curves comparing all three modalities are computed via `run_roc.sh`, which was used to determine the appropriate threshold multiplier by evaluating detection performance across a range of standard deviation values.

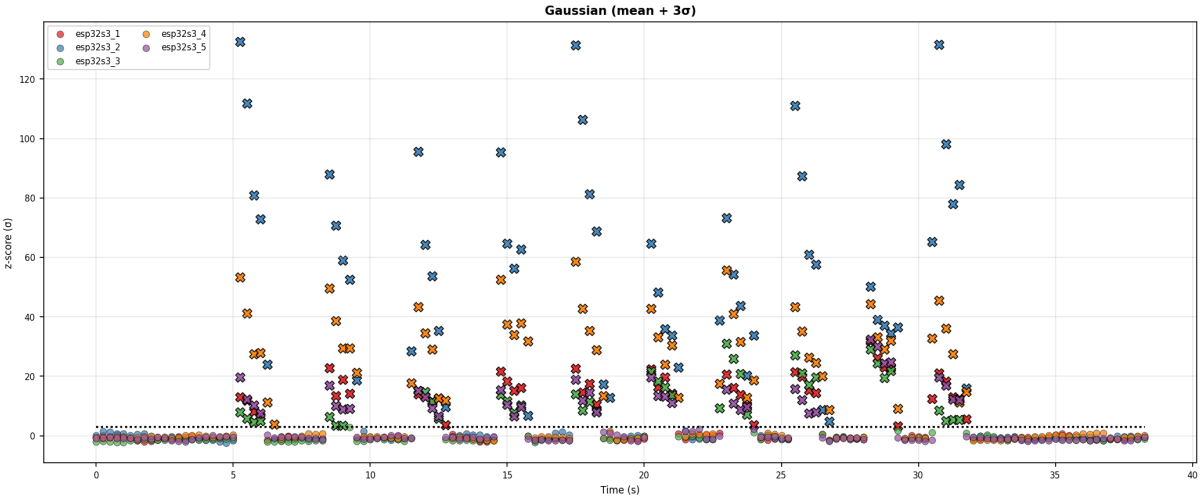

Figure 4.2: Visualizing Reconstruction Errors of 10 Anomalies

4.5 Localization

The localization logic is implemented in `localization.py`. The core `compute_locations()` function takes the per-device reconstruction errors from the inference NPZ and the error statistics JSON as input, applies the per-device threshold, and passes the normalized reconstruction errors to the localization algorithms described in Section 2.1, producing a list of estimated source coordinates together with the contributing device scores and window range for each event. The `plot_localization()` function renders the estimated locations as a density heatmap using a two-dimensional histogram with Gaussian smoothing. The `localization_error()` function computes the mean, median, and standard deviation of distances between estimated locations and the true source position, providing the localization metrics reported in the evaluation chapter.

Four analysis scripts build on the core algorithm. `run_localization_grid.sh` generates a heatmap grid for a single session showing all modality and method combinations side by side, with rows corresponding to modalities and columns to methods.

`run_localization_intensity_grid.sh` produces a spatially arranged 3×3 heatmap grid for a single modality and method across all nine source locations, mirroring the physical layout of the experimental area. `run_localization_method_comparison.sh` computes the mean distance and hit rate for all four methods across all sessions and saves the results to a JSON file, producing the aggregate results reported in the evaluation chapter. `run_localization_location_comparison.sh` groups the same results by source location.

4.6 From Recording to Localization

The repository is cloned from `https://github.com/maximilianhuwyler/DAVSIPM`. Docker and Docker Compose [48] are required to run the server side pipeline, and the Arduino IDE [46] is required to flash the sensor node firmware.

Before flashing, the sensor node firmware is configured by editing `ESP32/PublisherHybrid/config.h` with the WiFi credentials, server address, and a unique device ID for each node. The pin assignments for the INMP441 and MPU-6050 must also be verified and adjusted to match the physical wiring of each node. The firmware is then flashed to each ESP32-S3 via the Arduino IDE over USB-C. The experimental layout used in this work is described in Chapter 5. For a different setup, the device coordinates in `localization/localization.py` must be updated to reflect the physical positions of the deployed nodes.

Data acquisition is started by running `docker compose up` in the `Pipeline/-data_acquisition/` directory, after which the system automatically begins collecting audio and vibration data from all connected nodes. Normal sessions are recorded for training, and anomaly sessions are recorded separately for inference. Stopping the recording via `./stop_recording.sh` shuts down the containers, fixes any timestamp offsets between the audio and vibration files, and prints a preliminary quality report. Preprocessing is then run for each session via `run_preprocessing.sh -timestamp <timestamp>`, after which `visualize_windows.py` should be used to inspect the window coverage, since the preliminary quality report is only an estimate. Sessions with insufficient coverage should be discarded before training.

With the preprocessed data in place, all three autoencoder variants are trained on the normal sessions via `run_training_all_modes.sh -timestamps <ts1> <ts2> ...`, which runs the audio, vibration, and multimodal training in sequence using the default configuration. Once training is complete, `run_error_statistics.sh -model-name <name> -timestamps <ts1> <ts2> ...` is run on the same normal sessions to compute the per-device reconstruction error baseline, producing the `error_stats.json` file used for anomaly thresholding. Inference on an anomaly session is then run via `run_inference.sh -model-name <name> -timestamp <timestamp>`, yielding a compressed file of per-device, per-window reconstruction errors.

Finally, localization results are visualized by running `run_localization_grid.sh` with the three model stems and the session timestamp alongside the true source coordinates,

producing a heatmap grid showing the estimated source positions across all modality and method combinations. For further details on individual script arguments and advanced usage, the README files in each module directory and the comments within the shell scripts serve as the primary reference.

Chapter 5

Evaluation

This chapter presents the experimental evaluation of the proposed anomaly detection and localization system. The experimental design, including the physical setup, sensor placement, and recording procedure, is described in Section 5.1. Detection performance is evaluated in Section 5.2 using threshold-independent and fixed-threshold metrics across all three models and intensity levels. Localization performance is evaluated in Section 5.3, comparing localization methods across modalities, positions, and intensity levels. The results are analyzed and discussed in Section 5.4, followed by a discussion of the limitations of the experimental setup and the transferability of the findings to real-world deployments in Section 5.5.

5.1 Experimental Design

The intended application of this work is to develop a prototype pipeline for anomaly detection and localization in the machine room of a pump storage plant. Conducting experiments directly in such an environment is impractical for several reasons. Access to the facility is restricted, making it difficult to collect sufficient amounts of data that are necessary to train convolutional models [55]. Real anomalies are rare and cannot be deliberately triggered without safety risks, which makes it difficult to obtain reliable inference data. The complex background noise from the machinery makes it hard to isolate individual events [5]. More stable background noise can be induced in a controlled manner for developing a prototype.

A controlled living-room setup was chosen as a proxy environment for the development and evaluation of the system. This allows full control over the time, place, and types of anomalies, making it feasible to generate more training and inference data on demand. A living room differs acoustically from a machine room. We assume that the underlying signal processing and localization concepts are general enough to transfer. Small pumps should be used to simulate a mini pump storage power plant on which the experiments can be conducted.

As mentioned before, the background noise can be induced but should be controlled during prototype development. Ideally, a source is used that can produce both noise and vibration, such that there is a constant background vibration simulating an operating machine room. The small pumps should run at a controlled, constant pressure as well. The combination of these should give us a baseline that represents a normally operating machine room. This contrasts with the artificially induced anomalies, whose intensity, placement, and timing are varied during experimentation. Anomalous events can be artificially induced by hitting or knocking on the surface on which the sensor setup is placed.

Experimental Setup

The experimental setup is built on a table measuring $80 \times 120\,cm$. Two wooden boxes, 32 cm and 16 cm in height, respectively, are used to create different elevation levels. A water basin with a capacity of 5 liters is placed directly on the table in one corner; a second basin of the same size is placed on top of the taller box in the diagonally opposite corner, creating a height difference between the two reservoirs. Two aquarium pumps [56] inside the lower basin continuously pump water up to the higher basin, and a third pump inside the upper basin returns the water back down, establishing a closed circulation loop. The basins are connected by elastic PVC tubes. A vibration unit is mounted directly to the table surface and secured with a tie-down strap, serving as a controllable source of acoustic and mechanical disturbance. All the components are powered via a multi-outlet strip mounted on the table.

This setup simulates a small-scale pumping system with continuous mechanical activity, providing a realistic and repeatable environment for evaluating the anomaly detection and localization system. The experimental setup is shown in Figures 5.2 and Figure 5.1.

Sensor Placement

The sensor nodes are mounted either directly on the small pumps or placed on the surface within the experimental area. Their arrangement is such that they enclose a defined region of $60 \times 80\,cm$ into which all anomalies are introduced, ensuring that every anomaly falls within the sensor array and is simultaneously observable from multiple nodes. The localization approach relies on the assumption that both sound and structural vibration decay with increasing distance from the source [24, 25], causing an anomaly to produce a stronger response at nearby sensors than at distant ones.

Devices 1 and 2 are mounted on the small pumps located in two corners of the table: Device 1 on the lower basin and Device 2 on the higher basin. Device 3 is positioned in the center of the table, resting on a raised wooden surface mounted on a box. Devices 4 and 5 occupy the two remaining corners, with Device 4 resting on a raised wooden surface mounted on a box and Device 5 on the table surface. The primary source of background noise and vibration is situated at the end of the table closest to Devices 1 and 4, at a position slightly closer to Device 4. Figure 5.3 shows the abstract plan of the experimental setup, which directly corresponds to Figure 5.1.

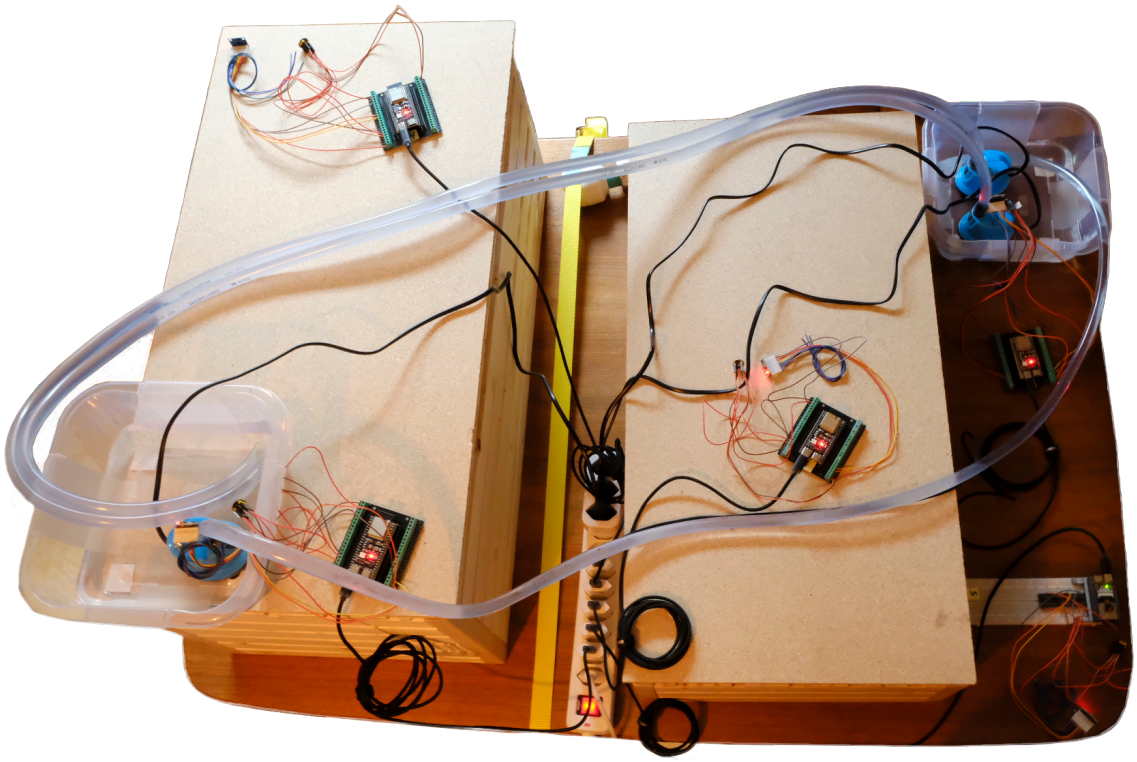

Figure 5.1: Experimental Setup (Bird View)

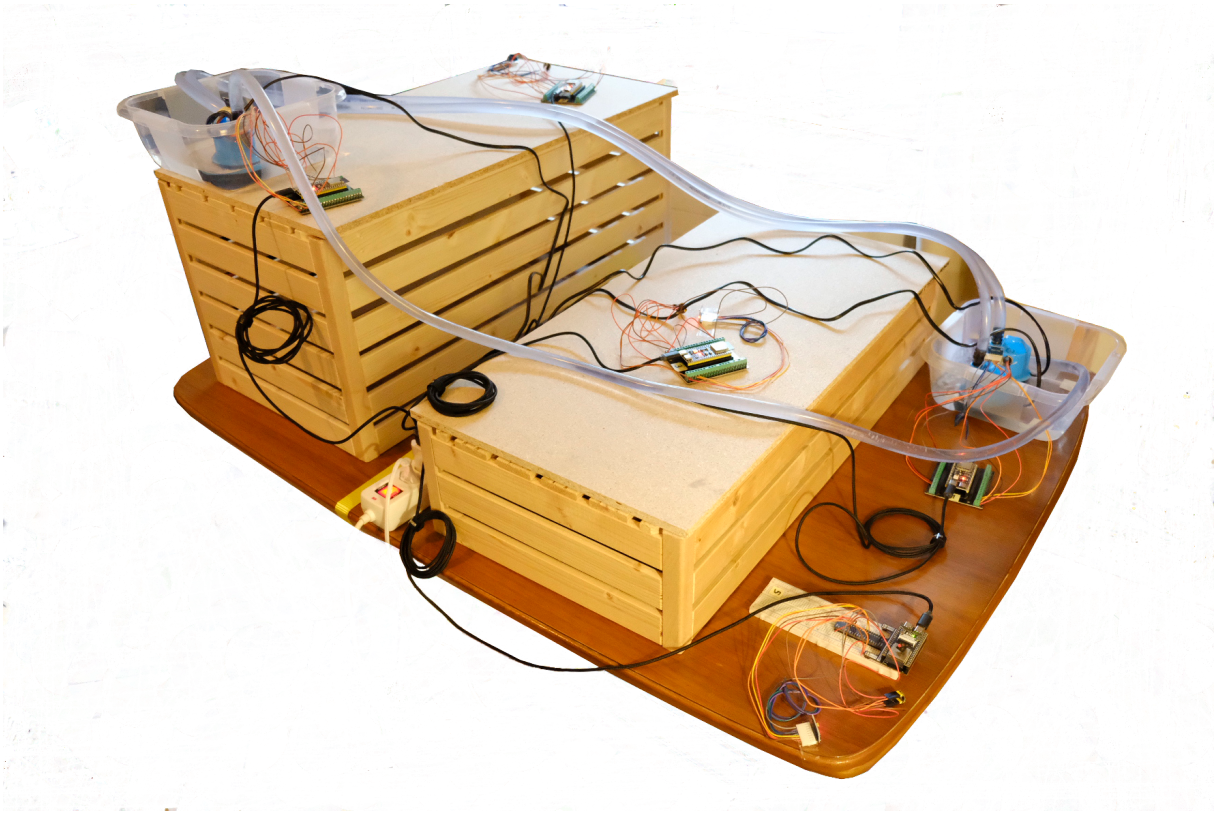

Figure 5.2: Experimental Setup (Side View)

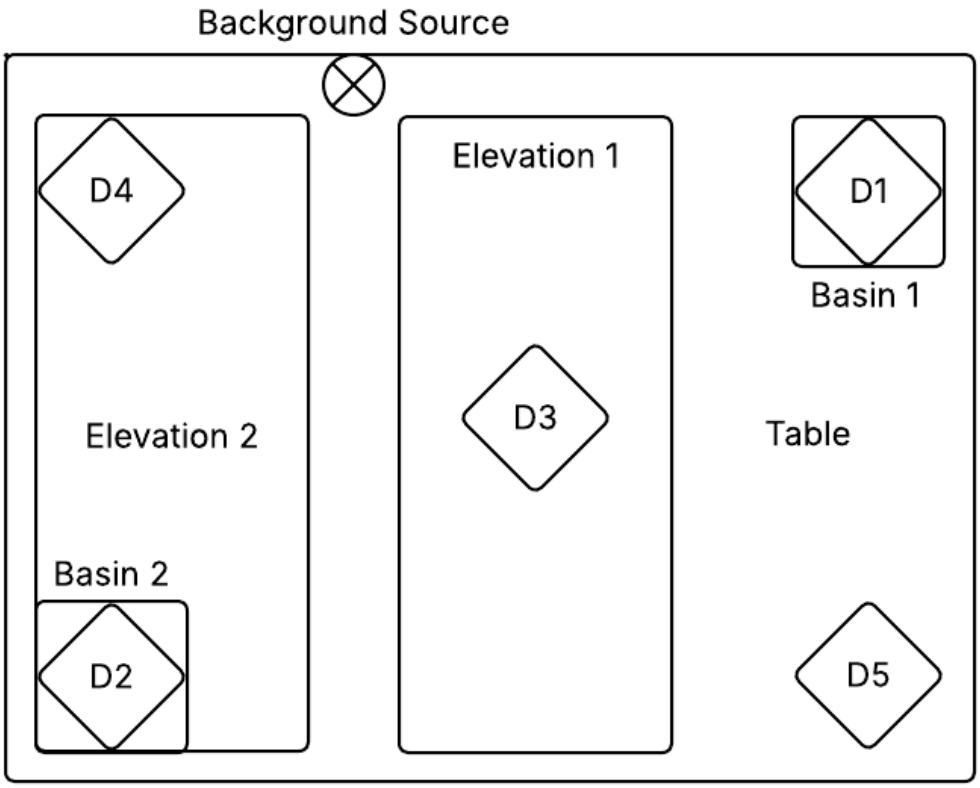

Figure 5.3: Conceptual Plan (Bird View)

Recordings

Anomalies are introduced by dropping a ball at nine predefined positions distributed across the experimental area. The positions are chosen such that each device has a nearby drop point, with additional positions placed at intermediate locations between neighboring devices, collectively covering the full spatial extent of the sensor array. Each position is tested at three drop heights of approximately $1\,cm$, $5\,cm$, and $15\,cm$, representing low, medium, and high anomaly intensity levels respectively, resulting in 27 distinct anomaly conditions in total.

Each anomaly recording session consists of ten consecutive drops at a single position and height, with approximately two seconds between consecutive drops. Between anomaly sessions, a normal recording of three minutes duration is captured, reflecting the undisturbed operating state of the system. These normal recordings serve as training data for three separate autoencoder models: an audio-only model, a vibration-only model, and a multimodal model that combines both modalities. In total, the dataset comprises 27 anomaly recording sessions and 26 corresponding normal recording sessions. The normal recording sessions are used for model training, whereas the anomaly recordings are subsequently used for inference, on the basis of which both anomaly detection and spatial localization are performed.

5.2 Detection Results

The anomaly intervals within each inference recording were annotated manually using Audacity [57], an open-source audio editing tool, by marking the time intervals corresponding to each ball drop and its subsequent bounce sequence. A window is labeled as anomalous if it overlaps with at least 33% of any annotated anomaly interval, allowing short-duration anomalies to contribute to labeling without requiring full window coverage. The annotated intervals were then used to assign window-level labels for evaluation. Each model was evaluated across a range of thresholds, yielding a receiver operating characteristic (ROC) curve and corresponding area under the curve (AUC) score for each.

Figure 5.4 shows the ROC curves for all three models evaluated across all sessions. The vibration and multimodal models achieve AUC scores of 0.865 and 0.871 respectively, substantially outperforming the audio model, which reaches 0.762. The vibration and multimodal curves track closely throughout the full operating range, with the multimodal model showing a marginal but consistent advantage. The audio model, while still above chance, exhibits a notably lower true positive rate at any given false positive rate, suggesting that audio features alone are less discriminative for this type of anomaly.

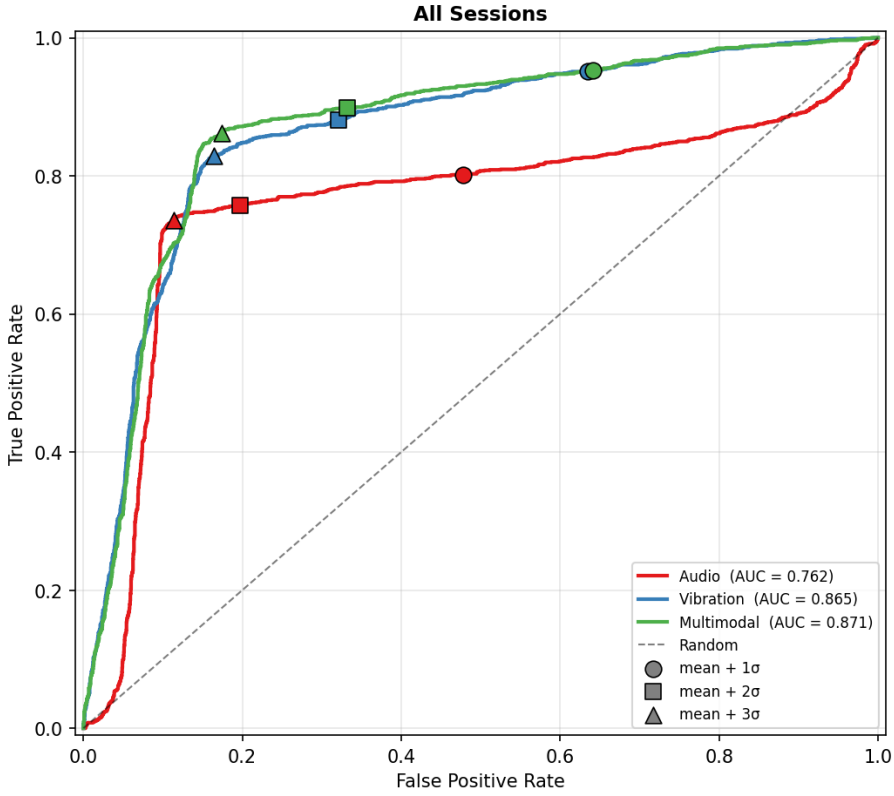

Figure 5.4: ROC Curves Across All Anomaly Sessions

The per-intensity results, shown in Figures 5.5–5.7, reveal a clear dependence on anomaly intensity. At low intensity (see Figure 5.5), the audio model performs poorly with an AUC of 0.575, while vibration and multimodal reach 0.813 and 0.832 respectively, with the multimodal model showing a more pronounced advantage over vibration than at other

intensity levels. This suggests that low-energy impacts produce vibration signatures detectable by the autoencoder but insufficient airborne sound for reliable audio-based detection. At medium intensity (see Figure 5.6), all three models improve, with vibration and multimodal reaching 0.884 and 0.883, respectively, and audio recovering to 0.802. At high intensity (see Figure 5.7), performance is strongest across the board, with vibration and multimodal achieving 0.922 and 0.924, while audio reaches 0.871. The convergence of all three models at high intensity indicates that large impacts produce strong enough signals across both modalities for the audio model to approach the performance of the vibration-based models.

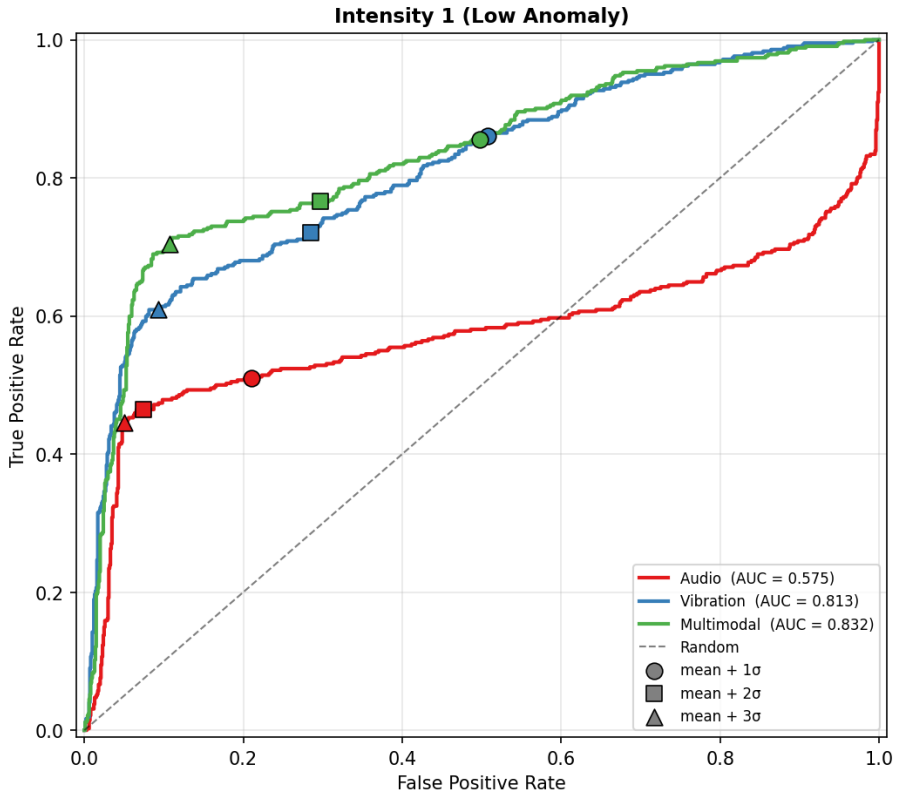

Figure 5.5: ROC Curves at Low Anomaly Intensity

Table 5.1 reports precision, recall, and F1-score at the fixed 3σ threshold, alongside the AUC-ROC for all three models across all intensity levels. Several patterns emerge from the fixed-threshold metrics that complement the ROC analysis. Across all sessions, the audio model achieves the most balanced precision-recall trade-off, with precision and recall of 0.731 and 0.736 respectively, yielding an F1-score of 0.733. The vibration and multimodal models achieve higher recall at 0.829 and 0.861, but at the cost of lower precision, resulting in comparable F1-scores of 0.747 and 0.757. This indicates that at the 3σ threshold, the vibration and multimodal models are more sensitive but produce more false positives than the audio model.

At low intensity, the balance between precision and recall breaks down sharply for the audio model. Precision rises to 0.790, but recall drops to 0.446, reflecting the model's inability to detect the majority of low-energy impacts while remaining conservative in its predictions. The vibration model improves recall to 0.609 with only a modest precision

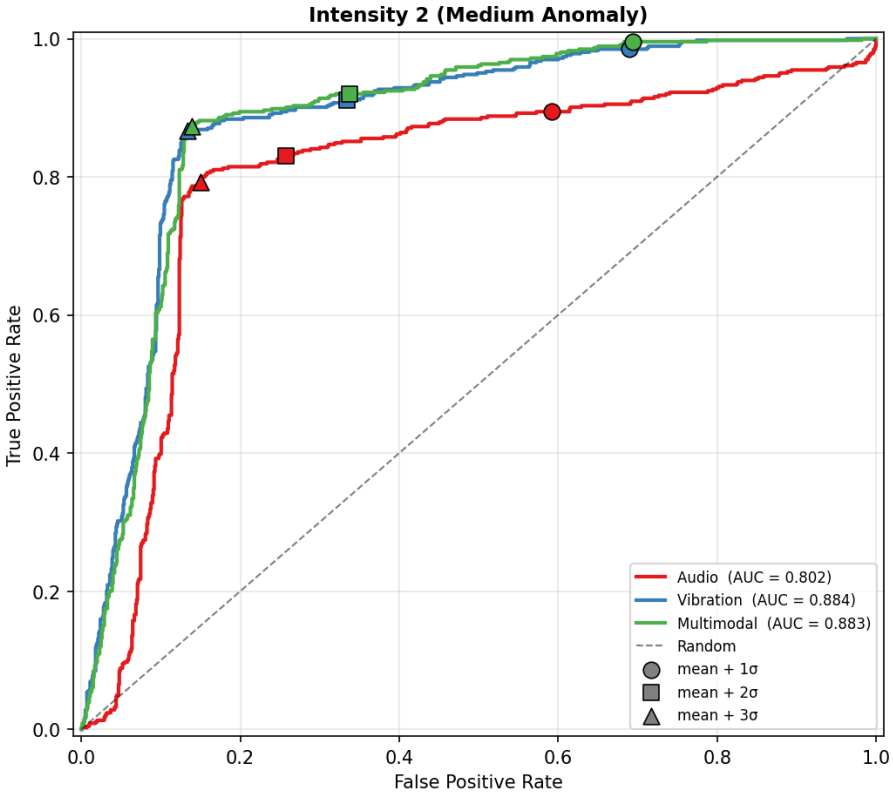

Figure 5.6: ROC Curves at Middle Anomaly Intensity

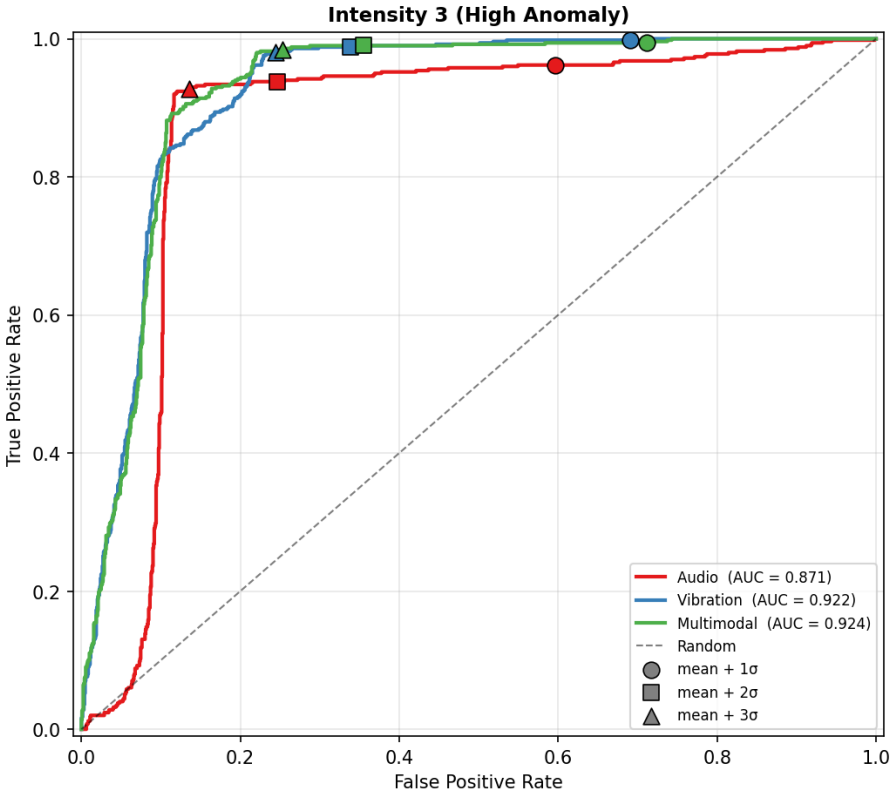

Figure 5.7: ROC Curves at High Anomaly Intensity

reduction, and the multimodal model achieves the best balance at this intensity level, with a recall of 0.704 and an F1-score of 0.718, the largest margin over vibration alone observed across all conditions. This suggests that the audio modality, while insufficient on its own at low intensity, contributes complementary information during fusion that improves detection of subtle anomalies. At medium intensity, vibration and multimodal perform nearly identically, with F1-scores of 0.804 and 0.803, while audio reaches 0.749. All three models exhibit a reasonable precision-recall balance, indicating that medium-energy impacts are reliably captured across modalities. At high intensity, a notable pattern emerges. The audio model achieves its highest precision of 0.726 with a recall of 0.928 and an F1-score of 0.814, making it the strongest performer in terms of F1 at this intensity level. The vibration and multimodal models, by contrast, achieve near-perfect recall of 0.980 and 0.984 but with substantially lower precision of 0.609 and 0.601, yielding F1-scores of 0.751 and 0.746. This suggests that at high intensity, the vibration and multimodal models flag almost every anomalous window but also produce a considerable number of false positives, whereas the audio model maintains a better balance.

Dataset	**Model**	**AUC-ROC**	**Precision**	**Recall**	**F1**
All Sessions	Audio	0.762	**0.731**	0.736	0.733
	Vibration	0.865	0.679	0.829	0.747
	Multimodal	**0.871**	0.675	**0.861**	**0.757**
Intensity 1 (Low)	Audio	0.575	**0.790**	0.446	0.570
	Vibration	0.813	0.732	0.609	0.665
	Multimodal	**0.832**	0.733	**0.704**	**0.718**
Intensity 2 (Medium)	Audio	0.802	0.709	0.793	0.749
	Vibration	**0.884**	**0.750**	0.866	**0.804**
	Multimodal	0.883	0.743	**0.873**	0.803
Intensity 3 (High)	Audio	0.871	**0.726**	0.928	**0.814**
	Vibration	0.922	0.609	0.980	0.751
	Multimodal	**0.924**	0.601	**0.984**	0.746

Table 5.1: Anomaly Detection Metrics at Threshold 3σ Across Intensity Levels.

5.3 Localization Results

Localization performance is evaluated using two metrics. The mean distance error measures the Euclidean distance between the estimated anomaly location and the true source position, averaged across all localization attempts, expressed in grid units where one unit corresponds to $20\,cm$. The hit rate measures the proportion of localization attempts for which the estimated position falls within a radius of one unit from the true source position.

An overview of localization performance across all sessions is given in Table 5.2. For the audio modality, the Centroid DW method achieves the lowest mean distance of 1.085 and the highest hit rate of 0.452. For vibration, Centroid All yields the lowest mean

distance at 1.764, while the Max method achieves the highest hit rate at 0.298. For the multimodal model, Centroid DW again produces the lowest mean distance at 1.601, while Max achieves the highest hit rate at 0.326. Across all modalities, the audio model consistently outperforms both vibration and multimodal in terms of mean distance and hit rate. The standard deviation of localization distance is relatively high across all methods and modalities, indicating considerable variability between predicted locations. The audio model shows smaller standard deviations than vibration and multimodal across all localization methods, while the Max method exhibits the highest standard deviations across all models.

Modality	**Method**	**Mean Distance**	**Standard Deviation**	**Hit Rate**
Audio	Centroid	1.177	0.564	0.395
	Centroid All	1.169	0.521	0.405
	Centroid DW	**1.085**	0.626	**0.452**
	Max	1.275	1.107	0.360
Vibration	Centroid	1.790	0.946	0.252
	Centroid All	**1.764**	0.874	0.246
	Centroid DW	1.779	1.042	0.269
	Max	1.931	1.536	**0.298**
Multimodal	Centroid	1.631	0.841	0.267
	Centroid All	1.630	0.789	0.261
	Centroid DW	**1.601**	0.944	0.287
	Max	1.719	1.432	**0.326**

Table 5.2: Localization Method Comparison Overall

Whether the anomaly source is located at a device position or at an intermediate position between devices has a pronounced effect on localization accuracy, as shown in Table 5.3. For anomalies at device positions, the Max method achieves the highest hit rate across all three modalities, reaching 71%, 54%, and 59% for audio, vibration, and multimodal, respectively, albeit with a complete absence of hits for between-device positions. For between-device positions, the centroid-based methods perform more consistently, with Centroid All achieving the best hit rate for vibration and multimodal at 24% each, and Centroid DW achieving 36% for audio.

The influence of anomaly intensity on localization accuracy is reported in Table 5.4. At low intensity, the Max method achieves the lowest mean distance and the highest hit rate for all three modalities, with audio reaching a mean distance of 0.692 and a hit rate of 56%. At medium intensity, results are more mixed across methods, with Centroid DW performing well for audio and multimodal. At high intensity, performance generally degrades for vibration and multimodal, while audio with Centroid DW achieves a hit rate of 44% and a mean distance of 1.132. A general trend is observable across all modalities and methods: as anomaly intensity increases, mean distance error tends to increase and hit rate tends to decrease, indicating that higher-energy impacts are harder to localize accurately. The audio model with the Centroid DW methods is the only configuration yielding consistent localization results across all intensities.

		Device Locations		Between Devices	
		Mean Distance	Hit Rate	Mean Distance	Hit Rate
Audio	Centroid	1.061	48%	1.253	35%
	Centroid All	1.064	48%	1.239	36%
	Centroid DW	0.903	58%	**1.227**	**36%**
	Max	**0.835**	**71%**	1.677	0%
Vibration	Centroid	1.673	26%	1.907	22%
	Centroid All	1.681	24%	**1.836**	**24%**
	Centroid DW	1.558	35%	2.049	15%
	Max	**1.451**	**54%**	2.496	0%
Multimodal	Centroid	1.513	28%	1.729	23%
	Centroid All	1.535	27%	**1.698**	**24%**
	Centroid DW	1.372	37%	1.842	17%
	Max	**1.188**	**59%**	2.302	0%

Table 5.3: Localization Performance at Device Positions vs. Between Devices

		Intensity 1		Intensity 2		Intensity 3	
		MD	HR	MD	HR	MD	HR
Audio	Centroid	0.823	52%	1.183	40%	1.220	38%
	Centroid All	0.844	52%	1.162	42%	1.217	38%
	Centroid DW	0.762	54%	1.079	**45%**	**1.132**	**44%**
	Max	**0.692**	**56%**	**1.065**	41%	1.481	30%
Vibration	Centroid	1.478	31%	1.590	27%	1.978	**23%**
	Centroid All	1.387	28%	1.570	27%	**1.964**	22%
	Centroid DW	1.419	37%	**1.539**	31%	2.000	22%
	Max	**1.349**	**43%**	1.546	**38%**	2.286	22%
Multimodal	Centroid	1.277	38%	1.456	28%	1.821	23%
	Centroid All	1.275	35%	1.462	27%	**1.817**	23%
	Centroid DW	1.167	44%	**1.388**	33%	1.834	22%
	Max	**0.992**	**50%**	1.445	**38%**	2.062	**25%**

Table 5.4: Localization Performance per Intensity Level

Figure 5.8 shows the localization estimate density for the audio model combined with the Centroid DW method across all sessions and drop positions, with each subplot corresponding to one position. The red cross marks the true source location, the diamonds indicate device positions, the dashed circle represents the one-unit hit radius, and the heatmap shows the spatial distribution of localization estimates across all drops at that position.

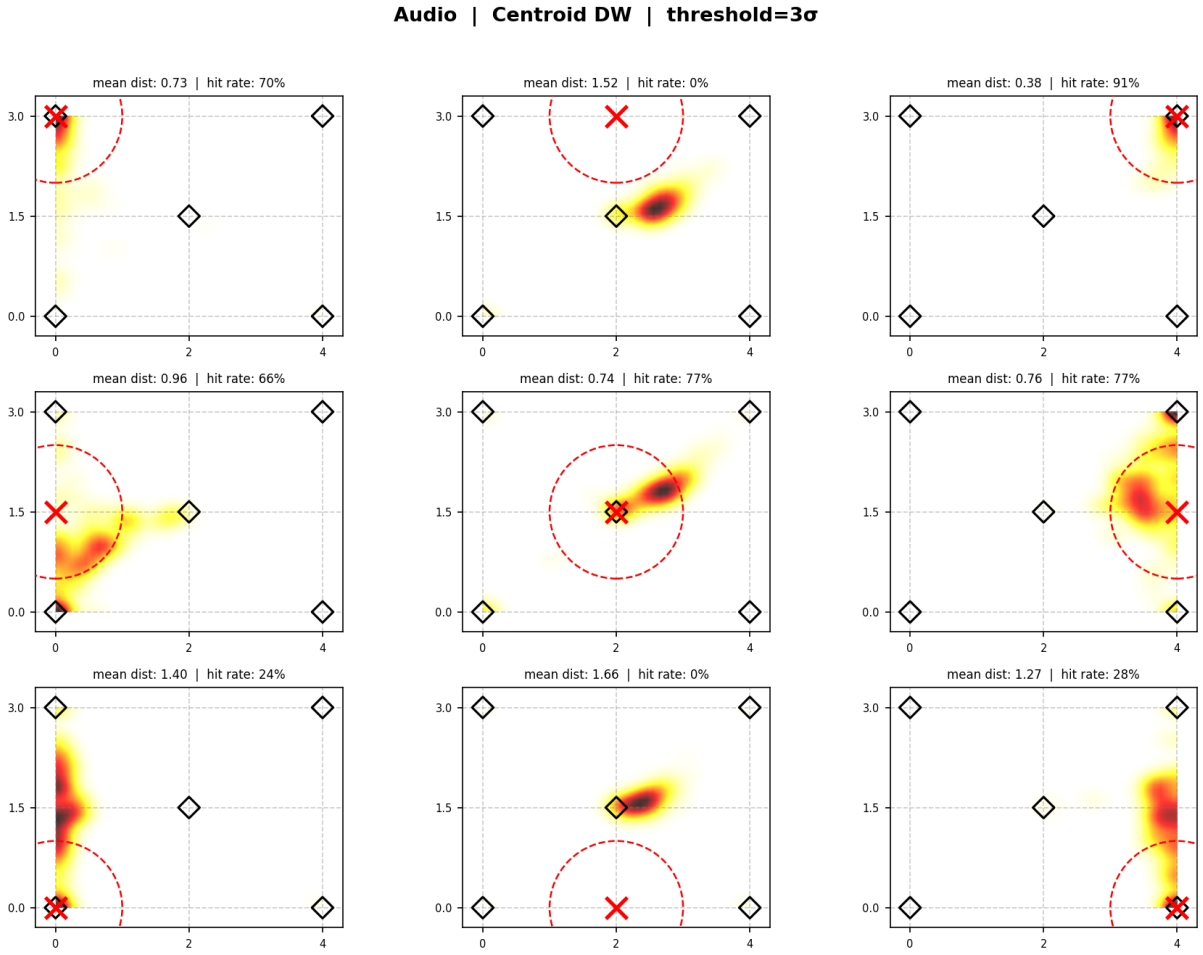

Figure 5.8: Localization Heat Map

The figure illustrates considerable variation in localization accuracy depending on drop position. Positions at or near device locations are generally well estimated. The top corners and the center position achieve hit rates of 91%, 70%, and 77% respectively, with heatmap peaks falling clearly within the hit radius. Positions along the left and right borders of the experimental area, situated between device locations, are estimated with sufficient accuracy, achieving hit rates of 66% and 77%. Localization at the bottom corners performs considerably worse, with estimates being drawn toward the upper corners rather than toward the true source positions. Positions along the top and bottom edges, situated between the corner positions, fail almost completely, with estimates collapsing toward the center and showing only a slight directional bias. In these cases, the signals perceived by the bottom and top sensor nodes are insufficient to contribute meaningfully to the localization and are dominated by the center device, preventing the spatial estimate from being pulled toward the correct region.

Nevertheless, several positions between device locations are localized with reasonable accuracy, demonstrating that the audio model combined with the Centroid DW method is not limited to device-proximal positions. Overall, this configuration yields the best localization results across all evaluated combinations and represents the trends observed in other configurations well. The corresponding localization heatmaps for all methods and modalities are provided in Appendix A.

5.4 Analysis of Findings

This section analyzes the experimental results presented in the previous section with respect to the three research questions guiding this thesis, examining how the detection and localization findings collectively contribute to answering each of the following:

- Does vibration sensing improve anomaly detection compared to audio alone in a noisy industrial environment, such as a hydropower plant machine room?
- Does combining acoustic and vibration sensing within a multimodal autoencoder framework yield additional detection gains over either modality individually?
- Can a low cost distributed sensor deployment without precise time synchronization estimate the spatial origin of anomalies within a machine room?

Anomaly Detection

Across all sessions, all three models demonstrate good anomaly detection capability, with reasonable AUC scores. The vibration modality contributes substantially to detection performance, as reflected in the strong vibration AUC and the further marginal improvement seen in the multimodal model, which benefits from the vibration signal as its primary discriminative source. This suggests that the multimodal model is able to exploit the stronger modality when the two modalities are not equally informative. A plausible explanation for the audio model's weaker performance is that the background noise in the experimental environment was relatively loud compared to the impact sound, whereas the background vibration was comparatively weaker relative to the impact-induced vibration, making the vibration signal inherently easier to discriminate. To confirm that the multimodal model can adapt to whichever modality provides the better signal-to-noise ratio, a complementary experiment with lower background noise and stronger background vibration should be conducted.

This pattern is further supported by the per-intensity ROC curves. At high intensity, all three models perform strongly, as the impacts of sufficient energy produce acoustic and vibration signatures that are clearly distinguishable from the background. As anomaly intensity decreases, detection performance degrades across all modalities, which is expected, assuming that weaker impacts produce signals closer to the normal operating baseline. However, the rate of degradation differs substantially between modalities. Vibration maintains good discriminability down to medium intensity but degrades more

gracefully than audio. Audio, by contrast, deteriorates rapidly at low intensity, reaching an AUC that is close to chance level. At the 3σ threshold level, the audio model produces few false positives but detects barely half of all anomalous windows, rendering it unreliable for low-intensity fault detection. This is consistent with the hypothesis that background noise drowns out low-energy impacts in the acoustic channel, while background vibration is insufficient to mask the corresponding vibration signature. Nevertheless, the vibration and multimodal models operating at the 3σ threshold constitute a solid anomaly detection system across all tested intensity levels, combining a low false positive rate with strong recall.

The fixed-threshold metrics in Table 5.1 offer additional insight into model behavior at the 3σ operating point. Across all sessions, the vibration and multimodal models achieve higher recall than audio at the cost of lower precision, reflecting their tendency to flag more windows as anomalous, including some false positives. The audio model, while showing a more moderate recall, maintains comparatively higher precision. At low intensity, this divergence becomes more pronounced: the audio model achieves its highest precision but its lowest recall, meaning it is highly conservative, only flagging windows it is confident about while missing the majority of anomalous events. The vibration and multimodal models, by contrast, maintain substantially higher recall even at low intensity. At high intensity, an interesting reversal occurs: the audio model achieves the highest F1-score, outperforming the vibration and multimodal models, which, despite near-perfect recall, suffer from lower precision, pulling their F1-scores down. This high recall, at the cost of precision for vibration and multimodal at high intensity, has direct implications for localization. A large number of flagged windows, including false positives, are passed to the localization module, which may introduce noise into the spatial estimates and partially explain the degraded localization performance observed at higher intensity levels.

The experimental results provide a clear answer to the first research question. Vibration sensing substantially improves anomaly detection compared to audio alone, with the vibration model achieving an overall greater ROC AUC than the audio model. The performance gap is most pronounced at low anomaly intensity, where the audio model degrades to near chance level while the vibration model maintains meaningful discriminability. A plausible explanation for this pattern is that the background noise in the experimental environment was sufficiently loud to mask low-energy acoustic impacts, while the corresponding vibration signatures remained distinguishable against the comparatively weaker background vibration. The results therefore support the conclusion that vibration sensing is a more robust modality for anomaly detection in acoustically noisy environments, though this finding is specific to the signal-to-noise conditions of the present experimental setup. A different acoustic environment in which background vibration was stronger relative to impact-induced vibration could shift the balance between modalities.

The answer to the second research question is less clear-cut. The multimodal model achieves a marginally higher overall AUC, and the gain is most visible at low intensity, where audio contributes some additional discriminative signal on top of vibration. However, the improvement is small and inconsistent across intensity levels. At high intensity, the multimodal and vibration models perform essentially equally, and the audio model actually achieves the highest F1-score due to its superior precision. The multimodal model appears to rely primarily on the vibration signal as its main discriminative source, with

audio playing only a secondary role. This limits the conclusions that can be drawn about the general benefit of multimodal fusion. To demonstrate that the fusion architecture can genuinely adapt to whichever modality provides the better signal-to-noise ratio, a complementary experiment would be needed in which the acoustic conditions favor audio over vibration, allowing the model to be evaluated in both directions. The present experiment tests only one of these two scenarios, and the marginal improvement observed here cannot be taken as sufficient evidence of robust multimodal adaptation.

Anomaly Localization

The overall localization results in Table 5.2 reveal a surprising pattern: the audio model achieves substantially better localization performance than both vibration and multimodal across all methods. Two factors likely contribute to this. First, the audio model is highly selective in what it flags as anomalous. As established in the detection results, it only detects the most clearly anomalous windows, meaning the samples passed to the localization module are cleaner and more confidently anomalous. Second, vibration propagation through the experimental setup is physically inconsistent across nodes due to the heterogeneous mounting conditions, with sensors placed on pumps, raised wooden surfaces, and bare table surfaces, as well as the path-dependent nature of structural vibration transmission. This makes the per-node reconstruction error pattern less reliable as a spatial cue. Audio, propagating through air, is comparatively more uniform across nodes. The similarity between vibration and multimodal localization performance is consistent with the detection results, where both models showed near-identical behavior. Since the multimodal model fuses both modalities but produces reconstruction error patterns that closely resemble those of the vibration model, it inherits the same localization weaknesses.

Table 5.3 reveals a clear performance split depending on whether the anomaly source is located at a device position or between devices. The Max method achieves the highest hit rates at device positions across all modalities but fails completely at between-device positions. This is a direct consequence of its discrete nature: the estimate is always snapped to one of the five device positions, making it structurally incapable of localizing anomalies that occur between nodes. The notably high standard deviation of Max across all modalities reflects this same property. In a denser sensor deployment, where device positions cover the space more finely, Max could become a viable approach. The centroid-based methods, by contrast, are capable of producing continuous spatial estimates and, therefore, perform more consistently at between-device positions. The generally poor between device performance across all methods suggests that when an anomaly occurs between devices, the reconstruction error at one node tends to dominate over the others, preventing the centroid estimate from being pulled towards the correct intermediate location. Furthermore, even at device positions, the Max method does not achieve perfect hit rates, indicating that reconstruction errors at non-proximal devices can occasionally exceed those at the nearest device. This represents a fundamental challenge for reconstruction error-based localization that would require further investigation, for instance, through a more detailed analysis of per-node reconstruction error distributions under different anomaly conditions.

A consistent trend is observed across all modalities and methods in Table 5.4: localization performance degrades as anomaly intensity increases, with mean distance error rising and hit rate falling from low to high intensity. This is counterintuitive at first glance, as stronger anomalies are detected more reliably. At low intensity, the Max method performs strongest across all modalities, suggesting that weak impacts are picked up predominantly by the nearest node while remaining largely below the detection threshold at more distant nodes, giving Max a clean signal to snap to. As intensity increases, the impact propagates more strongly across multiple nodes simultaneously, making it harder for Max to identify the correct device. A similar effect is observed for the centroid-based methods: at higher intensities, the reconstruction errors across nodes appear to represent the spatial distribution of the anomaly less reliably, pulling the centroid estimate away from the true source. Among all evaluated configurations, the audio model combined with the Centroid DW method is the only one that maintains consistent localization performance across all three intensity levels, suggesting that distance weighting combined with the more precise detection behavior of the audio model provides a degree of robustness that other configurations do not achieve.

A closer inspection of the per-position localization heatmaps reveals a clear spatial trend in accuracy. Most drop positions are localized with moderate success using the audio model, with estimates concentrating reasonably close to the true source (see Figure 5.8, Appendix A). A notable exception is the top and bottom center positions, located at the upper and lower edges of the array, where localization fails completely despite being within the sensor perimeter. This suggests that even for audio, the impact signal at these positions was not picked up sufficiently by the surrounding nodes, with the reconstruction error concentrated almost entirely at the nearest device, preventing the centroid from being pulled towards the correct location. For the vibration and multimodal models, localization results are consistently worse across positions, which is attributed to the inhomogeneous physical environment, where varying mounting surfaces and structural coupling conditions prevent vibration from propagating evenly across nodes, making the per-node reconstruction error an unreliable spatial cue. How localization algorithms can be made more robust against such propagation inhomogeneity would require dedicated further investigation.

The third research question asks whether a low cost distributed sensor deployment without precise time synchronization can estimate the spatial origin of anomalies within a machine room. The results demonstrate that reconstruction error based localization is feasible in principle, but the answer depends strongly on which modality and localization method are used.

The audio model combined with the Centroid DW method is the only configuration that shows consistent localization performance across all intensity levels. Two factors explain why audio outperforms vibration on localization despite being the weaker detection modality. First, the precise detection behavior of the audio model means that only the most confidently anomalous windows are passed to the localization module, resulting in cleaner and more reliable spatial estimates. Second, vibration propagation through the experimental setup is physically inconsistent across nodes due to the heterogeneous mounting conditions, with sensors placed on pumps, raised wooden surfaces, and bare table surfaces. The path-dependent nature of structural vibration transmission makes the per-node re-

construction error an unreliable spatial cue, whereas audio propagating through air is comparatively more uniform across nodes.

Several reasons were identified for why overall localization methods did not meet performance expectations. The centroid-based methods struggled in two ways: when the anomaly source was between devices, the estimate was often pulled too strongly toward one dominant node rather than settling at the correct intermediate location, and at device positions, reconstruction errors at nearby but incorrect devices occasionally pulled the estimate away from the true source. The Max method similarly did not meet expectations, as reconstruction errors at incorrect devices occasionally exceeded those at the nearest device, causing the estimate to snap to the wrong location. Localization performance also degrades at higher anomaly intensity across all configurations, as more intense anomalies produce reconstruction error patterns across nodes that represent the spatial origin of the fault less reliably, pulling the spatial estimates away from the true source. How reconstruction error based localization can be made more robust against these effects would require dedicated further investigation.

Overall, the results show that reconstruction error based localization could be a viable low cost alternative to classical time-based methods such as TDOA, requiring neither precise time synchronization nor specialized hardware. However, in its current form, the approach achieves only modest accuracy and robustness, and significant further work is needed to address the challenges of between-device localization, propagation inhomogeneity, and sensitivity to false positive contamination at higher anomaly intensities.

5.5 Limitations

This section discusses the key limitations of the work, covering the model architecture, the experimental setup, the annotation process, sensor placement, and the challenges associated with transferring the system to a real-world deployment.

Model Architecture

The three autoencoder models evaluated in this work share the same bottleneck architecture, which enables direct comparison across modalities but introduces a potential imbalance. The bottleneck was designed for the multimodal case, where two encoder branches of different input sizes feed into a shared latent space. When the same bottleneck is used for the unimodal variants, the ratio between input size and bottleneck capacity differs from the multimodal case, particularly for the vibration model, whose input is considerably smaller. This may give the multimodal model an architectural advantage that is not purely attributable to the additional modality, and how the architecture should be adjusted for unimodal settings to enable a fully fair comparison would require further investigation. Furthermore, detection and localization were always performed using the same model and the same modality. It remains unexplored whether using different modalities for detection and localization independently could improve overall system performance; for instance, by exploiting the stronger detection performance of vibration while

leveraging the better localization behavior of audio. Finally, only feature level fusion was employed in the multimodal model. A decision level fusion approach, where separate unimodal models each produce their own reconstruction errors and a dedicated fusion strategy combines these independently computed scores, could potentially enhance both detection and localization performance. Whether such an approach would outperform the feature level fusion evaluated here remains unexplored in this work.

Heterogeneous Experimental Setup

The heterogeneous physical setup is one of the primary limitations of this work. The varying mounting conditions, with sensors placed on pumps, raised wooden surfaces, and a bare table (see Figure 5.2 and 5.1), introduce inconsistency in how vibration propagates across nodes, making it difficult to attribute localization failures solely to the algorithm rather than the physical medium. A more homogeneous setup would have allowed for clearer conclusions about the localization approach itself. In the current setup, with loosely connected wood and plastic components, the vibration propagation chain can be interrupted unpredictably, meaning the reconstruction error at a given node may reflect whether vibration reached it at all rather than how far the source was. This reduces the utility of reconstruction error as a spatial cue for localization, since the signal behaves as effectively present or absent rather than decaying predictably with distance. Whether this is a fundamental limitation of vibration-based localization in heterogeneous environments or an artifact of this particular setup remains an open question that would require dedicated investigation to resolve.

Anomaly Annotation

The manual annotation of anomaly intervals in Audacity [57] introduces imprecision in the ground truth labels, as the exact onset and decay boundaries of each bounce sequence are difficult to mark consistently by hand. This directly affects the precision metric, since windows that the model correctly identifies as anomalous may fall just outside the annotated interval and are counted as false positives. The effect propagates further into the localization evaluation, as the detection and localization pipeline is treated as a whole rather than evaluated independently at the localization stage. Windows that are incorrectly labeled as normal due to annotation imprecision but are nonetheless flagged by the detection module are passed to the localization module and contribute spatial estimates that are then counted as errors, even if the underlying localization is correct. Conversely, true anomalous windows that fall outside the annotated interval and are missed by the detection module never reach localization at all, potentially skewing the spatial distribution of evaluated estimates. A more precise alternative would be to use timed, automated anomaly triggers that record the exact moment of each impact, allowing for precise window-level ground truth labels. However, this would require significant additional effort and was therefore not pursued in this work.

Sensor Placement

The sensor placement resulted in good localization accuracy at some positions but complete failure at others. The placement was chosen based on what seemed to be a natural way to cover the experimental area, without systematic optimization for spatial coverage. Different environments would require different sensor configurations, and further research into alternative sensor topologies and varying coverage densities would be needed to understand how placement affects localization performance. It is expected that the Max method, in particular, could benefit from a denser sensor deployment, as it relies on the nearest device having the highest reconstruction error, which becomes more likely when devices are positioned closer together. Additionally, the varying mounting conditions across devices influenced the results, and it would be beneficial to study a setup where all sensors are mounted in the same way to isolate the effect of sensor placement from the effect of mounting conditions on vibration propagation.

Real World Application

Beyond the experimental limitations of this setup, transferring the system to a real hydropower machine room introduces a distinct set of challenges. While normal operating data under steady-state conditions may be readily available, transition phases between generation and pumping modes represent distinct operating regimes that produce signal patterns differing substantially from steady-state operation. Data from these transitions is rarer and more costly to collect, and a model trained exclusively on steady-state normal data may produce elevated reconstruction errors during such transitions, regardless of whether a fault is present. Whether the learned normal representation transfers across operating regimes or whether separate models would need to be trained for each remains an open question.

The experimental evaluation is limited to a single anomaly type: ball drops at varying heights and positions, producing short impulsive impacts that decay rapidly. While this allows for controlled and reproducible experiments, it raises the question of whether the findings transfer to other anomaly types. Testing with a broader range of anomaly patterns would give more confidence in the generalizability and transferability of the results. More fundamentally, it is unclear whether anomalies in a real hydropower machine room would share the same acoustic and vibration characteristics as ball drop impacts. The relative performance of the models may shift considerably depending on the nature of the fault signature, and the conclusion that audio outperforms vibration in localization may be specific to the present anomaly type. To assess transferability to real machine faults, the acoustic and vibration signatures of actual anomalies in hydropower machinery would need to be analyzed and used to guide the design of more representative experimental conditions.

The acoustic and vibration environment of a real machine room is substantially more complex than the controlled experimental setup. Background noise and vibration originate from multiple concurrent sources, vary over time, and are harder to characterize, making it more difficult for the autoencoder to learn a stable normal representation and increasing

the risk of false positives. The physical scale and geometry of a real machine room also differ fundamentally from the tabletop experimental area, with larger distances between potential fault sources and sensors, more complex sound and vibration propagation paths, and a three-dimensional spatial layout that the current planar localization approach does not account for. Ultimately, the only way to assess how well the system performs under these conditions is to deploy it in a real machine room environment and evaluate it there directly.

Chapter 6

Final Considerations

This chapter summarizes the work presented in this thesis, draws conclusions with respect to the research questions, and outlines directions for future work.

6.1 Summary

Condition monitoring in industrial machinery is critical for preventing costly failures and ensuring safe operation. In hydropower machine rooms, access to the facility is restricted, and collecting sufficient data is challenging [55]. Hydropower machine rooms are characterized by strong and complex background noises [5]. These conditions make automated fault detection and localization a capability worth pursuing in such environments. While acoustic anomaly detection has been established as a viable approach across various industrial settings [5, 8–11], the potential of combining acoustic and vibration sensing within a unified distributed system for hydropower condition monitoring has not been explored.

This thesis presents the design, implementation, and evaluation of a low cost distributed multimodal anomaly detection and localization system without high precision time synchronization. Five sensor nodes, each combining an audio and a vibration sensor on a wireless microcontroller, were deployed across a small scale experimental setup simulating a pumped storage hydropower plant machine room. Each node continuously captures audio and vibration data, which is transmitted over a local network to a server side pipeline for preprocessing, model training, and inference.

Three autoencoder models were trained exclusively on normal operating data: an audio only model, a vibration only model, and a multimodal model combining both modalities through feature level fusion. Anomalies were artificially induced by dropping a ball at nine predefined positions across the experimental area at three intensity levels, yielding 27 distinct anomaly conditions. The models were evaluated on both anomaly detection and spatial localization performance.

Vibration and multimodal models outperformed the audio model on anomaly detection across all intensity levels, with the performance gap being most pronounced at low intensity, where the audio model performed near chance level. Multimodal fusion did not

consistently improve over vibration alone on detection, though it showed the most benefit at low intensity. Spatial localization without precise time synchronization was shown to be feasible, though the results were modest overall. Audio outperformed both vibration and multimodal models on localization, a result attributed to the selective nature of audio detection and the inconsistent vibration propagation across the heterogeneous physical setup.

6.2 Conclusions

This thesis set out to investigate three core research questions: whether vibration sensing improves anomaly detection compared to audio alone, whether multimodal fusion provides additional gains, and whether reconstruction error based localization can estimate the spatial origin of anomalies in a low cost distributed sensor deployment without precise time synchronization.

Regarding detection, vibration based sensing substantially outperforms audio across all tested intensity levels, with the performance gap being most pronounced at low anomaly intensity, where the audio model performs near chance level. The multimodal model consistently matches or marginally exceeds the vibration model, suggesting that feature level fusion can rely on the stronger modality when the two modalities are not equally informative. A plausible explanation for the audio model's weaker performance is that the background noise in the experimental setup was loud relative to the impact sound, whereas the background vibration was comparatively weaker relative to the impact induced vibration, making the vibration signal inherently easier to discriminate. This finding should be interpreted with caution, as it reflects the specific signal to noise conditions of this experimental setup, and a different acoustic environment could shift the balance between modalities. The contribution of multimodal fusion is most visible at low anomaly intensity, where the multimodal model shows the clearest advantage over vibration alone. At medium and high intensities, vibration and multimodal models perform comparably, indicating that audio adds little when the vibration signal is already highly discriminative.

The localization results show that the audio model substantially outperforms both vibration and multimodal models on spatial localization, despite being the weakest modality for detection. Two factors contribute to this. First, the audio model is more selective, passing only the most clearly anomalous windows to the localization module, resulting in cleaner spatial estimates. Second, vibration propagation through the heterogeneous experimental setup is physically inconsistent across nodes, making the per-node reconstruction error an unreliable spatial cue. Audio, propagating through air, is comparatively more uniform. Among all evaluated configurations, the audio model combined with the Centroid DW method is the only one that maintains somewhat consistent performance across all intensity levels. Overall localization accuracy remains modest, with several identified reasons: centroid-based methods are pulled too strongly toward dominant nodes, the Max method fails between device positions by design, and at higher anomaly intensity, the impact is picked up strongly by multiple nodes simultaneously, causing reconstruction errors that no longer reliably reflect the distance from the source.

From a practical perspective, the results demonstrate that reconstruction error based localization is feasible as a low cost alternative to TDOA in wireless deployments where precise time synchronization is not achievable. However, the approach in its current form achieves only modest accuracy and robustness, and significant further work is needed before it could be considered for operational use.

6.3 Future Work

The most immediate priorities concern the experimental setup. Testing in a more homogeneous environment, where all sensors are mounted on the same type of surface, would allow localization failures to be attributed more clearly to the algorithm rather than the physical medium. Closely related to this, testing with different background noise and vibration intensities would help establish under which signal-to-noise conditions each modality dominates and whether the multimodal model genuinely adapts to the stronger modality. Using automated anomaly triggers with precise timestamps would eliminate the imprecision introduced by manual annotation and allow detection and localization to be evaluated independently. Finally, testing with a broader range of anomaly types beyond ball drop impacts would reveal whether the observed modality preferences generalize and whether the conclusion that audio outperforms vibration in localization holds for other fault signatures.

On the system and algorithm side, the localization approach requires further development to become robust. Denser sensor deployments and alternative geometric layouts, such as a circular arrangement, could improve spatial coverage and reduce the number of positions where localization fails completely. Localization algorithms that are more robust to heterogeneous propagation conditions, for instance, by weighting nodes based on signal confidence rather than raw reconstruction error, represent a natural next step. The model architecture should also be revisited: the shared bottleneck designed for the multimodal case may not be optimal for the unimodal variants, and adjusting the architecture for each modality independently would enable a fairer comparison. Furthermore, using different modalities for detection and localization independently, for instance, vibration for detection and audio for localization, could improve overall system performance. Decision level fusion, where separate unimodal models produce independent reconstruction errors that are then combined by a dedicated fusion strategy, should also be explored as an alternative to feature level fusion.

In the longer term, deploying the system in a real hydropower or industrial setting would be the ultimate validation of the approach. This introduces additional challenges, including the collection of sufficient normal data across varying operating regimes such as steady state operation and transition phases between pumping and generation modes. The identification of suitable mounting positions within the machine room, where sensor deployment is physically feasible and where vibration can be picked up in a meaningful manner, presents a further open question. Optimizing sensor placement for spatial localization coverage and investigating the feasibility of real time implementation would also need to be addressed before the system could be considered for operational use.

Bibliography

[1] Swiss Federal Office of Energy SFOE. *Hydropower*. https://www.bfe.admin.ch/bfe/en/home/supply/renewable-energy/hydropower.html/. 2025.

[2] Astrid Björnsen et al. „Rethinking Pumped Storage Hydropower in the European Alps: A Call for New Integrated Assessment Tools to Support the Energy Transition“. In: *Mountain Research and Development* 36 (May 2016), pp. 222–232.

[3] Juan I Pérez-Díaz et al. „Trends and challenges in the operation of pumped-storage hydropower plants“. In: *Renewable and Sustainable Energy Reviews* 44 (2015), pp. 767–784.

[4] illwerke vkw AG. *Rodundwerk II*. https://www.illwerkevkw.at/rodundwerk-ii. 2025.

[5] Karim Khamaisi et al. *From Noise to Knowledge: A Comparative Study of Acoustic Anomaly Detection Models in Pumped-storage Hydropower Plants*. 2025. URL: https://arxiv.org/abs/2509.22881.

[6] R. Keith Mobley. *An Introduction to Predictive Maintenance*. 2nd ed. 2002.

[7] Siemens AG. *The True Cost of Downtime 2024*. Tech. rep. Siemens AG Digital Industries, 2024.

[8] Youde Liu et al. „Anomalous Sound Detection Using Spectral-Temporal Information Fusion“. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 816–820.

[9] Hejing Zhang et al. *Anomalous Sound Detection Using Self-Attention-Based Frequency Pattern Analysis of Machine Sounds*. 2023. URL: https://arxiv.org/abs/2308.14063.

[10] Dewei Kong et al. „ASDNet: An Efficient Self-Supervised Convolutional Network for Anomalous Sound Detection“. In: *Applied Sciences* 15.2 (2025).

[11] Bo Peng et al. „Acoustic-Based Industrial Diagnostics: A Scalable Noise-Robust Multiclass Framework for Anomaly Detection“. In: *Processes* 13.2 (2025).

[12] Younjeong Lee et al. „LSTM-Autoencoder Based Anomaly Detection Using Vibration Data of Wind Turbines“. In: *Sensors* 24.9 (2024).

[13] Luca Radicioni, Francesco Morgan Bono, and Simone Cinquemani. „Vibration-Based Anomaly Detection in Industrial Machines: A Comparison of Autoencoders and Latent Spaces“. In: *Machines* 13.2 (2025).

[14] Fataneh Dabaghi-Zarandi et al. „Convolutional Variational Autoencoder for Anomaly Detection in On-Load Tap Changers“. In: *IEEE Access* 13 (2025), pp. 50838–50848.

[15] Sara Khan, Mehmed Yüksel, and Frank Kirchner. *Robust Anomaly Detection through Multi-Modal Autoencoder Fusion for Small Vehicle Damage Detection*. 2025. URL: https://arxiv.org/abs/2509.01719.

[16] S. S. Stevens, J. Volkmann, and E. B. Newman. „A scale for the measurement of the psychological magnitude pitch“. In: *Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190.

[17] S. B. Davis and P. Mermelstein. „Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), pp. 357–366.

[18] Shawn Hershey et al. „CNN Architectures for Large-Scale Audio Classification“. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.

[19] A. Grossmann and J. Morlet. „Decomposition of Hardy functions into square integrable wavelets of constant shape“. In: *SIAM Journal on Mathematical Analysis* 15.4 (1984), pp. 723–736.

[20] P. Goupillaud, A. Grossmann, and J. Morlet. „Cycle-octave and related transforms in seismic signal analysis“. In: *Geoexploration* 23.1 (1984), pp. 85–102.

[21] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. „Learning representations by back-propagating errors“. In: *Nature* 323.6088 (1986), pp. 533–536.

[22] Mayu Sakurada and Takehisa Yairi. „Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction“. In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. MLSDA'14. 2014, pp. 4–11.

[23] Charles H. Knapp and G. Clifford Carter. „The Generalized Correlation Method for Estimation of Time Delay“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.4 (1976), pp. 320–327.

[24] Lawrence E. Kinsler et al. *Fundamentals of Acoustics*. 4th. 2000.

[25] Karl F. Graff. *Wave Motion in Elastic Solids*. 1991.

[26] Stephanie Holly et al. *Autoencoder based Anomaly Detection and Explained Fault Localization in Industrial Cooling Systems*. 2022. URL: https://arxiv.org/abs/2210.08011.

[27] Xuyang Li et al. „Mechanics-informed autoencoder enables automated detection and localization of unforeseen structural damage“. In: *Nature Communications* 15 (2024), p. 9229.

[28] Jan Blumenthal et al. „Weighted Centroid Localization in Zigbee-based Sensor Networks“. In: *IEEE International Symposium on Intelligent Signal Processing*. 2007, pp. 1–6.

[29] Donald Shepard. „A Two-Dimensional Interpolation Function for Irregularly-Spaced Data“. In: *Proceedings of the 1968 23rd ACM National Conference*. 1968, pp. 517–524.

[30] D. Mills et al. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905. Internet Engineering Task Force, June 2010.

[31] David L. Mills. „Internet time synchronization: the network time protocol“. In: *IEEE Transactions on Communications* 39.10 (1991), pp. 1482–1493.

[32] NTP Pool Project. *NTP Pool Project*. 2003. URL: https://www.ntppool.org.

[33] Andrew Banks and Rahul Gupta. *MQTT Version 3.1.1*. OASIS Standard. OASIS, Oct. 2014.

[34] Eclipse Foundation. *Eclipse Mosquitto: An Open Source MQTT Broker*. `https://mosquitto.org/`.

[35] I. Fette and A. Melnikov. *The WebSocket Protocol*. RFC 6455. Internet Engineering Task Force, Dec. 2011.

[36] Sheng Chen et al. „MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices“. In: *Proceedings of the Chinese Conference on Biometric Recognition (CCBR)*. 2018, pp. 428–438.

[37] Harsh Purohit et al. „MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection“. In: *Proceedings of the 4th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*. 2019.

[38] Yuxiao Wang et al. „HybridCube: Integrating Multi-Sensor Cross-Modal Data for Early Fault Detection in Power Transformers“. In: *2024 IEEE International Conference on e-Business Engineering (ICEBE)*. 2024, pp. 250–255.

[39] Shuaicheng Zhang et al. „MentorPDM: Learning Data-Driven Curriculum for Multi-Modal Predictive Maintenance“. In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*. KDD '25. 2025, 2837–2847.

[40] Christian Lessmeier et al. „Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification“. In: *European Conference of the Prognostics and Health Management Society*. 2016.

[41] Mostafijur Rahman et al. „MultiSenseNet: Multi-Modal Deep Learning for Machine Failure Risk Prediction“. In: *IEEE Access* 13 (2025), pp. 120404–120416.

[42] B. P. Welford. „Note on a method for calculating corrected sums of squares and products“. In: *Technometrics* 4.3 (1962), pp. 419–420.

[43] InvenSense. *INMP441 Omnidirectional Microphone with Bottom Port and I2S Digital Output*. `https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf`.

[44] InvenSense. *MPU-6000 Product Specification Revision 3.4*. `https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf`.

[45] Espressif Systems. *ESP32-S3 Series SoC*. `https://www.espressif.com/en/products/socs/esp32-s3`.

[46] Arduino. *Arduino IDE*. `https://docs.arduino.cc/software/ide/`.

[47] AWS open source. *FreeRTOS*. `https://www.freertos.org/`.

[48] Docker Inc. *Docker Compose*. `https://docs.docker.com/compose/`.

[49] F5, Inc. *Nginx*. `https://nginx.org/`.

[50] Brian McFee et al. „librosa: Audio and Music Signal Analysis in Python“. In: *Proceedings of the 14th Python in Science Conference*. 2015, pp. 18–25.

[51] Gregory Lee et al. „PyWavelets: A Python package for wavelet analysis“. In: *Journal of Open Source Software* 4.36 (2019), p. 1237.

[52] Tony F. Chan, Gene H. Golub, and Randall J. LeVeque. *Updating Formulae and a Pairwise Algorithm for Computing Sample Variances*. Technical Report STAN-CS-79-773. Stanford University, 1979.

[53] François Chollet et al. *Keras*. 2015.

[54] Martín Abadi, Ashish Agarwal, Paul Barham, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[55] Dominic Tobler. „Acoustic Anomaly Detection in Hydropower Plants“. Bachelor's thesis. University of Zurich, Apr. 2025.

[56] SEVEN MASTER AQUARIUM. *Aquarium Water Pump*. https://smaquarium.com/products/aquarium-water-pump-fish-tank-bottom-suction-pump-submersible-pump-for-fountain.

[57] Audacity Team. *Audacity: Free, open source, cross-platform audio software*. https://www.audacityteam.org. 2026.

Abbreviations

AE	Autoencoder
AM	Attention Mechanism
AUC	Area Under the Curve
BPF	Band-Pass Filtering
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Network
CVAE	Convolutional Variational Autoencoder
CWT	Continuous Wavelet Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DW	Distance Weighted
FFT	Fast Fourier Transform
FSST	Synchrosqueezing of Short-Time Fourier Transform
GNN	Graph Neural Networks
HHT	Hilbert-Huang Transform
HPF	High Pass Filter
IP	Interpolation
KNN	K-Nearest Neighbours
LMS	Log Mel Spectrogram
LPF	Low Pass Filter
LSTM	Long Short Term Memory
MFCCs	Mel Frequency Cepstral Coefficients
MFN	MobileFaceNet
MHSA	Multi-Head Self-Attention
MQTT	Message Queuing Telemetry Transport
NTP	Network Time Protocol
OC-SVM	One-Class Suport Vector Machine
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
SPWVD	Smoothed Pseudo Wigner-Ville Distribution
STFT	Short-Time Fourier Transform
TDOA	Time Difference of Arrival
TCL	Temporal Contrastive Learning
TF	Temporal Features
VAE	Variational Autoencoder
WPT	Wavelet Packet Transform

List of Figures

List of Tables

List of Listings

Appendix A

Localization Heatmaps

Each figure shows a 3×3 grid of source locations arranged to match the physical room layout (x increases left to right, y increases bottom to top). The red cross marks the true source position; the dashed red circle indicates a 1 unit ($20cm$) hit radius, and the diamond markers show device positions.

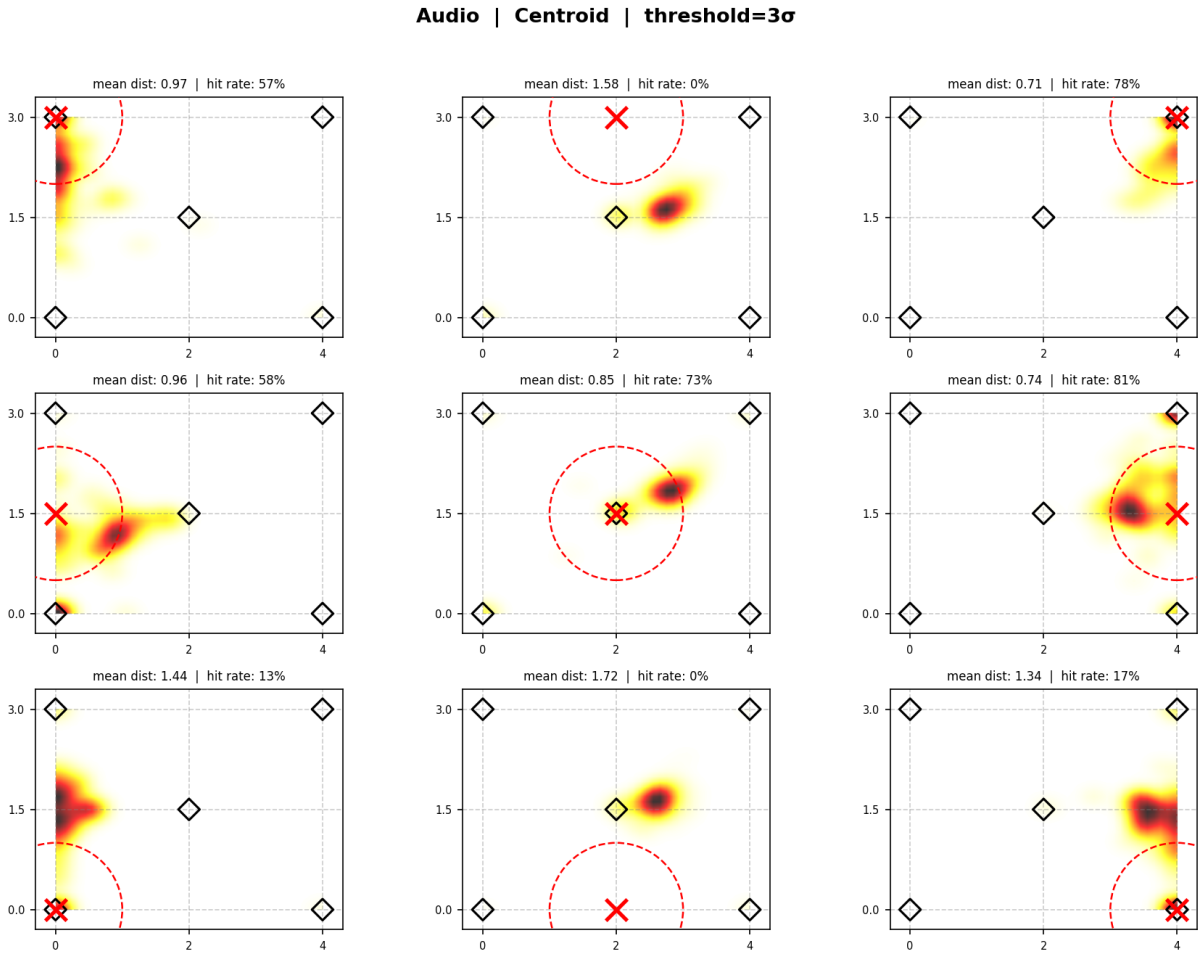

Figure A.1: Audio – Centroid

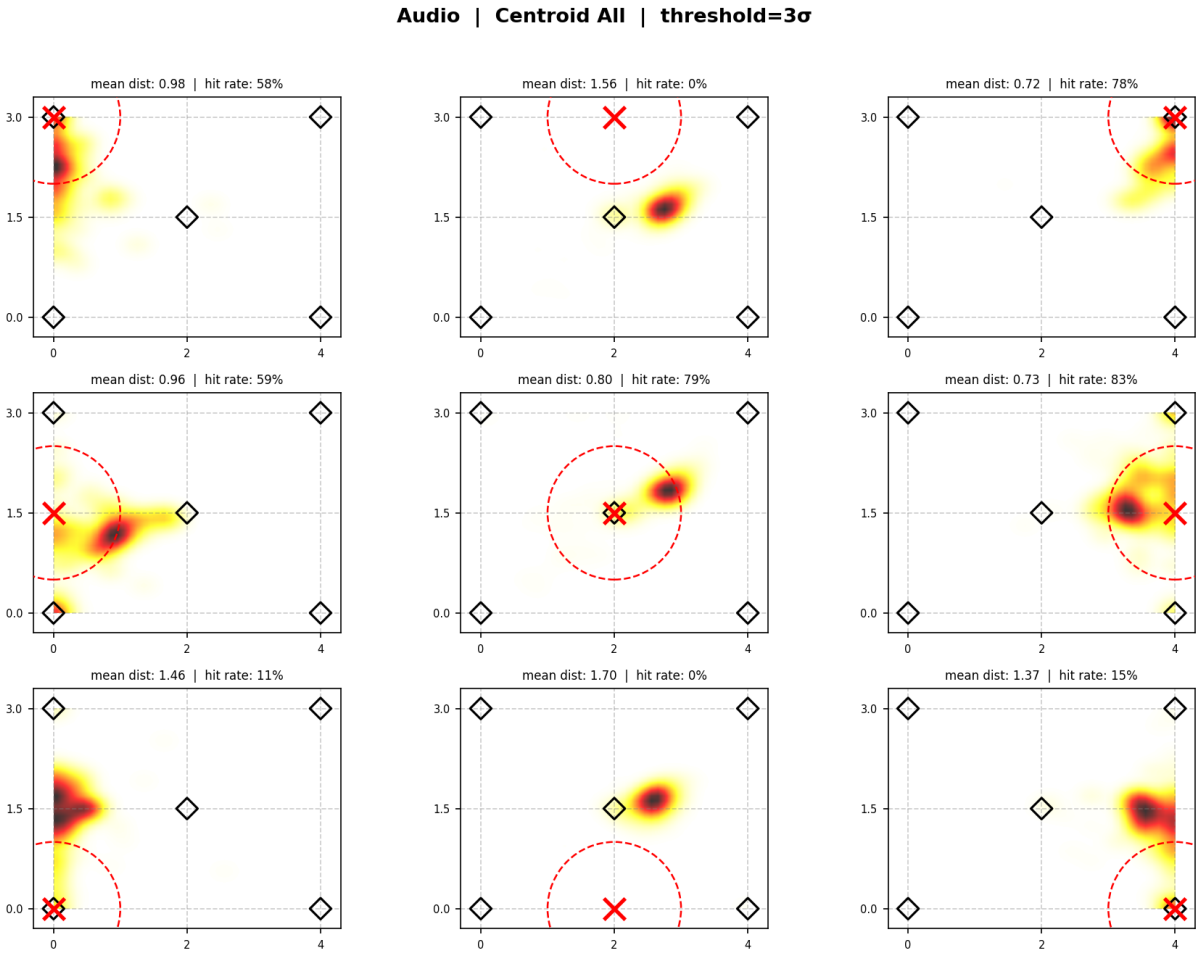

Figure A.2: Audio – Centroid All

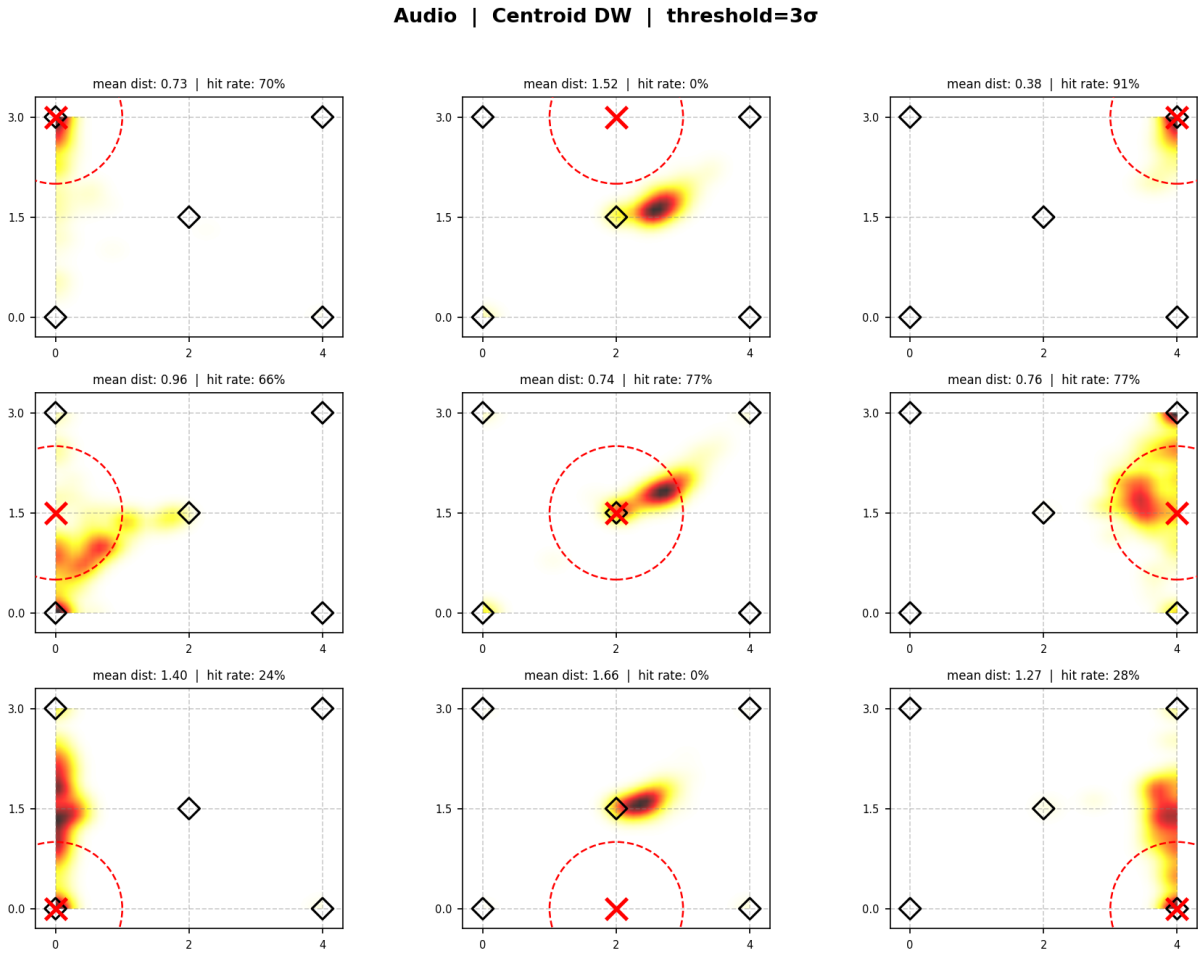

Figure A.3: Audio – Centroid DW

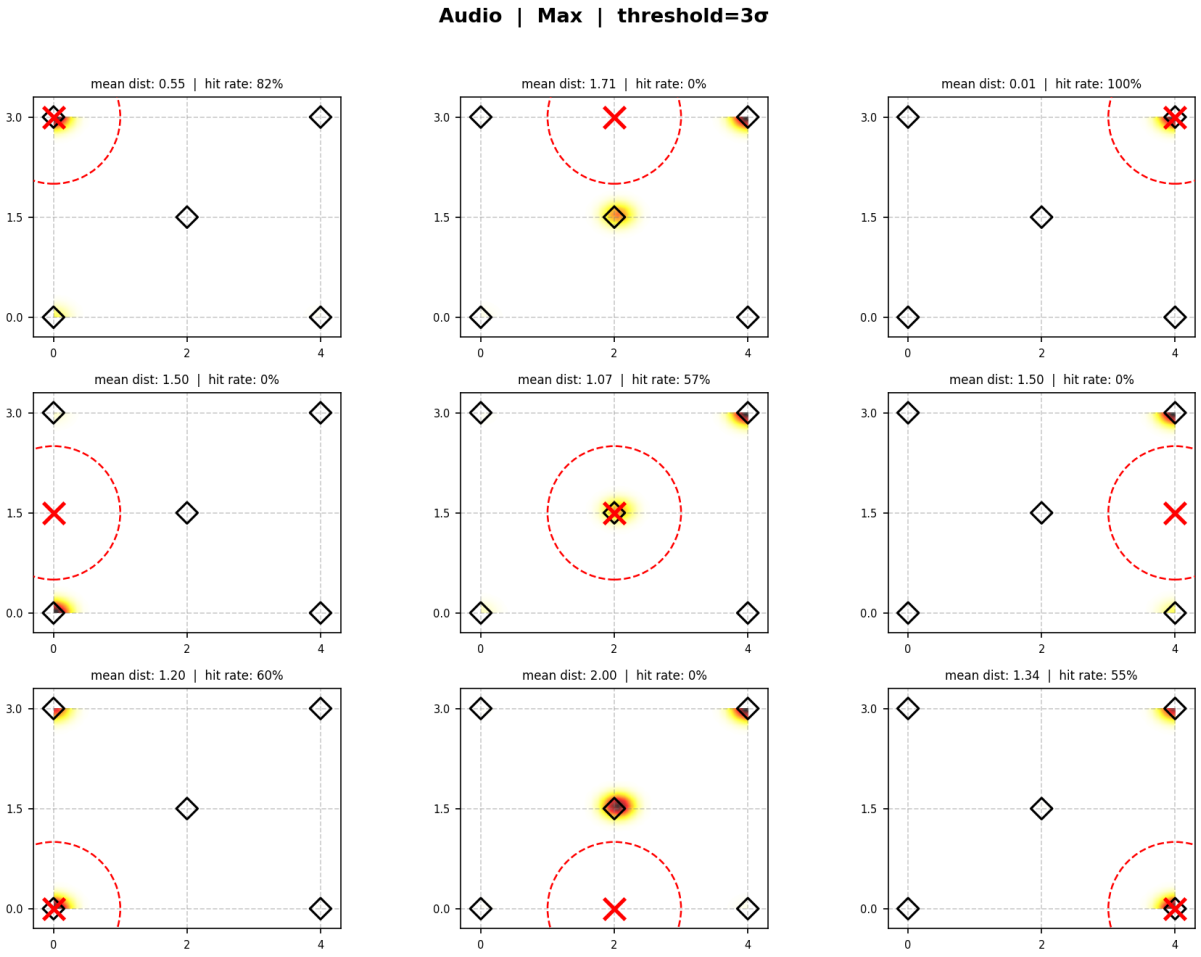

Figure A.4: Audio – Max

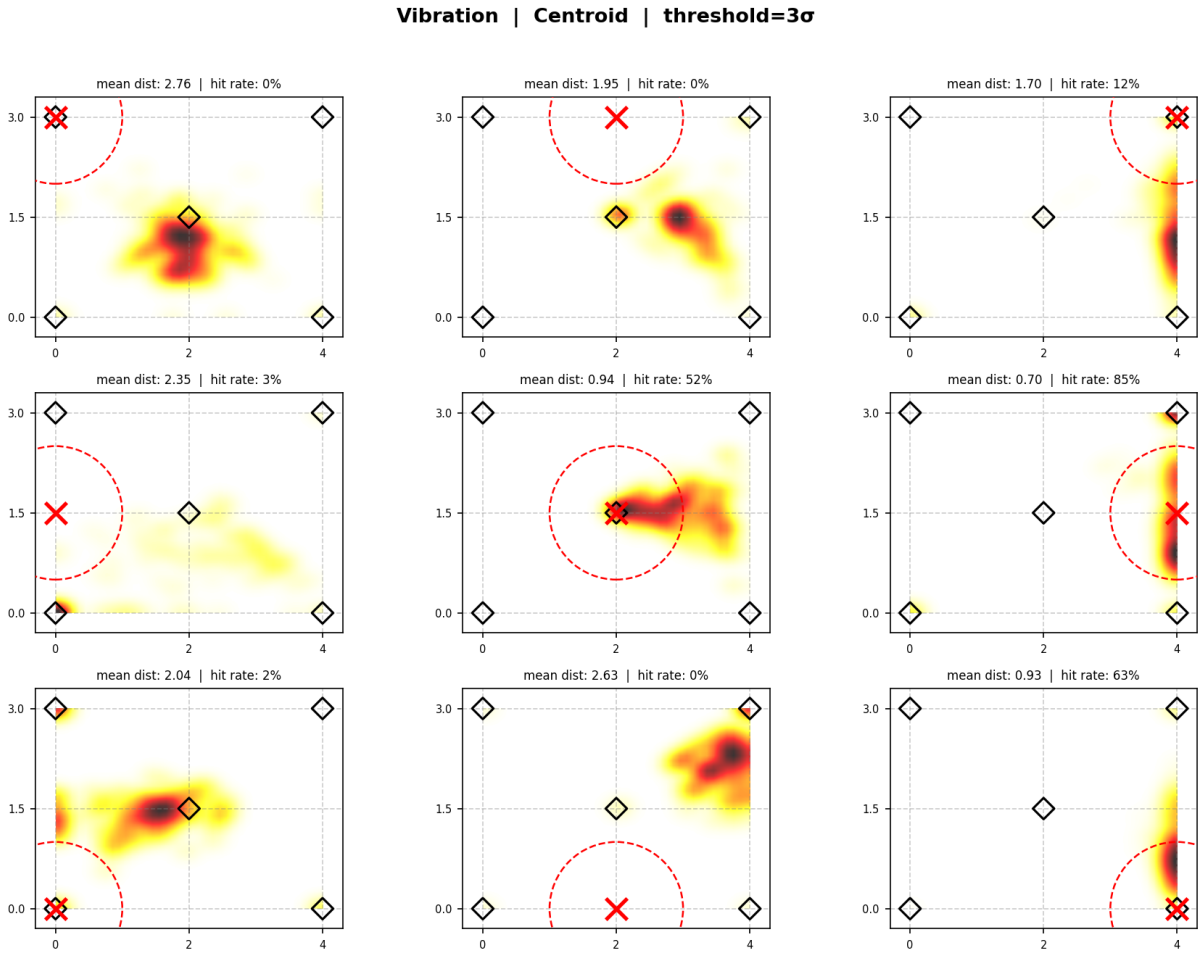

Figure A.5: Vibration – Centroid

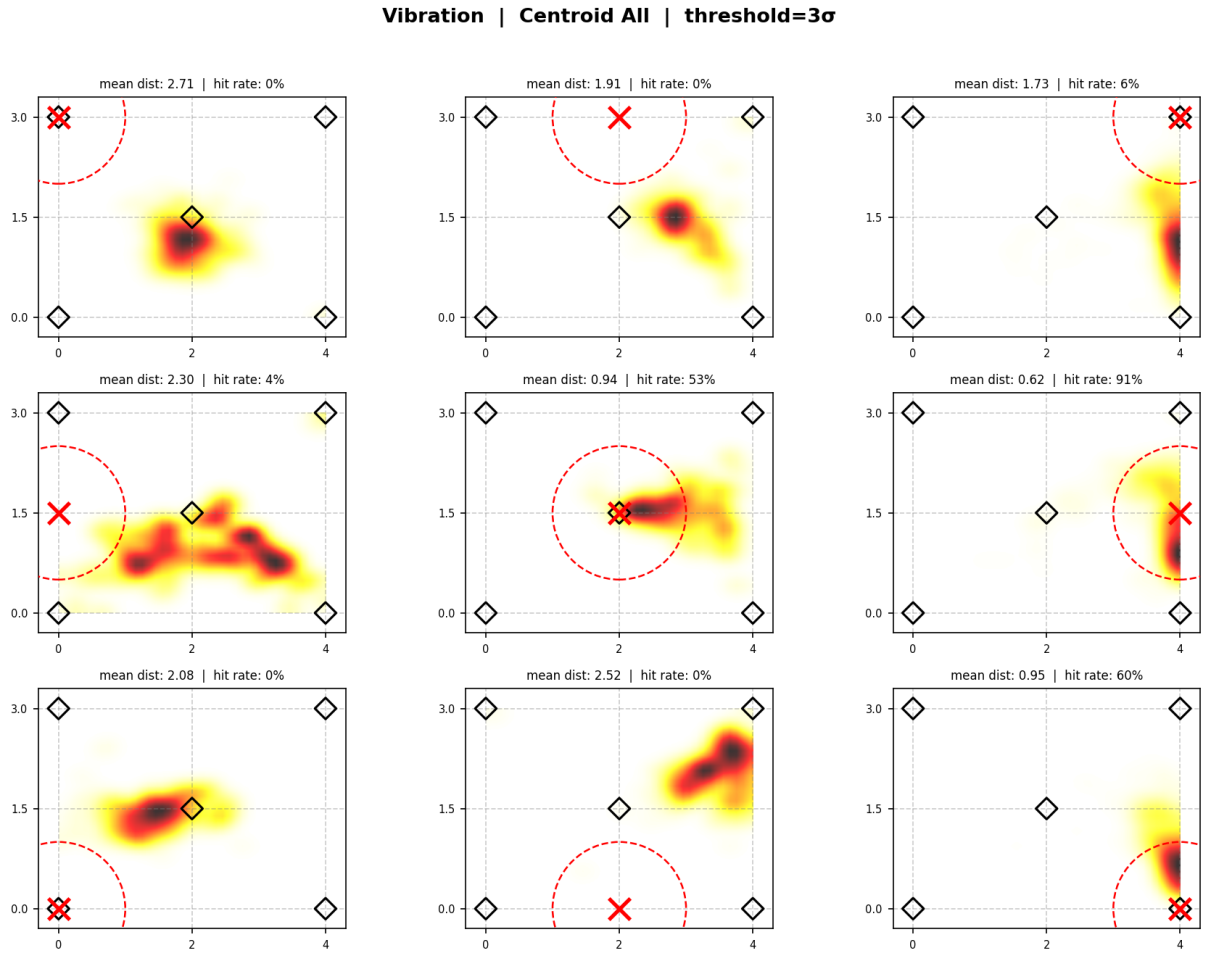

Figure A.6: Vibration – Centroid All

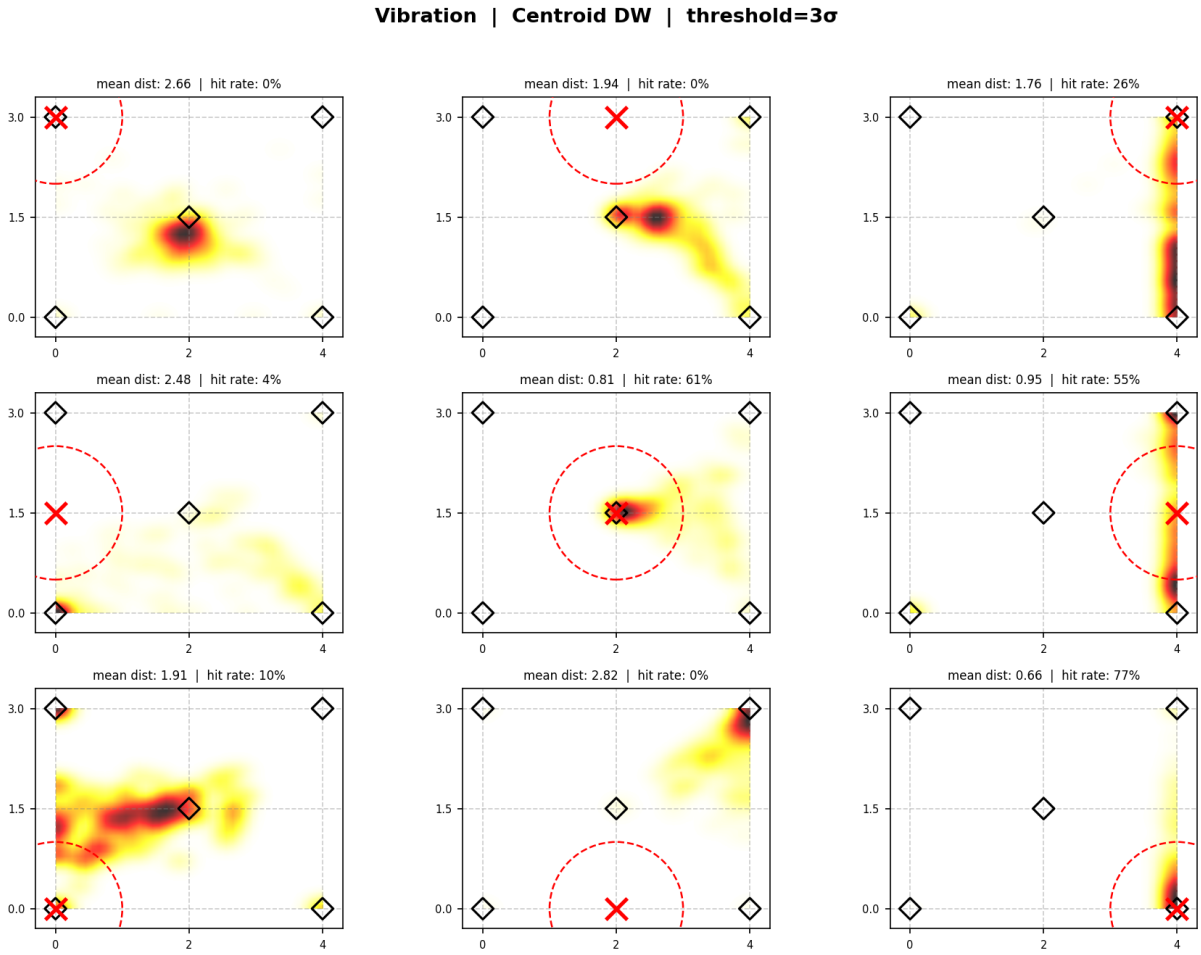

Figure A.7: Vibration – Centroid DW

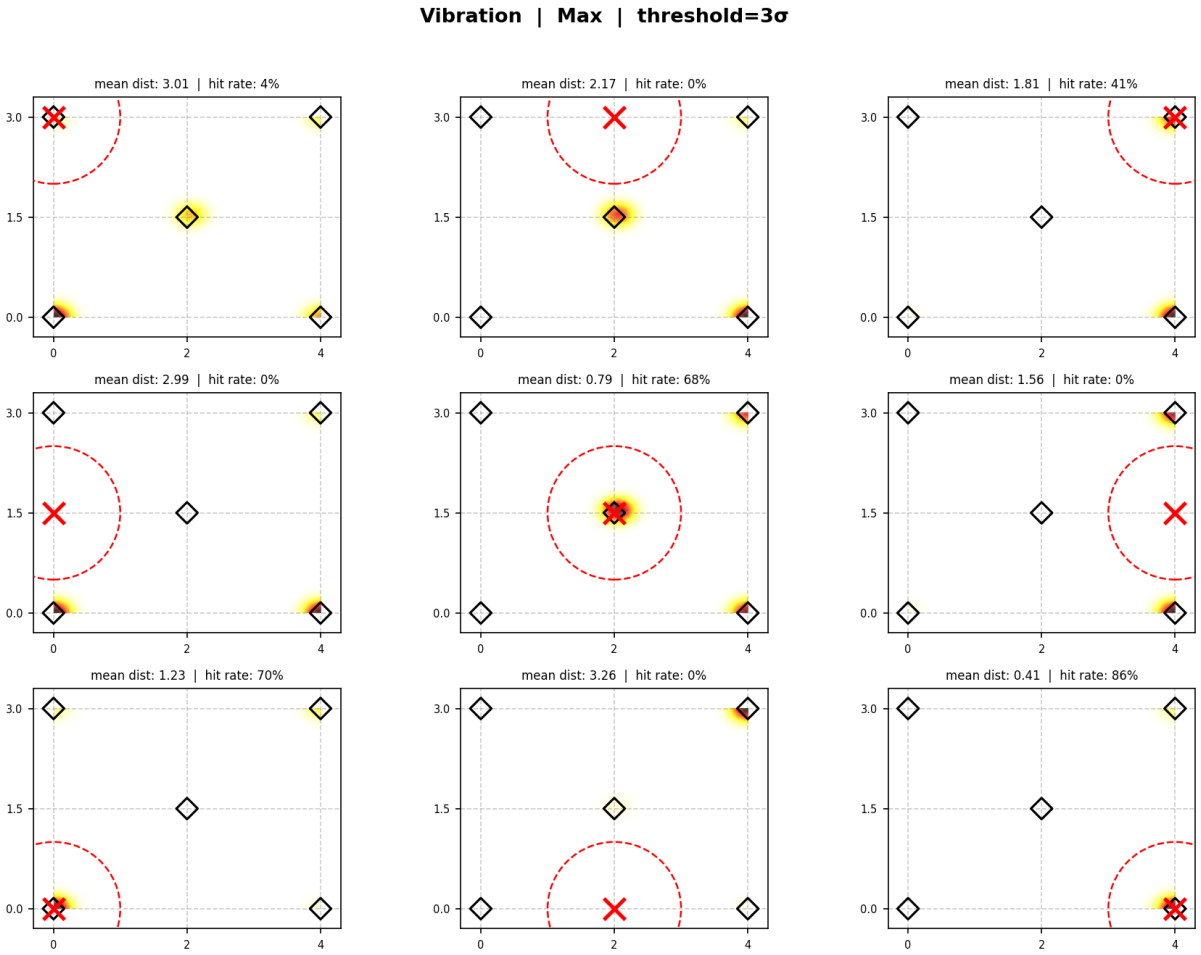

Figure A.8: Vibration – Max

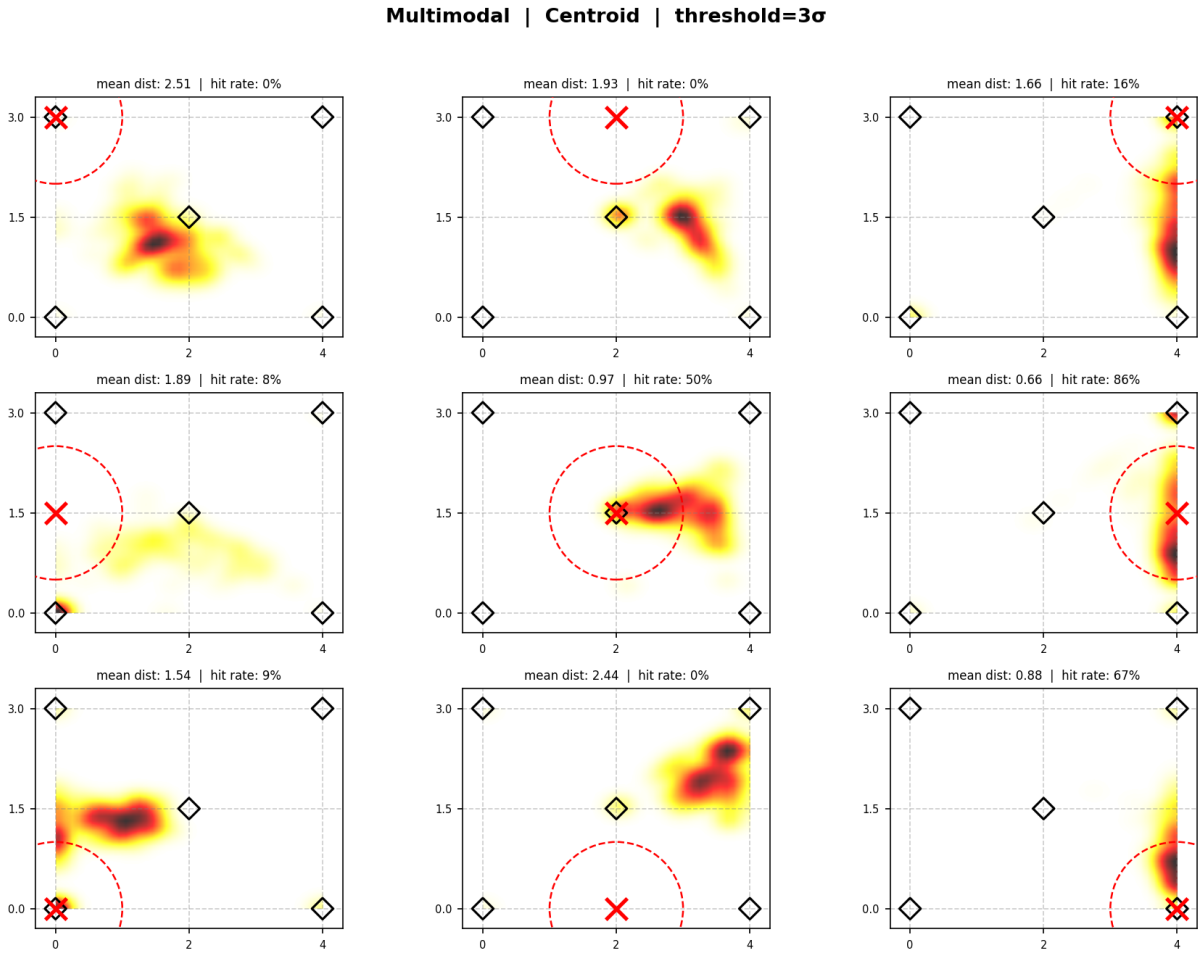

Figure A.9: Multimodal – Centroid

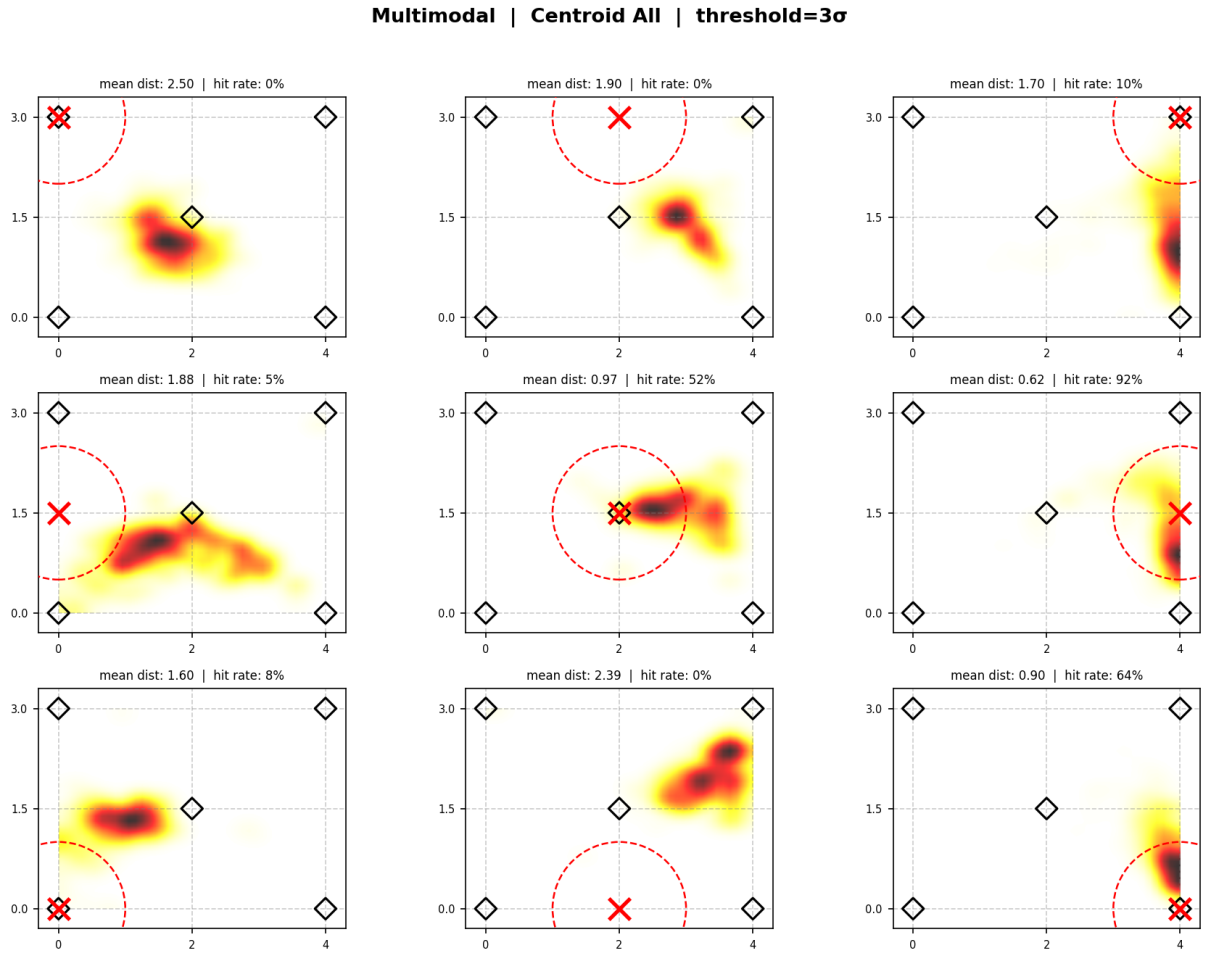

Figure A.10: Multimodal – Centroid All

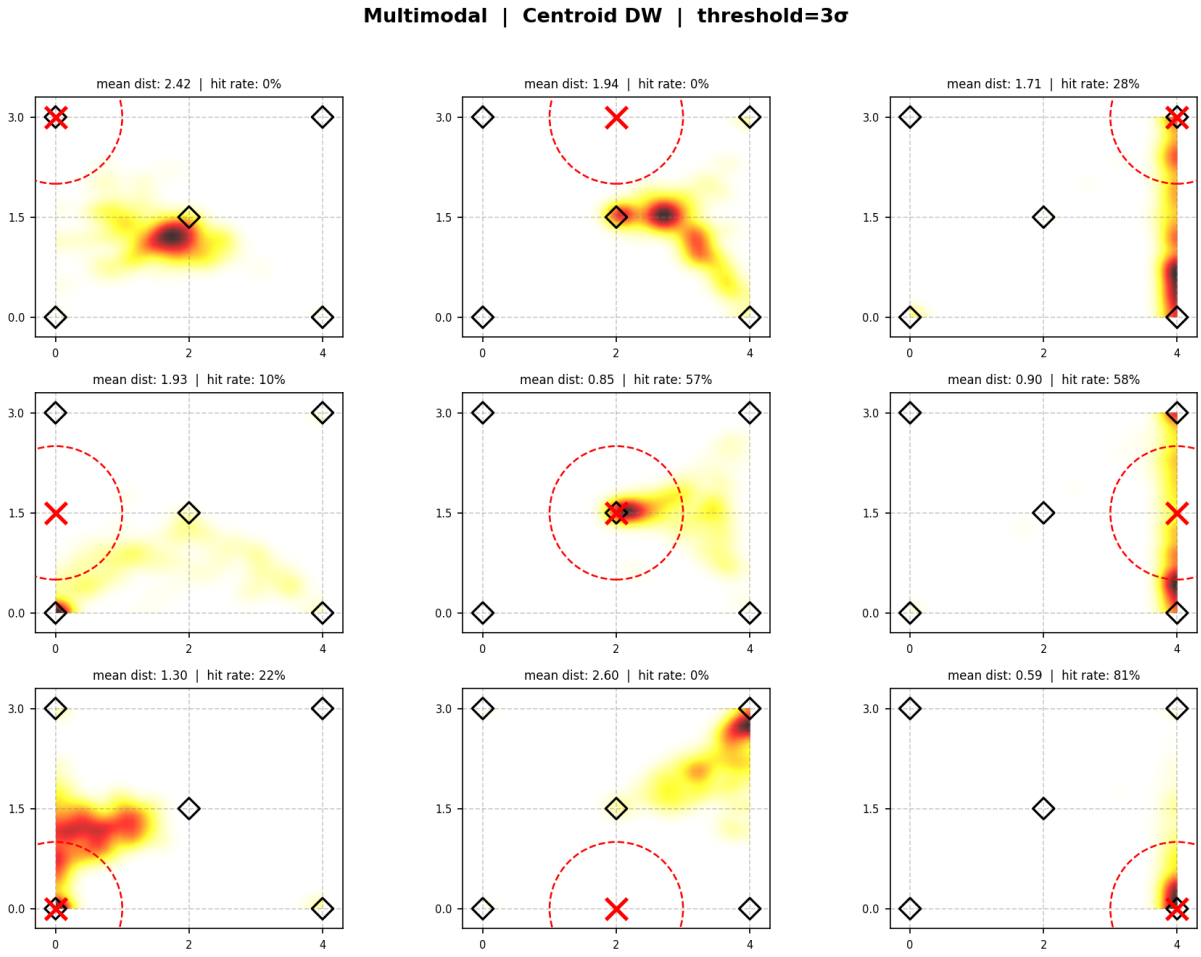

Figure A.11: Multimodal – Centroid DW

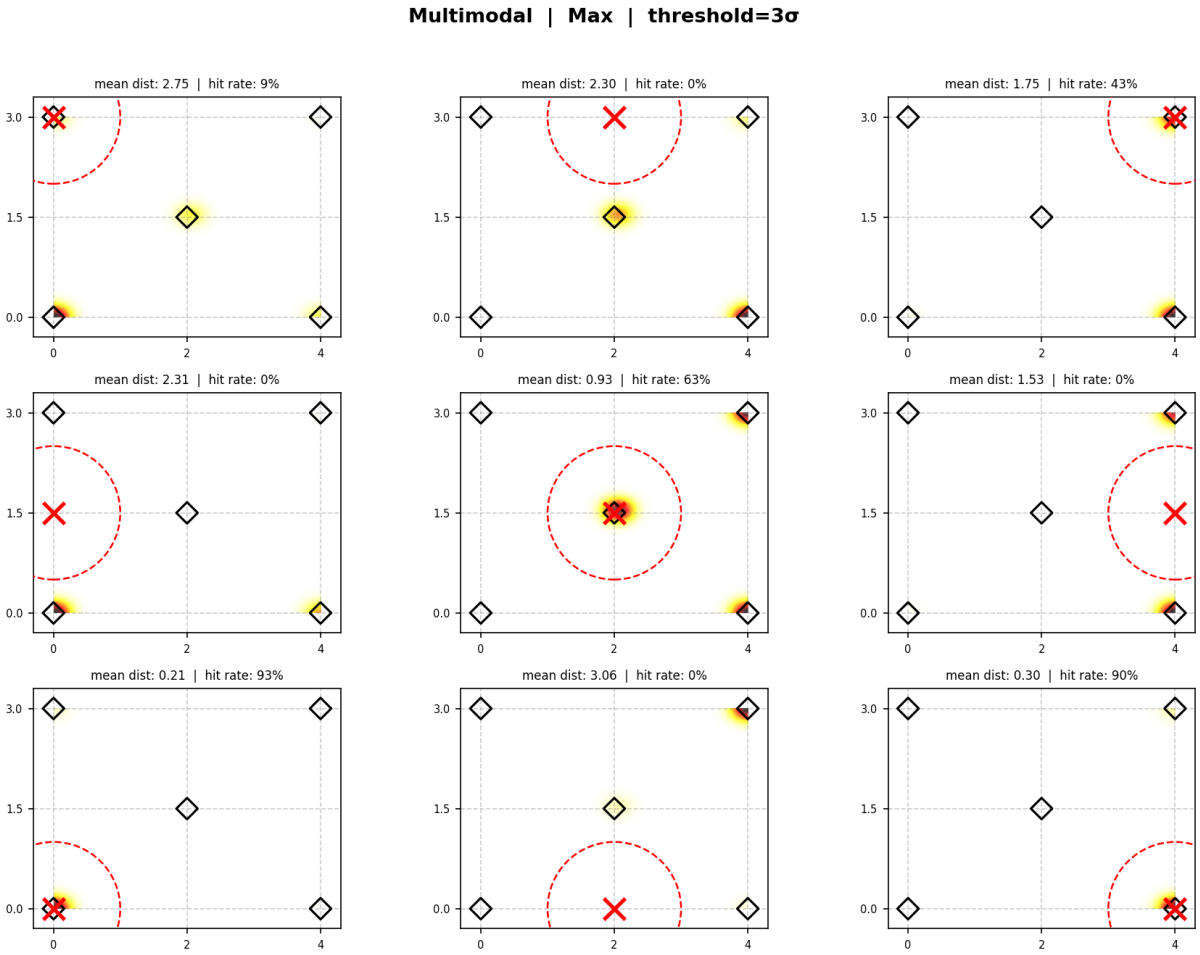

Figure A.12: Multimodal – Max