University of St. Gallen

School of Computer Science

to obtain the title of

Bachelor of Science in Computer Science

Bachelor Thesis on

Acoustic Sensing for Anomaly Detection

Spectral-Temporal Anomaly Detection for Remote, High-Altitude Hydro-Power Plants

---

Submitted by:

Nicolas Keller

22-610-422

Approved on the application by:

Prof. Dr. Bruno Rodrigues

Date of Submission:

May 19, 2025

Acknowledgements

I want to extend my sincere gratitude to my thesis advisor, Prof. Dr. Bruno Rodrigues for his constant support throughout the process of writing this thesis. The completion of this thesis was made possible, to a great extent, by his essential guidance, valuable feedback and his active role in steering me into the right direction.

I am indebted to Valentin Huber and Stefan Krummenacher for providing the dataset and for their foundational research on acoustic-based anomaly detection, which enabled me to make my thesis possible.

To my family especially, who despite the distance, supported and encouraged me in every step of the way during my studies. It would not have been possible without their cheering from afar. This achievement is as much yours as it is mine.

St. Gallen, May 19, 2025 Nicolas Keller

Abstract

In the context of industrial factories and energy producers, unplanned outages are costly and difficult to service. This thesis explores the application of acoustic sensing for anomaly detection in predictive maintenance (PdM) systems, with a focus on remote high-altitude hydropower plants. The thesis addresses the challenges of monitoring legacy industrial machinery in harsh environments by proposing a holistic hybrid framework that integrates spectral-temporal analysis with Machine Learning (ML). The framework consists of three complementary stages. (1) A rigorous exploratory data analysis (EDA) module that denoises, normalizes and segments audio recordings before extracting time- and frequency-domain insights (waveforms, FFT/STFT, Mel-spectrograms, MFCCs, wavelet coefficients, and high-level spectral statistics). (2) A hybrid model training of ML models on the datasets: K-means clustering, a One-Class SVM for tight boundary learning, and an LSTM autoencoder for temporal reconstruction errors. (3) Holistic and standardized evaluation of model performance, including ROC-AUC, precision, recall, F1, as well as train- and inference-time to expose accuracy–latency trade-offs.

The framework is benchmarked on three datasets: (i) washing machine run with induced anomalies, (ii) operating high-altitude pumped-storage plant with induced anomalies, and (iii) real anomalies captured in a high-altitude pumped-storage plant. The findings reveal that the OC-SVM outperformed other models with robust performance (ROC AUC: 0.9659–0.9976) and minimal training times. The LSTM AE also proved strong anomaly detection capabilities (ROC AUC: 0.8885–0.9974) but required higher computational resources and thus longer training times. Although K-means performed well under evident anomalies (ROC AUC: 0.9974), it failed with subtler anomalies in the washing machine dataset. The EDA proved to be a crucial phase of the framework, identifying irregularities in the recordings and thus enabling manual anomaly detection before model training.

The results underscore the viability of hybrid PdM systems using lightweight ML models in resource-constrained industrial settings. This work contributes to a scalable acoustic-based PdM framework while lowering barriers to cost-efficiency and practicality for high-altitude hydropower plants, through a robust and lightweight system that detects acoustic anomalies in industrial machinery.

Table of Contents

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
AE	Autoencoder
DCAE	Deep Convolutional Autoencoder
Conv-LSTMAE	Convolutional Long Short Term Memory Autoencoder
KDE	Kernel Density Estimation
OC-SVM	One-Class Support Vector Machine
GMM	Gaussian Mixture Model
B-GMM	Bayesian Gaussian Mixture Model
LOF	Local Outlier Factor
IF	Isolation Forest
PCA	Principal Component Analysis
STFT	Short-Time Fourier Transform
ASD	Anomalous sound detection
PdM	Predictive Maintenance
CPS	Cyber-Physical Systems
LMS	Least Mean Squares
MoGs	Mixture of Gaussians
AUC-ROC	Area Under the Curve - Receiver Operating Characteristic

List of Figures

List of Tables

Chapter 1

Introduction

In industrial sectors, such as manufacturing, transportation, and energy production, the efficiency and reliability of industrial machinery play a critical role in ensuring smooth operations. Thus, the maintenance of this machinery is a crucial activity in industrial environments [73]. Unexpected equipment failures can cause significant operational downtime and losses for companies [43]. In fact, ineffective maintenance management results in losses greater than $60 billion for the industrial sector each year in the US [67] and $129 million per facility for Fortune Global 500 companies [94]. Thus, companies strive to minimize unplanned downtime and avoid significant over-maintenance costs [108]. This is especially relevant for industrial sectors in which equipment is located in hard-to-access areas such as in high-altitude hydropower plants, where environmental factors such as extreme weather conditions and fluctuating load demands impact machinery performance, but also for facilities located in more modest terrain. Therefore, being able to determine the ideal time to perform maintenance is crucial for the industrial sector [19].

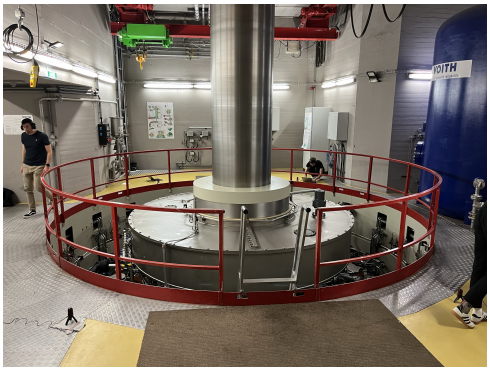


Figure 1.1: Hydropower Plant Rodundwerk II

Predictive maintenance (PdM) has emerged as a vital strategy to minimize downtime and enhance operational efficiency [3]. The common premise of PdM is that by constant

monitoring of mechanical conditions and operational efficiency, the systems will provide the necessary data to ensure anticipated minimal repair times and reduce the costs of unplanned outages. In reality though, PdM is much more than that; it is the means of improving product quality, productivity, and overall efficiency for large machinery [67]. In practice, however, PdM refers to the intelligent monitoring of industrial equipment to avoid future failures [74]. Using data-driven technologies such as Internet-of-Things (IoT) devices and methodologies, including acoustic sensing and machine learning (ML), PdM aims to detect early signs of equipment failure to enable proactive maintenance interventions before fatal breakdowns occur [14].

Traditional methods of monitoring the health of machines in industrial settings consist of measuring vibrations, temperature, and production performance. Unfortunately, contact-type sensors, such as accelerometers, proximity probes, pressure transducers, and thermometers cannot always be easily installed on all machines due to limitations such as space constraints, high costs, or sensor reliability issues. Alternatively, acoustic sensing proposes a new approach to monitoring the health of machines through sound. Acoustic sensors for monitoring machines lowers the barrier to entry for many businesses; they can be quickly and cheaply implemented, and do not require physical contact, which can be advantageous in various industrial scenarios considering that acoustic signals contain valuable insights into a machine's health state [51].

Acoustic sensing for anomaly detection is therefore seen as a promising way to automatically identify anomalous conditions in industrial machinery and equipment [53]. The detection of anomalies has received increasing attention lately due to the development and movement of the industry towards a more ubiquitous environment [101]. This approach of sensing anomalous sounds aims to detect unknown anomalous sounds by training models on normal sounds [39]. Anomalous sound detection (ASD) systems have focused on utilizing processed acoustic features—such as air pressure signals and spectrograms (often represented using the Mel scale)—to support the training of neural network models. These models frequently employ autoencoder architectures as the underlying framework [23, 65, 71]. Consequently, anomaly detection has been recognized as one of the leading techniques for accurate PdM [101] and automation of industrial equipment supervision [53].

1.1 Thesis Goals

The objective of this Bachelor's Thesis is to evaluate and integrate state-of-the-art approaches and methods for the spectral analysis of acoustic signals to detect anomalies. Although spectral sound analysis for anomaly detection is not a relatively new topic, its application in high-altitude legacy hydropower plants introduces challenges and opportunities. Thus, the main contributions are related to the real-world analysis of modern approaches, identifying and addressing these challenges by testing modern anomaly detection techniques [30, 31, 60, 75]. As such, we enhance the predictive maintenance capabilities and operational efficiency of older hydropower facilities, ensuring their long-term reliability and safety. The specific objectives of this thesis are as follows:

- To analyze the applicability of acoustic sensing to detect anomalies in hydropower plants.
- To investigate state-of-the-art machine learning techniques for processing and classifying acoustic signals.
- To develop a framework for integrating spectral analysis techniques into predictive maintenance models.
- To evaluate the proposed approach using real-world and simulated datasets from hydropower plants.

1.2 Methodology

This thesis follows an applied research methodology to design, implement, and evaluate an acoustic-based anomaly detection system for predictive maintenance. The work begins with a comprehensive overview of the fundamental concepts in spectral analysis including audio pre-processing techniques.

Following the fundamentals, the focus shifts to a literature review aimed at understanding the current state of PdM in industrial applications, with a focus on acoustic sensing. This involves both reviewing relevant studies on signal processing, spectral analysis, and machine learning for anomaly detection, as well as exploring existing state-of-the-art frameworks for anomaly detection using acoustic sensing systems.

The research utilizes acoustic data obtained from hydropower plant machinery as well as a dataset from a washing machine that includes both normal operations and anomalous sounds introduced artificially. Various preprocessing techniques are applied, such as

noise reduction, feature extraction, and conversion to spectrogram representations, to prepare the data for subsequent analysis.

Subsequently, selected machine learning models designed to detect anomalies in acoustic signals are developed and trained on the acoustic datasets.

Finally, the effectiveness of trained models is evaluated using metrics such as precision, recall, F1-score, and both training and inference time.

1.3 Thesis Outline

The remainder of this thesis is structured as follows.

- **Chapter 2: Fundamental** - This chapter covers the fundamental concepts related PdM and anomaly detection, along with a comprehensive review of existing solutions and their limitations.
- **Chapter 3: Methodology** - This chapter outlines the details of the research methodology, from the data acquisition process to model development.
- **Chapter 4: Implementation** - The practical implementation of the proposed anomaly detection system is presented and explained in detail.
- **Chapter 5: Results and Discussion** - This chapter presents the findings of the proposed anomaly detection system and discusses the results and their contribution to this thesis.
- **Chapter 6: Conclusions & Future Work** - The final chapter summarizes the findings, discusses implications, and suggests future research directions.

Chapter 2

Fundamentals

This chapter presents the fundamental theoretical concepts and established frameworks associated to anomaly detection for predictive maintenance. An overview of the most relevant techniques and principles for anomaly detection is given in Section 2.1, offering a foundational understanding of key concepts. Specifically, signal processing techniques are discussed in Section 2.1.2, while relevant machine learning models for anomaly detection are introduced in Section 2.1.3. Section 2.2 presents a comprehensive review of existing literature and methodologies applied in anomaly detection. A comparative analysis of various related works is provided in Section 2.2.1, discussing relevant insights and suitability concerns for this thesis. Finally, Section 2.2.2 analyzes the key insights derived from the literature review, identifying research gaps and outlining considerations that shape the methodology of this thesis.

2.1 Background

The topics presented in this section form the theoretical foundation for the development and implementation of an acoustic-based anomaly detection system.

2.1.1 Predictive Maintenance

PdM is a proactive approach to anticipate equipment failures and schedule maintenance to optimize the performance and lifespan of the equipment by constantly monitoring its health in real time [46, 84] all while reducing costs [112]. Unlike traditional preventive maintenance, PdM predicts equipment failures based on condition monitoring, which can range from infrared thermography and acoustic monitoring, to vibration analysis and oil analysis [15]. This is made possible through the integration of sensor technology, data acquisition systems and applied analytics based on artificial intelligence (AI) methods such as ML and signal processing [49]. The goal is to identify patterns and make

preventive decisions based on the predictions of the detected anomalies through the AI methods [6].

2.1.1.1 Motivation for Innovation in Predictive Maintenance

The adoption and innovation of PdM is driven mainly by several key motivations:

- **Reducing unplanned downtime:** The idea that PdM can generate scheduling actions for maintenance by forecasting failures before they happen to help mitigate disruptions [112]. Consequently, diminishing maintenance costs and reducing unplanned downtime losses, represents significant opportunities for businesses to improve their productivity and efficiency [109].

- **Cost efficiency:** On one side businesses can schedule maintenance work only when needed, avoiding unnecessary servicing and reducing labor and material costs [47, 91]. Over-maintaining equipment is a typical downfall of many industries and lead to unnecessary higher costs [89]. On the other side, PdM itself can be expensive. Most businesses do not have the resources or personnel to develop their own PdM systems and turn to third-party companies to analyze the data and generate reports for them [59]. This can become expensive in the long rung and might not even be necessary for smaller companies. Consequently, the need for lower entry barriers into this technology and the development of more effective and easier solutions motivates innovation in businesses [89].

- **Technological advancements:** The widespread use of AI and ML to analyze large amounts of data in real time to predict failures is bigger than ever and continues to grow and improve [8]. Traditional methods of prediction accuracy were far behind 78% accuracy metrics, whereas AI models have been able to provide prediction accuracy of 92% and more, which also leads to more confidence in these systems [2].

2.1.2 Signal Processing

Among various approaches to predictive maintenance, acoustic sensing stands out as a promising technique. Because of factors such as being non-intrusive, its cost-effectiveness, and ability to capture early-stage anomalies in machinery, acoustic sensing continuous to grow and innovate. Because acoustic sensing relies on complex raw acoustic data from equipments, it requires advanced processing techniques to extract meaningful

insights. This sets signal processing to be a fundamental part of PdM, as it enables the transformation of acoustic signals into interpretable data representations.

It involves the process of converting raw digital signals into interpretable graphs, numbers, and discrete-time signals for analysis or to improve specific features through algorithms and other techniques [36, 88]. It is an engineering discipline that studies how to capture, analyze, and transmit information using mathematical tools and numerical algorithms [26]. In this case, the focus is on transforming raw audio signals into meaningful representations that facilitate the identification of faults in industrial machinery. This section covers preprocessing steps to applied acoustic data as well as spectral analysis methods.

2.1.2.1 Preprocessing of Acoustic Data

As first step, audio preprocessing is the process of converting and enhancing audio datasets into a suitable format to extract relevant features and prepares the datasets for further analysis or to be fed into ML models [35] [90].

2.1.2.2 Noise Reduction

In audio recordings, noise is an inevitable component of industrial acoustic data that surges from the inherent environment of mechanical vibrations and sensor limitations [24]. Noise reduction techniques aim to isolate background disturbances from relevant features [83].

1. **Wavelet Denoising**: *Discrete Wavelet Transform (DWT)* aims to represent a discrete time series, *x(n)*, as a set of wavelet coefficients [87]. It decomposes a signal into approximation and detail coefficients, enabling the suppression of noise while retaining essential signal characteristics [13, 61]. The mathematical representation of DWT is:

$$X(j,k) = \sum_{n} x(n)\psi_{j,k}(n) \tag{2.1}$$

   where $\psi_{j,k}(n)$ represents the wavelet basis function at scale j and translation k. Wavelet transforms in which the wavelets are continuously sampled is referred to as a Continuous Wavelet Transform (CWT) [10].

2. **Adaptive Filtering**: Adaptive filters dynamically adjust their parameters to reduce noise while obtaining an uncorrupted desired signal [105]. For example, the Least

Mean Squares (LMS) algorithm updates filter coefficients iteratively to minimize the mean square error between output and desired signal [33]:

$$w(n+1) = w(n) + \mu e(n)x(n) \tag{2.2}$$

where μ is the learning rate, $e(n)$ is the error signal, and $x(n)$ is the input signal [41].

2.1.2.3 Normalization

Normalization treats the input equal regardless of its source and maps it to a given domain. It ensures consistency in amplitude variations across different recordings thus improving consistency and stability of feature extraction [95].

1. **Min-Max Scaling**: Normalizes a signal to a fixed range, typically [0,1] or [-1,1] [17]:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2.3}$$

2. **Gain Normalization**: Another alternative to normalize audio can be using the Root Mean Square and normalize the audio volume based on the average sound level of the signal. For example, calculating the average amplitude of a sound wave using its RMS and adjusting the gain as needed to get to a normalized audio target [81].

2.1.2.4 Segmentation and Framing

Audio segmentation is a preprocessing technique aimed at dividing audio signals into smaller frames (chunks) for better temporal analysis [86]. The role of audio segmentation is to divide audio samples in order to be further processed by the corresponding systems [102]. In this case for example, it allows the application of time-frequency analysis techniques such as Short-Time Fourier Transform (STFT) which requires smaller windows to capture spectral characteristics. In short, anomalies can be detected more effectively in smaller segmented signals as in longer signals where changes in frequency or amplitude can get lost.

1. **Fixed Windowing**:

   Is the process of dividing the audio signal into subsequences of equal length using a window function $w(n)$. Fixed windowing is simple and computationally efficient

but may split important segments which can lead to ambiguities or unwanted translations [7].

2. **Overlapping Frames**:

   To be continuous and capture transient anomalies, overlapping frames are generated by shifting the window with a predefined *hop size* [4, 97]. If F is the frame size and H is the hop size, the number of overlapping frames N_f can be computed as:

$$N_f = \frac{T - F}{H} + 1 \tag{2.4}$$

   where T is the total number of time steps.

2.1.2.5 Feature Extraction

Audio feature extraction is the process of finding insightful characteristics from audio signals to create compact and representative samples that can be efficiently processed by machines [85]. These features capture essential characteristics of the raw audio signal that is then fed to ML models [27].

1. **Mel-Frequency Cepstral Coefficients (MFCCs):** MFCCs represent a compact set of features that describe the short-term power spectrum of an audio signal. Their goal is to model human auditory perception by mapping the signal's energy into the *Mel-scale*, which is designed to reflect how humans perceive sound. [1, 21, 34]:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.5}$$

   To compute the MFCC, five steps need to be followed: Pre-emphasis, Framing and Windowing, DFT, Mel-Frequency filter bank, Logarithm and apply DCT [21].

2. **Spectral Features:** A Mel-spectogram transforms the frequencies of an audio file into the Mel scale. The Mel-scale is a perceptual scale that aims to approximate the nonlinear frequency response of the human ear [80]. It can be mathematically described as:

$$S(m,t) = \sum_{f} X(f,t) H_m(f) \tag{2.6}$$

where $S(m,t)$ is the Mel-spectrogram at time t, $X(f,t)$ is the frequency representation of the signal, and $H_m(f)$ are the Mel filters. This conversion represents the audio signal in a time-frequency plot [44] as can be seen in Figure 2.1.

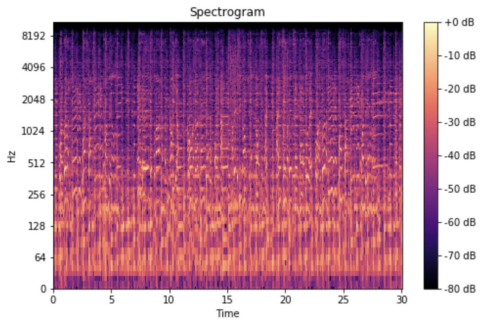


Figure 2.1: Mel-spectogram [80]

2.1.2.6 Spectral Analysis Methods

Having established the importance of pre-processing acoustic data, it is now explored how spectral analysis techniques can be applied to detect anomalies in hydropower systems. Spectral analysis has become a crucial part of the process for the identification and diagnosis of anomalies within acoustic signals [98]. It is a statistical method used to estimate how the total power of a finite record of a dataset is distributed over a frequency domain [52, 107]. By decomposing time-domain signals into their corresponding frequency components, spectral analysis facilitates the detection of anomalies indicating potential faults in stationary data sequences [107]. This section covers several prominent spectral analysis techniques relevant to the work.

1. **Fast Fourier Transform (FFT):** The FFT is a highly efficient computational algorithm to compute the Discrete Fourier Transform (DFT) of a sequence from a time domain to the frequency domain [16]. The algorithm has significant savings in computer time, having a complexity of $O(nlog2n)$ [12]. For spectral analysis, this transformation represents a significant part in the identification of the spectral content of signals, which is essential to diagnose anomalies in industrial equipment [9, 78]. The mathematical representation of the DFT is the following:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \tag{2.7}$$

where $x(n)$ represents the signal in the time domain, N is the number of samples and $X(k)$ is the frequency domain representation. FFT is widely used to analyze periodic components in acoustic signals [72].

2. **Short-Time Fourier Transform (STFT):** While the FFT represents a major milestone for spectral analysis, it provides only a global view of a signal's stationary frequency content and lacks temporal resolution; the short-time Fourier transform is introduced to overcome the problems of the FFT [38]. The STFT addresses this limitation by segmenting the signal into overlapping time windows and applying the FFT to each segment [63]. This approach yields a time-frequency representation, capturing how spectral content varies over time [111]. The STFT is defined as follows:

$$X(t,f) = \int_{-\infty}^{\infty} x(\tau) w(\tau - t) e^{-j2\pi f \tau} d\tau \tag{2.8}$$

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

3. **Wavelet Transform:** The wavelet transform is a technique used to decompose a signal into various scaled and translated versions of an oscillating wave-like function called wavelet [103]. Unlike the FFT and STFT, the wavelet transform provides a multi-resolution analysis of signals [111]. This allows us to analyze a signal locally at different frequencies, making it possible to detect both high- and low-frequency transients and identify anomalies that may not be visible after using FFT or STFT [103]. The continuous wavelet transform (CWT) is defined as follows:

$$W_x(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt \tag{2.9}$$

where $\psi(t)$ is the mother wavelet, a is the scale parameter, and b is the translation parameter [103].

2.1.3 Machine Learning Approaches for Anomaly Detection

2.1.3.1 Machine Learning (ML)

Signal processing techniques indeed play a significant role in the preparation of data for analysis. However, identifying anomalies requires more robust classification and pattern

recognition methods. Machine learning provides an effective framework for learning from data, detecting deviations, and improving PdM strategies [28]. This section explores key theoretical machine learning techniques relevant to anomaly detection in industrial acoustic systems.

ML is a broad field within AI that focuses on developing algorithms capable of learning patterns from data and making informed decisions [70]. These algorithms use statistical techniques to identify patterns in data and use those patterns for classification or prediction tasks [82]. ML aims to mimic human cognitive abilities by adapting to new information and improving predictions based on historical data [28]. In the context of acoustic anomaly detection, ML algorithms learn and analyze sound patterns to distinguish between normal and anomalous behaviors in the signals.

In recent years, ML has seen significant advancements and continues to evolve. ML types can be categorized into 3 big categories: supervised learning, unsupervised learning, and reinforcement learning. The focus of this thesis falls on unsupervised learning, as it offers the most suitable techniques for anomaly detection. In this context, unsupervised learning has the ability to identify acoustic deviations from normal patterns without requiring labeled datasets.

2.1.3.2 Supervised

Supervised machine learning algorithms generate a function that maps a given input to a desired output [70]. The term "supervised" stems from the need for labeled training data, where models learn from input-output pairs to make predictions on new data [76]. The learning process consists of two stages: training and testing. In the training phase, the model learns patterns from labeled examples, while in the testing phase it applies its learned knowledge to predict unseen cases [70]. Some popular supervised learning algorithms include decision trees, logistic regression, random forests, and neural networks [96]. Although they can be highly effective for classification and regression tasks, supervised learning is often impractical due to the difficulty of acquiring labeled datasets, especially in the context of anomaly detection.

2.1.3.3 Unsupervised

In contrast to supervised learning where algorithms learn from labels how to predict or classify things, unsupervised learning algorithms work on their own to discover the underlying structure of unlabeled data [45]. These methods are particularly useful in scenarios where labeled data is readily accessible, making it more suitable for various

scenarios [69]. For instance, they are well suited for anomaly detection because they do not rely on predefined normal or abnormal states but instead discover underlying structures and deviations within the dataset.

In the context of acoustic anomaly detection, unsupervised learning methods analyze the statistical properties of the sound signals to establish a base model of normal behavior [32]. Anomalies are detected when new data or sound signals diverge considerably from this acquired distribution [18]. Common techniques include clustering methods (*e.g.*, K-means, DBSCAN), dimensionality reduction techniques (*e.g.,* principal component analysis), and deep learning models such as autoencoders [69]. The capacity to identify unforeseen patterns without explicit supervision renders unsupervised learning profoundly pertinent to real-world industrial monitoring scenarios [32, 62].

2.1.3.4 Reinforcement Learning

Reinforcement learning (RL) is recognized as the third paradigm of machine learning, where agents learn to make decisions by interacting with their environment in a closed-loop feedback system [93, 100]. In essence, it simulates the human trial-and-error learning process to achieve specific goals. Unlike supervised learning, which relies on fixed inputs and labels, RL makes decisions sequentially, with each output depending on the current state of the input [100]. Although RL is a rapidly evolving field with diverse applications in robotics, finance, healthcare, and decision-making, it is beyond the scope of this thesis [64].

2.1.4 Overview of Anomaly Detection Algorithms

ML-based anomaly detection leverages the wide range of algorithms to identify anomalies in acoustic data patterns. This subsection provides an overview of the most common techniques applicable to anomaly detection in PdM.

2.1.4.1 Autoencoders

Introduced in 1986, autoencoders are a type of neural network with the intention of learning informative representations of data in an unsupervised manner by learning to reconstruct a set of input observations with the lowest error possible [66]. They consist of an encoder and a decoder. The encoder compresses input data into lower dimensional representations and the decoder reconstructs the original input from this representation [68, 99].

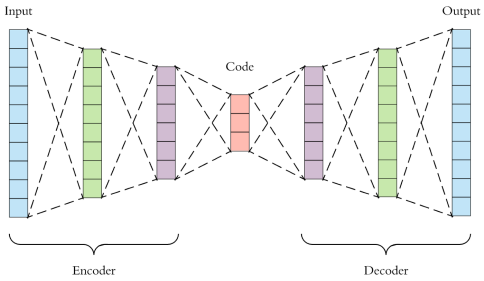


Figure 2.2: General structure of an autoencoder [77]

The general architecture of an autoencoder can be seen in Figure 2.2, where the encoder and decoder are usually neural networks, since they can be easily trained with existing libraries on Python [66].

In the context of this work, it is important to note how autoencoders can detect anomalies. As mentioned above, autoencoders learn to reconstruct normal data patterns with minimal reconstruction error; high reconstruction accuracy means the input could be reconstructed in the output with low error, but when an anomalous signal is encountered, the reconstruction error increases significantly, making it possible to detect anomalies [104].

Furthermore, there are several variations of AE architectures. For example LSTM autoencoders, leverage deep learning to capture and reconstruct the temporal dynamics in sequential acoustic data even outperforming traditional and convolutional autoencoders. In that sense LSTM autoencoders are capable of dealing with sequence as input, while regular autoencoders are not [29, 54].

2.1.4.2 Isolation Forest (IF)

Another unsupervised ML algorithm is IF, which is primarily used for anomaly detection [110]. Unlike traditional clustering algorithms, IF explicitly isolates anomalies by recursively partitioning datasets using decision trees [106]. The assumption is that anomalies take fewer splits (average path lengths) to be isolated compared to the rest of the points of the dataset.

The anomaly score is computed based on the average path length $h(x)$ of a data point x in the isolation trees [58]:

$$s(x,n) = 2^{-\frac{h(x)}{c(n)}} \tag{2.10}$$

where n is the number of samples, $h(x)$ is the average path length of x, and $c(n)$ is the average path length of an unsuccessful search in a binary tree:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{2.11}$$

with $H(i)$ being the harmonic number.

2.1.4.3 Principal Component Analysis (PCA)

PCA is a popular statistical technique used for reducing the dimensionality of dataset while preserving their variance [50]. PCA replaces p original variables by a smaller number, q, of derived variables, the principal components, that describe the dataset and are ordered by the amount of variance they explain [20, 50]. After reducing the dimensionality of the dataset, PCA is useful to spot trends, patterns and outliers in the data [79]. In the context of audio signals, PCA can be helpful to detect anomalies as it can filter noise and highlight significant deviations from normal data.

2.1.4.4 K-means Clustering

K-means clustering is an unsupervised algorithm used to partition data into K different clusters. It aims to minimize the intra-cluster variance [56]:

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

where J is the total intra-cluster variance, K denotes the number of clusters, x_i is a data point in cluster C_k, and μ_k is the centroid (mean) of the points in C_k.

In anomaly detection, K-means is applied to features derived from the audio dataset, such as spectral features, to represent normal operating conditions. To create clusters, the algorithm groups similar features together and creates centroids that categorize the normal operation behavior (*i.e.,* normal acoustic signals). Thus, anomalies are classified as data points that fall significantly far from the nearest centroid. This technique serves as a baseline for quick assessment of anomalies within a dataset [5].

2.1.4.5 One-Class Support Vector Machine (OC-SVM)

A one-class SVM is designed to learn the decision boundary that best encapsulates the normal data, which means that it aims to use a maximum-margin hyperplane that separates all data from the origin [18, 32]. For anomaly detection, the model is exclusively trained

on features extracted from normal audio samples, thus capturing the decision boundary that represents non-anomalous behavior. If new data points are outside of the learned boundary, they would be classified as anomalies [55].

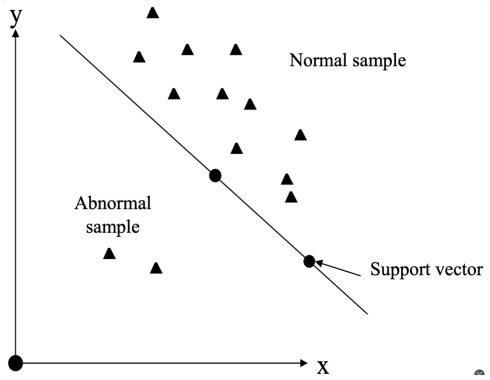


Figure 2.3: OC-SVM Diagram [42]

Each of these algorithms presents a different methodology for analyzing data, and detecting anomalies from acoustic signals. While autoencoders leverage deep learning for feature extraction, IF efficiently identifies outliers by isolating anomalies using decision trees. K-means and OC-SVMs are computationally inexpensive compared to larger deep learning models, yet offering high accuracy due to the simple nature of their computations. Deciding on which approach to take is contingent upon multiple factors such as dataset characteristics, the manner in which the data has been preprocessed, computational limitations, and the prevalence of actual anomalies in a real-world dataset.

2.2 Related Work

This section reviews existing research and methodologies regarding acoustic-based anomaly detection in industrial settings. It examines various approaches to identify the most effective strategies for anomaly prediction systems while highlighting gaps and areas that require further investigation in this thesis.

As seen in the previous sections, advancements in ML translate automatically into inherent improvements in anomaly detection systems. However, currently no single algorithm consistently outperforms others across all scenarios. This represents both a challenge for industries, as there is no standardized approach that companies can universally apply to detect acoustic anomalies, but gives the opportunity to combine models for hybrid approaches. The large number of available algorithms requires a

comparative analysis of related works to determine the most suitable approach for this specific scenario.

2.2.1 Review of Existing Systems

The below listed Table 2.1 presents a comparison of the ten most relevant studies for this thesis. The table focuses on the most critical aspects of the anomaly detection process: the type of data used, the methods applied for feature extraction, the machine learning model trained to predict anomalies, and the primary objective of each study.

This comparative analysis aims to provide a better understanding of which methods have been the most successful and where challenges persist. By examining these key dimensions, the table helps identify trends, limitations, and potential areas for improvement, ultimately guiding the selection of the most suitable approaches for acoustic anomaly detection in industrial settings, but also specifically for the given scenario of a high-altitude hydropower plant that requires simple and fast deployment while ensuring high accuracy anomaly detections.

Table 2.1: Comparison of Current Research

Work	Dataset	Feature Extraction	ML Model	Target
[57]	Real-world (factory)	Mel-spectrogram, Pre-trained CNNs	IF, GMM, B-GMM, OC-SVM, KDE, DCAE	Machine health monitoring for fans, pumps, valves, and slide rails in industrial environments.
[65]	Real-world (industrial)	MFCC, STFT, Mel-spectogram	OC-SVM, AE	Comparison of AE architectures to find the best suited to be implemented on hardware for real-time applications
[11]	Real-world (industrial)	Mel-spectogram	Conv-LSTMAE, CAE	Real time detection of acoustic anomalies in industrial processes using sequential autoencoders (explosions, fire, glass breaking).
[43]	Real-world & Synthetic	Mel-spectrogram, STFT	AE, LOF, IF, PCA	Evaluate anomalies in both synthetic and real-world datasets by leveraging a combination of signal processing techniques
[37]	Real-world (insutrial)	AKF, ASEF	Adaptive Order-Tracking	Implemented Acoustic Sensing System for Online Early Fault Detection in rotating machinery (industrial fans).
[32]	Real-world (industrial)	Mel-spectrogram	LSTM-AE, CNN-AE	Implemented LSTM and CNN based AEs to process continuous streams of audio signals from different audio sources in real-world factories based on window sliding.
[18]	Real-world & Synthetic (industrial & in-vehicle)	MFECs, Mel-spectogram	Dense AE, CNN-AE, LSTM-AE	Compared performances of three proposed AE architectures using real and synthetic data in three different experiments
[5]	Real-world (vehicles)	Not specified	SVM, K-means, KNN, CNN	Mounted multiple microphones on a vehicle to collect acoustic data to detect abnormal operations and improve general customer satisfaction
[25]	Real-world (industrial)	Mel-spectogram	CAE	Detect abnormal event in industrial plants for the new generation of factories.
[62]	Google Data Set	ICA	AE	Introduced ICA and AE for the detection of anomalous sound in industrial environments with Google's dataset.

In general, most studies have recorded their own real-world audio datasets, primarily in industrial environments with machinery sounds during operation. In [5], multiple microphones are deployed simultaneously to better capture vehicle sounds, allowing the detection of deviations from normal acoustic patterns.

An interesting example is found in [43], which utilizes three datasets: a real-world dataset, a synthetic one (with induced anomalies) from a hydropower plant and a third dataset from a washing machine. This thesis builds on that paper, as the same dataset is used in this thesis. The advantage of using synthetic audio datasets lies in their utility for validation. Controlled environments, such as the washing machine setup, where anomalies can be deliberately induced, allow rigorous testing of anomaly detection algorithms to evaluate their accuracy. However, most of the datasets are collected in industrial settings, highlighting the growing emphasis on applying machine learning techniques in real-world scenarios.

In terms of feature extraction, Mel-spectrograms are the most commonly used technique [11, 18, 25, 32, 43, 57, 65], followed by MFCCs and STFT [43, 65]. This demonstrates the importance and effectiveness of converting raw audio signals into meaningful inputs for machine learning models. Although feature extraction is widely employed, it appears to be somewhat undervalued. A more in-depth analysis of feature extraction techniques could lead to better data understanding and, consequently, improved anomaly detection performance.

The most used machine learning models across the reviewed works are AEs. This is logical given that AEs are highly adaptable to the specific audio signals of an environment, enabling them to effectively detect abnormal sounds that deviate from the learned baseline. These models are particularly suitable for capturing complex temporal and spectral patterns, making them effective for anomaly detection in noisy environments. Interestingly, various different AE architectures are seen in these works. For instance, [18] explores three different AE architectures, while [32] compares LSTM-AE and CNN-AE models. In contrast, [11] utilizes Conv-LSTMAE and CAE. This diversity in architectural choices highlights a limitation in terms of standardization, as no single model consistently outperforms others across all scenarios or applications.

There are two studies that stood out for their innovative methodologies:

- Huber (2023) [43]: This work employs a hybrid approach by combining AE, LOF, IF, and PCA. Although IF and PCA do not perform as well as expected, LOF and AE achieved near-perfect classification metrics on their synthetic dataset. This

demonstrates the potential of hybrid models to improve classification accuracy and robustness.

- Müller *et al.* (2020) [57]: This study stands out due to its approach of extracting features using neural networks pre-trained on image classification tasks, rather than relying solely on deep autoencoders. These features were then used to train a variety of anomaly detection models. The authors demonstrated that this approach outperformed conventional convolutional autoencoders, particularly in noisy environments with recordings from four different types of factory machines. This reinforces the importance of using hybrid approaches that combine multiple models and feature extraction methods to achieve better results.

Both works highlight the effectiveness of using ensemble methods and hybrid systems in anomaly detection, suggesting that integrating different models and extraction techniques can lead to more reliable and accurate results.

2.2.2 Research Gaps & Considerations

Despite the progress in the industry so far, some research gaps remain in the field. This thesis tries to address these pain points and will consider them in the methodology.

- **Scalability and Noise Resilience:** Some anomaly detection frameworks seem to be very specific to the research being carried out. This limits the scalability of the systems in other industries. In addition, noise-resilient systems could be further explored by implementing hybrid strategies that integrate reconstruction- and density-based models to build more robust anomaly detection frameworks.
- **Feature Extraction:** Not enough emphasis has been placed on manual data analysis and feature extraction of datasets. A more in-depth analysis of the datasets would allow one to gain deeper insights into the recordings, and thus absorb a better understanding of the anomalies.
- **Standardization and Benchmarking:** The lack of standardized architectures and benchmarking metrics makes it difficult to compare models directly. Establishing consistent standards would facilitate more meaningful evaluations and comparisons.

By addressing these gaps, future research can enhance the robustness, scalability, and generalization capabilities of acoustic anomaly detection systems, particularly in complex industrial environments.

2.2.2.1 Key Insights for Methodology

As seen previously in Section 2.2.1, there are multiple ways to analyze and preprocess audio samples, just as there are numerous ML models that can be trained for anomaly detection.

Given the constraints of limited computational resources and the critical need for high-accuracy and noise-resilient anomaly detection models, as well as building on the current research gaps in the literature, the methodology of this thesis will be carried out with the following considerations in mind:

- AEs are widely used because of their ability to model normal operating conditions and detect deviations effectively. Given their flexibility, this thesis considers LSTM-based AEs to leverage their ability to capture temporal dependencies in audio signals.
- Following the success of [43] in the integration of multiple models, this thesis incorporates a hybrid approach, also combining traditional ML algorithms such as OC-SVM and K-means with the autoencoder model. These models are computationally efficient while maintaining high accuracy and complying with the constraints of industrial environments.
- Feature extraction techniques such as Mel-spectrograms and MFCCs are effective in converting raw acoustic data into structured input for ML models. Furthermore, more emphasis will be placed on feature extraction in general as well as targeting noise resilience and scalability with preprocessing techniques like noise reduction and normalization.
- Standardized benchmarks for model evaluation will be crucial for fair comparison of model performance.

Chapter 3

Methodology

As discussed in the previous chapter, this thesis provides a comparative analysis of three different models trained for anomaly detection. The anomaly detection framework proposed in the methodology follows the pipeline shown in Figure 3.1, whose steps are described below. Section 3.1, explains the data acquisition process in detail, including the structure and limitations of the actual datasets used in this thesis. Section 3.2 outlines the acoustic features that are extracted from the datasets in the exploratory data analysis (EDA). The ML models selected for anomaly detection in this thesis are then presented in Section 3.3, followed by an explanation and discussion of the evaluation metrics used to compare the trained models in Section 3.4.

Figure 3.1 summarizes these steps and presents the necessary steps that must be taken to detect anomalies: including recording audio, processing the signals, training and fine-tuning models, and finally evaluating them. What may at first glance look like simple architecture is in reality a framework built by many smaller steps that lead to a complex system capable of detecting and predicting anomalies for predictive maintenance in the industry.

3.1 About the Dataset

The dataset is arguably one of the most critical components for anomaly detection, as its quality can have a direct impact on data analysis and model performance. This section explains the process of acquiring training data, as well as its format and recording conditions.

3.1.1 Data acquisition

The dataset used in this thesis is based on the dataset used in [43]. It consists of both controlled and real-world environments recordings. The recordings were taken under normal machine working conditions. To collect data, microphones were strategically

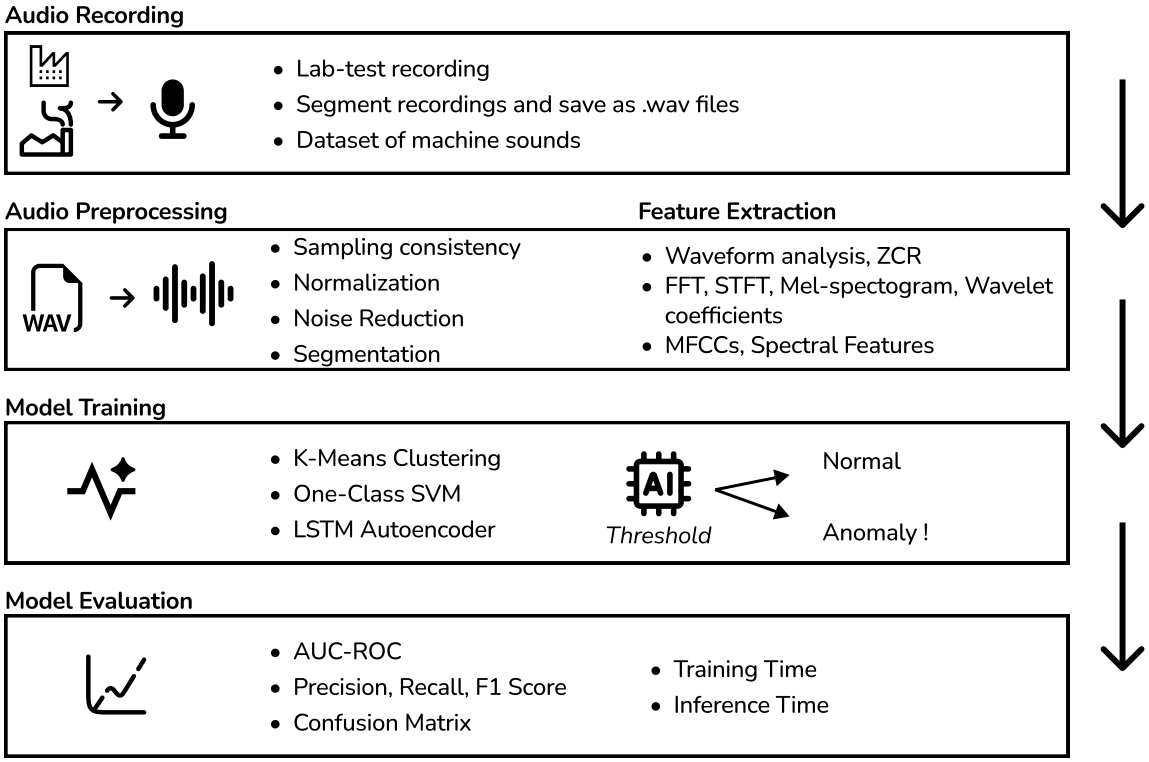


Figure 3.1: General Overview of the Proposed Pipeline for Anomaly Detection

placed to record machine sounds, and subsequently the recordings were segmented into normal and anomalous *.wav* files for further processing. Each audio file has a sampling rate of 44,100 Hz. As an example, Figure 3.2 shows one of three microphones that were placed to record the sounds of the industrial hydropower plant under normal operating conditions.

3.1.2 Splitting of Data for Training and Testing

To maximize robust model evaluation, the datasets are divided into training, evaluation, and testing subsets. This approach aims to balance the representation of normal and anomalous sounds for effective model training. The partitioning can be described as follows.

The normal data is split into three subsets: training, validation, and testing. A fairly extensive part of the normal data is allocated to the training set, with smaller portions reserved for validation and testing. The anomalous sound data is used only for the testing phase, ensuring that models are evaluated against previously unseen anomalies. The test set consists of a combination of normal and anomalous sounds, shuffled to prevent bias.

Figure 3.2: Hydropower Plant Rodundwerk II.

3.1.3 Limitations

The datasets used in this thesis are limited to a short period of real-world anomalies, which for training and testing purposes may not be ideal. However, no further datasets were recorded or added, as the proposed framework aims to show the applicability of the method, and this can still be achieved with a small amount of data.

3.2 Exploratory Data Analysis (EDA)

After the data acquisition process, the EDA allows us to expand our understanding of the recorded data. It is a crucial step in understanding the structure and unique characteristics that the dataset holds. It is a holistic approach that includes examining both the time and frequency domains, identifying the properties of the audio signals, comparing similarities and differences, and extracting features that are useful for subsequent machine learning processing.

3.2.1 Preprocessing Steps

In contrast to other related works, this thesis focuses heavily on the data analysis and preprocessing phases. It can be argued that anomalies can be spotted already before training a model just by looking at the data visualizations if these are processed correctly.

Furthermore, a well-defined preprocessing phase ensures that the audio dataset is clean and well-structured for analysis and model training. This thesis uses the following techniques:

- **Sampling Consistency**: Once each audio file is uploaded, the sampling rate of each recording is checked to ensure uniformity across all datasets. This is crucial to ensure consistent sampling rates across recordings and thus ensure that the recordings are comparable.
- **Audio Normalization**: The audio recordings are normalized using their root means square through their average amplitude levels to ensure consistency across recordings while preserving their feature stability and also enabling easier comparisons.
- **Noise Reduction**: Wavelet-based denoising is applied to each audio recording to enhance signal quality while preserving the characteristics of normal and anomalous sounds [87]. Noise reduction is a fundamental preprocessing step because it allows to provide cleaner features of the audio signals and reducing the influence of noisy backgrounds, potentially leading to better model performance or EDA visualizations. A concrete example in the EDA would be to better visualize the raw audio waveforms. These waveforms are not as useful if the audio recordings have too much noise in the background because you cannot distinguish the waveforms correctly, but if the amplitude structure of the recordings is well defined then they can provide valuable insights on the structure of the dataset.
- **Segmentation**: Before inputting the audio signals into the ML models, the recordings are split into smaller frames to facilitate meaningful feature extraction and maintain a balanced dataset for training and testing [86]. Audio segmentation, in this case specifically overlapping frames, can isolate short audio segments and ensure that important transitions in the audio that span across different boundaries are captured effectively, allowing the models to focus on localized frequency patterns instead of complete recordings or spectograms at once.

Without these steps, audio recordings could potentially be misleading or missing information, representing a systematic error that ultimately affects the results of the complete acoustic analysis.

3.2.2 Feature Extraction

Before training models, raw audio signals must be converted into stable and interpretable representations, particularly in the context of anomaly detection [43]. Their visualizations can also be helpful to spot anomalies by hand, or to understand the nature of the anomalies in a given dataset. Thus, feature extraction represents a pillar of the anomaly detection system. The EDA in this thesis covers all the following features:

1. **Time-Domain Features**

    - **Waveform Analysis**: Visualization of the raw audio signals to observe overall trends and potential anomalies can help detect obvious anomalies such as sudden spikes or irregular patterns in the amplitude of the the waveform. It also informs how the audio recording is structured, for example, if there are cyclical patterns that represent changes in operations mode of the machine or if the waveform is completely unstructured and contains anomalies.
    - **Zero-Crossing Rate (ZCR)**: The ZCR measures the number of times the signal changes polarity, which can indicate signal complexity. In general terms, high ZCR usually means more sudden or unusual changes from the baseline, which could potentially represent anomalies as they are deviations from the normal training data [92].

2. **Frequency-Domain Features**

    - **Fast Fourier Transform (FFT) Amplitude Spectrum**: It displays the energy distribution across different frequencies in the signal [9], enabling the identification of unexpected amplitude peaks in unusual frequencies that may indicate abnormal mechanical behaviors.
    - **Short-Time Fourier Transform (STFT) Spectogram**: The STFT displays the frequency and spectral content of a signal over time, offering a time-frequency view of the recording [63], which when converted into a spectogram can be helpful in determining the high or low intensity frequencies that correspond to anomalies.
    - **Mel-Spectrograms**: Converts the frequency domain into a perceptually meaningful representation using the Mel scale, so that the energy distribution of the signal becomes humanly perceivable [80], potentially visualizing anomalies if there are disruptions in the energy levels of the recording.

- **Wavelet Coefficients**: Because the wavelet transform decomposes a signal into multiple resolutions levels of a signal, it effectively lets us analyze the acoustic changes locally at different scales and can be helpful to spot irregularities of a recording at different frequencies [103].

3. **High-Level Features**

   - **Mel-Frequency Cepstral Coefficients (MFCCs)**: The MFCCs are inroduced into the thesis because they compress the frequency-domain of a signal into a compact set of coefficients [34] that in this context highlight the salient spectral features of the recording, making it helpful to detect anomalies.
   - **Spectral Centroid**: The spectral centroid of an audio signal indicates where most of the frequency energy is concentrated [40], thus providing a comparison between the frequencies between normal and anomalous data.
   - **Spectral Rolloff**: The spectral rolloff measures the frequency of the total spectral energy below 85%. So, for example, if a large part of the energy in the recording is concentrated in the lower frequencies, the spectral rolloff will be lower [40].

The selected features collectively enable robust analysis of audio signals, offering the opportunity to detect abnormal behaviors in the audio signals even before training models for anomaly detection, ultimately improving anomaly detection performance and general analysis depth.

3.3 ML Models

While the EDA enables a strong foundation to understand audio signals and identify preliminary patterns, ML models are the backbone of ADSs. ML models learn from the extracted acoustic features in the EDA and generalize them to detect patterns that may not be recognizable in plain sight, or in this case, anomalies. Therefore, in line with the goals of this thesis and insights from the related work, this thesis adopts a hybrid framework for acoustic anomaly detection by leveraging a combination of unsupervised and deep learning techniques. The three models implemented in this thesis have been selected based on their complementary strengths to ensure a robust ADS that can capture anomalies in diverse acoustic environments. These models include: clustering-based identification, boundary learning, and temporal pattern recognition. Each one of the

models processes the extracted features and acoustic signals uniquely and contributes to the anomaly detection pipeline in its own way.

1. **K-means Clustering**, a lightweight yet effective unsupervised approach for detecting anomalies based on feature distributions and grouping of data points into clusters [5]. The reason behind using K-means as an anomaly prediction algorithm is mainly its minimal computational overhead and intuitive clustering based approach, making it suitable for real-time applications with limited computational resources.

2. **One-Class Support Vector Machine (OC-SVM)**, a model that learns a tight decision boundary around normal (non-anomalous) data, flagging deviations from the decision boundary as anomalies [57, 65]. This model is well suited for anomaly detection due to its adaptability to the structure of the normal data to draw a decision boundary and its capability to identify subtle acoustic deviations.

3. **Long Short-Term Memory Autoencoder (LSTM AE)**, a deep learning model capable of capturing temporal dependencies in sequential acoustic signals, excelling at identifying deviations from normal behaviors, as can be seen in [29] where accuracy rates of 99% were achieved when detecting anomalies.

By integrating these three models into a unified anomaly detection framework, it allows the analysis to benefit from each of their strengths, thus also improving the robustness of the framework. While K-means and OC-SVM offer efficient unsupervised baselines for anomaly detection, the LSTM autoencoder captures more intricate patterns that might have been missed by traditional techniques. The selection of these models is based on the literature reviews in the previous chapter and aims to offer a scalable, robust, and complementary framework for detecting anomalies for PdM.

3.4 Evaluation Metrics

The evaluation of the different models must be supported by a systematic and objective comparison. Evaluation metrics serve as key performance indicators (KPIs) for assessing the accuracy and computational efficiency of the model. The metrics can provide tangible comparisons and quantitative insights into how well a model performs and differentiates itself from the others, identifying trade-offs among the different approaches. Furthermore, since anomaly detection models tend to be trained with rather unbalanced datasets -

where normal audios largely outnumber anomalies - using only accuracy metrics could be misleading by not providing the whole picture for companies who want to introduce ADSs into their factories and also want to consider computational metrics. That is why both performance and computational-focused metrics are used. First, we have the anomaly detection performance metrics:

AUC-ROC (Area Under the Curve - Receiver Operating Characteristic): This metric evaluates the model's capacity to distinguish normal and anomalous data. This means that it evaluates the overall classification performance of the model. It is measured from 0 to 1, 1 being the perfect score and means that the model can perfectly separate normal and anomalous data [25].

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{3.1}$$

Precision, Recall and F1-Score: While precision measures how many of the positive predictions of the model are correct, recall measures how many of the actual positives were correctly predicted by the model. F1-score is the harmonic mean of precision and recall, balancing the scores of both metrics into one. In cases where the datasets are imbalanced, which is the case in the anomaly datasets, the F1-score is preferred over accuracy because it provides a more balanced evaluation of the model considering both precision and recall [22].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

Confusion Matrix: A confusion matrix visualizes the prediction summary of a model in matrix form. It includes true positives, false positives, true negatives, and false negatives into the matrix and simplifies the analysis of trade-offs between missed and mistaken classifications [22, 43].

These metrics are the baseline for measuring the model overall performance in detecting anomalies, but beyond plain detection accuracy, real-world applications require models to be computationally feasible for deployment in industrial environments in a

timely manner. That is why both training and inference time have also been added as fundamental evaluation metrics.

Training Time: Measures how long it takes to train a model on the dataset. Some methods, such as deep learning models, tend to require significantly longer training times than traditional statistical models due to their architecture complexity, which could be an issue if a hydropower plant wants to introduce real-time ADSs. So, training time can be a valuable insight for companies or manufacturers that have tight time constraints and need fast ML models.

Inference Time: The inference time measures how quickly a model can classify an instance of the dataset. This is relevant because anomalies should be spotted quickly in order to proceed with predictive maintenance.

With these evaluation metrics, it is possible not only to compare the performance of the models objectively, but also to analyze the trade-offs that could arise in real-world scenarios, where time and computational resources could be a constraint. Or, for example, the fact that, in the context of anomaly detection, higher recall (less false negatives) can be considered one of the most important metrics because missed anomalies can be more costly as false positives [43].

As mentioned earlier, the EDA plays a fundamental role in this thesis and not only aligns with the goal of developing a robust and scalable anomaly detection framework, but also reinforces the importance of understanding the data before feeding it to ML models. Furthermore, the integration of three different ML models into the system ensures flexibility and adaptability in deployment and training across distinct industries, but especially in high-altitude hydropower plants.

Finally, the defined evaluation metrics proposed to rate the performance of the ML models offers a holistic approach that not only evaluates them in terms of detection performance but also computational feasibility, which can be crucial in real-world scenarios.

Together, the methodological steps presented in this chapter build a cohesive framework, guided by empirical evaluation for a solid anomaly detection system.

Chapter 4

Implementation

This chapter outlines the practical implementation of the anomaly detection framework proposed in Chapter 3. The general structure of the anomaly detection system and tools used to develop the system are presented in Section 4.1. Section 4.2 describes the implementation of Exploratory Data Analysis (EDA), focusing on how audio features are extracted and prepared for model input. Finally, Section 4.3 covers the data partitioning strategy for model training and evaluation, the implementation of each ML model, and its respective evaluation.

4.1 System Overview

The anomaly detection system is divided into two separate Jupyter Notebooks in a Python-based environment. One notebook is dedicated to the EDA 4.2 which consists of visualizing key features and extracting necessary data representations for model training, while the second notebook handles model trainings and the evaluation of anomaly predictions 4.3. This dedicated separation of the two notebooks aims to maintain clarity and modularity in the workflow, while highlighting the importance of both parts of the system.

For developing the complete framework in Python, key libraries were used. The most relevant ones in terms of contributing to anomaly detection aspects include:

- **NumPy & Pandas:** For numerical operations and data manipulation.
- **Matplotlib & Seaborn:** For visualizing data and results.
- **Librosa:** For audio loading, processing, and feature extraction.
- **scikit-learn:** For implementing K-means clustering and OC-SVM.
- **PyTorch:** For building and training of the LSTM AE.

4.2 Exploratory Data Analysis (EDA)

In the first notebook, key acoustic features are extracted and visualized to understand the properties of the dataset. The implementation of the most relevant functions of the EDA will be illustrated in the following code snippets.

4.2.1 Data preprocessing

The first step in the EDA is to prepare the data for subsequent analysis. Concretely, the first step is to denoise the audios and ensure consistent loudness. A wavelet-based denoising function is applied to the audio recording to suppress background noise while preserving the integrity of the features in the recordings. After denoising, gain normalization is applied to a target loudness level in decibels full scale (dBFS) to ensure the audios are comparable in further stages.

Listing 4.1: Data Preprocessing Steps

```
4.1.1  def wavelet_denoise(audio, wavelet='db4', level=3,
       threshold_multiplier=1.5):
4.1.2      coeffs = pywt.wavedec(audio, wavelet, level=level)
4.1.3      detail_coeffs = coeffs[1]
4.1.4      sigma = np.median(np.abs(detail_coeffs)) / 0.6745
4.1.5      threshold = sigma * threshold_multiplier
4.1.6      new_coeffs = list(coeffs)
4.1.7      for i in range(1, min(3, len(coeffs))):
4.1.8          coeffs_i = coeffs[i]
4.1.9          sign = np.sign(coeffs_i)
4.1.10         magnitude = np.abs(coeffs_i)
4.1.11         thresholded = sign * np.maximum(magnitude - threshold, 0)
4.1.12         new_coeffs[i] = thresholded
4.1.13     denoised_audio = pywt.waverec(new_coeffs, wavelet)
4.1.14     if len(denoised_audio) > len(audio):
4.1.15         denoised_audio = denoised_audio[:len(audio)]
4.1.16     elif len(denoised_audio) < len(audio):
4.1.17         denoised_audio = np.pad(denoised_audio, (0, len(audio) - len(
       denoised_audio)))
4.1.18     return denoised_audio
4.1.19
4.1.20 def normalize_audio(audio, target_dBFS=-20):
4.1.21     rms = np.sqrt(np.mean(audio**2))
4.1.22     current_dBFS = 20 * np.log10(rms) if rms > 0 else -80
4.1.23     gain = 10**((target_dBFS - current_dBFS) / 20)
```

```
4.1.24          normalized_audio = audio * gain
4.1.25          if np.max(np.abs(normalized_audio)) > 1.0:
4.1.26              normalized_audio = normalized_audio / np.max(np.abs(
        normalized_audio))
4.1.27          return normalized_audio
4.1.28
```

4.2.2 Generating a Normalized Mel-pectogram

To compute the Mel-spectogram of an audio file, first the STFT has to be calculated to obtain its spectogram, using the built-in librosa function. Then the results are converted into a Mel-spectogram by mapping the frequencies to the Mel scale with 128 Mel bins and a hop length of 512. The result is then converted to decibel and normalized between 0 to 1 to ensure consistent comparisons.

Listing 4.2: Generating Mel-spectrogram

```
4.2.1   def generate_mel_spectrogram(audio, sr, n_mels=128, n_fft=1024,
        hop_length=512):
4.2.2       S = np.abs(librosa.stft(audio, n_fft=n_fft, hop_length=hop_length)
        ) ** 2
4.2.3       mel = librosa.feature.melspectrogram(S=S, sr=sr, n_mels=n_mels)
4.2.4       mel_db = librosa.power_to_db(mel, ref=np.max)
4.2.5       mel_norm = (mel_db - mel_db.min()) / (mel_db.max() - mel_db.min())
4.2.6       return mel_norm
```

4.2.3 Individual Feature extraction

Various signal descriptors mentioned in 3.2.2 are extracted to characterize both normal and anomalous acoustic behavior. These features include MFCCs, spectral centroid, spectral rolloff, spectral contrast, and zero-crossing rate which all have a dedicated built-in function in the librosa library.

Listing 4.3: Individual Feature Extraction

```
4.3.1   def extract_mfccs(audio, sr, n_mfcc=13):
4.3.2       mfccs = librosa.feature.mfcc(y=audio, sr=sr, n_mfcc=n_mfcc)
4.3.3       return mfccs
4.3.4
4.3.5   def extract_spectral_centroid(audio, sr):
4.3.6       centroid = librosa.feature.spectral_centroid(y=audio, sr=sr)
4.3.7       return centroid
4.3.8
```

```
4.3.9     def extract_spectral_rolloff(audio, sr):
4.3.10        rolloff = librosa.feature.spectral_rolloff(y=audio, sr=sr)
4.3.11        return rolloff
4.3.12
4.3.13    def extract_spectral_contrast(audio, sr):
4.3.14        contrast = librosa.feature.spectral_contrast(y=audio, sr=sr)
4.3.15        return contrast
4.3.16
4.3.17    def extract_zero_crossing_rate(audio):
4.3.18        zcr = librosa.feature.zero_crossing_rate(audio)
4.3.19        return zcr
4.3.20
```

4.2.4 FFT, STFT and Wavelet Coefficients

Additionally, key frequency-domain metrics such as the FFT amplitude spectrum, the STFT, and the wavelet coefficients are extracted to provide a summary of the signal's frequency content and plotted for visual analysis. The computations are mostly done by libraries such as NumPy for numerical operations, librosa for audio processing tasks and PyWavelets for performing the discrete wavelet transforms. A summary of these functions is illustrated in the following code snippet:

Listing 4.4: Computing FFT, STFT, and Wavelet Coefficients (No Plotting)

```
4.4.1     def compute_fft(audio, sr):
4.4.2         N = len(audio)
4.4.3         yf = fft(audio)
4.4.4         xf = np.linspace(0.0, sr/2, N//2)
4.4.5         amplitudes = 2.0/N * np.abs(yf[:N//2])
4.4.6         return xf, amplitudes
4.4.7
4.4.8     def compute_stft(audio, sr, n_fft=1024, hop_length=512):
4.4.9         stft_result = librosa.stft(audio, n_fft=n_fft, hop_length=
          hop_length)
4.4.10        return stft_result
4.4.11
4.4.12    def compute_wavelet(audio, wavelet='db4', level=5):
4.4.13        coeffs = pywt.wavedec(audio, wavelet, level=level)
4.4.14        return coeffs
```

4.3 Model Training & Evaluation

4.3.1 Data Splitting Strategy

The data segmentation strategy begins by loading two separate audio files — one representing normal operation and one containing anomalous events - into the environment. The audio recordings are then converted into a Mel-spectogram and segmented into overlapping frames, where the frame size is determined by a specified duration relative to the hop length and a hop ratio (e.g., 20% overlap) defines the stride between frames; each frame retains its two-dimensional structure (Mel bins $\times$ frame width) and is saved as a NumPy array.

Next, only the frames derived from normal audio are used to construct the training and validation sets by applying scikit-learn's `train_test_split` — approximately 15% of the normal frames are reserved as test data while the remaining 85% is divided into roughly 70% for training and 15% for validation.

The final test set is assembled by combining these normal test frames with the anomalous frames, assigning the corresponding labels (0 for normal and 1 for anomalous), and shuffling the combined data set to avoid any ordering bias. Finally, all audio sets are saved as `.npy` files to ensure efficient loading and seamless integration into the subsequent model training and evaluation stages. The implementation of this data segmentation strategy was taken from [43].

Listing 4.5: Data Splitting for Model Training and Evaluation

```
4.5.1  mel_db_anomalous, sr_anomalous = generate_mel_spectrogram(
       anomalous_audio_path)
4.5.2  mel_db_normal, sr_normal = generate_mel_spectrogram(normal_audio_path)
4.5.3  assert sr_anomalous == sr_normal, "Sampling rates do not match!"
4.5.4
4.5.5  frame_size = int((time_per_frame * sr_anomalous) / hop_length)
4.5.6  hop_size = int(frame_size * hop_ratio)
4.5.7  anomalous_frames = generate_frames(mel_db_anomalous, frame_size,
       hop_size)
4.5.8  normal_frames = generate_frames(mel_db_normal, frame_size, hop_size)
4.5.9
4.5.10 np.save(output_anomalous_frames_path, anomalous_frames)
4.5.11 np.save(output_normal_frames_path, normal_frames)
4.5.12
4.5.13 normal_frames = np.load(output_normal_frames_path)
4.5.14 anomalous_frames = np.load(output_anomalous_frames_path)
```

```
4.5.15  normal_train_val, normal_test = train_test_split(normal_frames,
        test_size=0.15, random_state=42)
4.5.16  normal_train, normal_val = train_test_split(normal_train_val,
        test_size=0.1765, random_state=42)
4.5.17
4.5.18  test_frames = np.concatenate([normal_test, anomalous_frames], axis=0)
4.5.19  test_labels = np.concatenate([np.zeros(len(normal_test)), np.ones(len(
        anomalous_frames))])
4.5.20  indices = np.arange(len(test_frames))
4.5.21  np.random.shuffle(indices)
4.5.22  test_frames = test_frames[indices]
4.5.23  test_labels = test_labels[indices]
4.5.24
4.5.25  np.save(train_frames_path, normal_train)
4.5.26  np.save(val_frames_path, normal_val)
4.5.27  np.save(test_frames_path, test_frames)
4.5.28  np.save(test_labels_path, test_labels)
```

4.3.2 K-means Clustering

After the data is loaded and split accordingly, it is time to train the models. In the case of K-means, the algorithm starts by flattening the data frames into one-dimensional vectors of size $n\_mels \times$ frame_size. Since the K-means algorithm uses Euclidean distances, the data is standardized using a `StandardScaler` which centers and scales the features to units. For training, the model is set to use a single cluster ($n\_clusters = 1$) that represents the central tendency of the normal data. The model is then trained with a fixed random seed (e.g., `random_state=42`) to ensure reproducibility and the model is saved for later evaluation. The following code snippet shows the pipeline for the K-means model:

Listing 4.6: K-means Model Implementation

```
4.6.1  X_train_kmeans = train_frames.reshape(len(train_frames), -1)
4.6.2  scaler_kmeans = StandardScaler()
4.6.3  X_train_kmeans = scaler_kmeans.fit_transform(X_train_kmeans)
4.6.4  n_clusters = 1
4.6.5  kmeans = KMeans(n_clusters=n_clusters, random_state=42)
4.6.6  kmeans.fit(X_train_kmeans)
4.6.7  joblib.dump(scaler_kmeans, "../../Models/KMeans/scaler_kmeans.pkl")
4.6.8  joblib.dump(kmeans, "../../Models/KMeans/kmeans_model.pkl")
```

4.3.3 OC-SVM

Similarly to the K-means algorithm, the OC-SVM model is trained solely on normal data to learn a decision boundary. The workflow is similar, first, the audio frames are flattened into a one-dimensional vector using the `reshape(len(array), -1)` function and then a `StandardScaler` is used to center the data (zero mean) and scale it to unit variance. The SVM model is then created with a Radial Basis Function (RBF) kernel for capturing nonlinear relationships in the data and sets $nu = 0.05$ as an an upper bound on the fraction of training errors and a lower bound on the fraction of support vectors. This simplified code snippet shows the the steps mentioned:

Listing 4.7: =OC-SVM Model Implementation

```
4.7.1  X_train_ocsvm = normal_train.reshape(len(normal_train), -1)
4.7.2  scaler_ocsvm = StandardScaler()
4.7.3  X_train_ocsvm = scaler_ocsvm.fit_transform(X_train_ocsvm)
4.7.4  ocsvm = OneClassSVM(kernel='rbf', gamma='scale', nu=0.05)
4.7.5  ocsvm.fit(X_train_ocsvm)
4.7.6  joblib.dump(scaler_ocsvm, "../../Models/OCSVM/scaler_ocsvm.pkl")
4.7.7  joblib.dump(ocsvm, "../../Models/OCSVM/ocsvm_model.pkl")
```

4.3.4 LSTM Autoencoder

The LSTM Autoencoder captures both spectral and temporal dynamics within audio frames. Its encoder-decoder architecture first compresses the input sequence into a lower-dimensional latent space and then uses the decoder to reconstruct the original input from this compact representation.

In this thesis, the encoder uses a single-layer LSTM with input size of 128 and 64 hidden units. The encoder compresses the input into a latent projection of 32 dimensions by extracting a 64-dimensional hidden state and passing it through a fully connected layer.

The decoder reverses this process by first expanding the latent projections back to 64 dimensions and repeating the process along the time axis to match the sequence length. Then a second LSTM layer reconstruction of the original sequence was performed using a configuration of 64 input features and 128 hidden units.

The autoencoder is then trained to minimize the loss of Mean Squared Error (MSE) between the input x and its reconstruction $\hat{x}$. The MSE loss function is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2,$$

where x_i is the original input, $\hat{x}_i$ is the reconstructed output, and n is the number of data points [48]. In addition, an Adam optimizer is used with a learning rate of 1×10^{-4} to ensure stable convergence during training, which is run for a fixed number of 10 epochs. The trained model is then saved for later evaluation in the test set.

The definition of the explained architecture is represented in the following code snippet.

Listing 4.8: LSTM-AE Model Implementation

```
4.8.1 class LSTMAutoencoder(nn.Module):
4.8.2     def __init__(self, input_size=128, hidden_size=64, latent_size=32,
      num_layers=1):
4.8.3         super(LSTMAutoencoder, self).__init__()
4.8.4         self.encoder = nn.LSTM(input_size, hidden_size, num_layers,
      batch_first=True)
4.8.5         self.fc_enc = nn.Linear(hidden_size, latent_size)
4.8.6         self.fc_dec = nn.Linear(latent_size, hidden_size)
4.8.7         self.decoder = nn.LSTM(hidden_size, input_size, num_layers,
      batch_first=True)
4.8.8
4.8.9     def forward(self, x):
4.8.10        # Encode
4.8.11        enc_out, _ = self.encoder(x)
4.8.12        last_hidden = enc_out[:, -1, :]
4.8.13        latent = self.fc_enc(last_hidden)
4.8.14        # Decode
4.8.15        dec_input = self.fc_dec(latent).unsqueeze(1)
4.8.16        dec_input = dec_input.repeat(1, x.size(1), 1)
4.8.17        dec_out, _ = self.decoder(dec_input)
4.8.18        return dec_out
```

4.3.5 Model Evaluation

As discussed in Section 3.4, the performance of the model is evaluated in terms of both the accuracy of detection accuracy and computational effectiveness. This section explains

how relevant performance metrics such as ROC AUC, precision, recall, and F1-score are captured for each model, and how training and inference times are measured.

1. **K-means Evaluation:** The evaluation process begins by loading the pre-trained K-means model and its scaler. The test frames are reshaped and standardized before being passed to the model, and for each sample the minimum Euclidean distance to the cluster centroids is computed and used as the anomaly score. Using the 97th percentile of distances on the validation set, a threshold for anomalies is computed, and the samples with distances exceeding this threshold are flagged as anomalies. Although using the 95th percentile was initially considered, the 97th percentile was ultimately chosen to prioritize capturing all true anomalies, even at the cost of a few additional false positives. The ROC AUC score is computed using the raw distance values, reflecting how well the model separates normal and anomalous data. The precision, recall, and F1-score are computed based on binary predictions after thresholding. Additionally, the inference time is recorded using Python's `time` library and the training time is already captured during the model's training.

Listing 4.9: K-means Evaluation

```
4.9.1  scaler_kmeans = joblib.load("../../Models/KMeans/scaler_kmeans.
       pkl")
4.9.2  kmeans = joblib.load("../../Models/KMeans/kmeans_model.pkl")
4.9.3  X_test_kmeans = test_frames.reshape(len(test_frames), -1)
4.9.4  X_test_kmeans = scaler_kmeans.transform(X_test_kmeans)
4.9.5  start_time = time.time()
4.9.6  dist_test = kmeans.transform(X_test_kmeans).min(axis=1)
4.9.7  inference_time_kmeans = time.time() - start_time
4.9.8  X_val_kmeans = val_frames.reshape(len(val_frames), -1)
4.9.9  X_val_kmeans = scaler_kmeans.transform(X_val_kmeans)
4.9.10 dist_val = kmeans.transform(X_val_kmeans).min(axis=1)
4.9.11 threshold_kmeans = np.percentile(dist_val, 97)
4.9.12 y_pred_kmeans = (dist_test > threshold_kmeans).astype(int)
4.9.13 roc_auc_kmeans = roc_auc_score(test_labels, dist_test)
4.9.14 precision_kmeans, recall_kmeans, f1_kmeans, _ =
       precision_recall_fscore_support(test_labels, y_pred_kmeans,
       average='binary')
4.9.15 conf_kmeans = confusion_matrix(test_labels, y_pred_kmeans)
```

2. **OC-SVM:** The procedure for the OC-SVM is very similar to the one used for the K-means model. But in this case, the OC-SVM's decision function is used,

where negative decision scores are the ones that indicate anomalies. The process for computing the evaluation metrics is the same as can be seen in the following snippet.

Listing 4.10: OC-SVM Evaluation

```
4.10.1  scaler_ocsvm = joblib.load("../../Models/OCSVM/scaler_ocsvm.pkl")

4.10.2  ocsvm = joblib.load("../../Models/OCSVM/ocsvm_model.pkl")
4.10.3  X_test_ocsvm = test_frames.reshape(len(test_frames), -1)
4.10.4  X_test_ocsvm = scaler_ocsvm.transform(X_test_ocsvm)
4.10.5  start_time = time.time()
4.10.6  decision_scores = ocsvm.decision_function(X_test_ocsvm)
4.10.7  inference_time_ocsvm = time.time() - start_time
4.10.8  y_pred_ocsvm = (decision_scores < 0).astype(int)
4.10.9  roc_auc_ocsvm = roc_auc_score(test_labels, -decision_scores)
4.10.10 precision_ocsvm, recall_ocsvm, f1_ocsvm, _ =
        precision_recall_fscore_support(test_labels, y_pred_ocsvm,average
        ='binary')
4.10.11 conf_ocsvm = confusion_matrix(test_labels, y_pred_ocsvm)
```

3. **LSTM AE:** In the case of the LSTM, the model is first loaded and set to evaluation mode. The test frames are converted to PyTorch tensors and fed through the model to obtain reconstructions. Using the MSE, the reconstruction error is computed for each sample and the threshold is determined based on the 97th percentile of validation errors. Similarly to the previous models, if the threshold is exceeded, then the frame gets marked as anomaly. Additionally the code snippet also illustrates the computation of the relevant metrics discussed earlier, which are computed in the same way as for the other models.

Listing 4.11: LSTM AE Evaluation

```
4.11.1  lstm_ae = LSTMAutoencoder(input_size=128, hidden_size=64,
        latent_size=32, num_layers=1)
4.11.2  lstm_ae.load_state_dict(torch.load("../../Models/LSTM_AE/lstm_ae.
        pth"))
4.11.3  lstm_ae = lstm_ae.float()
4.11.4  lstm_ae.eval()
4.11.5  start_time = time.time()
4.11.6  with torch.no_grad():
4.11.7      test_outputs = lstm_ae(test_frames_tensor_lstm)
```

```
4.11.8          test_loss_tensor = nn.functional.mse_loss(test_outputs,
        test_frames_tensor_lstm, reduction='none')
4.11.9          test_mse = test_loss_tensor.reshape(test_loss_tensor.size(0),
         -1).mean(dim=1).numpy()
4.11.10     inference_time_lstm = time.time() - start_time
4.11.11     with torch.no_grad():
4.11.12         val_outputs = lstm_ae(val_frames_tensor_lstm)
4.11.13         val_loss_tensor = nn.functional.mse_loss(val_outputs,
        val_frames_tensor_lstm, reduction='none')
4.11.14         val_mse = val_loss_tensor.reshape(val_loss_tensor.size(0),
        -1).mean(dim=1).numpy()
4.11.15     threshold_lstm = np.percentile(val_mse, 97)
4.11.16     y_pred_lstm = (test_mse > threshold_lstm).astype(int)
4.11.17     roc_auc_lstm = roc_auc_score(test_labels, test_mse)
4.11.18     precision_lstm, recall_lstm, f1_lstm, _ =
        precision_recall_fscore_support(test_labels, y_pred_lstm, average
        ='binary')
4.11.19     conf_lstm = confusion_matrix(test_labels, y_pred_lstm)
```

Furthermore, for all models, the ROC curves and confusion matrices are plotted for visual assessment of their performance, but not shown in the code snippets for conciseness purposes. The training time for all models is measured when the models are trained. This overview shows that the evaluation process provides a comprehensive approach that measures each model's ability to detect anomalies in a standardized manner using the same techniques.

Chapter 5

Results and Discussion

This chapter presents the results and findings of the anomaly detection framework implemented in this thesis and offers a comprehensive discussion and evaluation of the performance with respect to the evaluation metrics mentioned in Section 3.4.

The discussion is organized in specific sections to provide a structured overview of the results. Section 5.1 gives an overview of the experimental setup, Section 5.2 presents the key results from the EDA in 5.2.1 and the performance of the individual models in 5.2.2. Furthermore, these results are then systematically compared and interpreted in Section 5.3 in the context of the research objectives and the review of the literature.

5.1 Experimental Setup

This section outlines the experimental setup used for the evaluation of the thesis to meet its research objectives. For reproducibility purposes, both the hardware and software environments used for the EDA and to train the models are summarized below.

- **Hardware Specifications:** MacBook Air (M2, 2022), 8-core CPU with 4 performance cores and 4 efficiency cores, 8-core GPU, 16-core Neural Engine, 100GB/s memory bandwidth, 8GB unified memory and running on macOS Sequoia 15.2.
- **Software Specifications:** PyCharm 2024.2.4 (Professional Edition), with VM OpenJDK 64-Bit Server VM by JetBrains s.r.o. Both EDA and model training were run using Jupyter Notebooks.

As mentioned in Section 3.1, the dataset and methodology of this thesis build upon the data made available in [43]. The dataset comprises recordings from both controlled laboratory settings and real-world environments. The same methodologies discussed in Section 3.1 were applied uniformly across all recordings, highlighting the reproducibility and transferability of the framework. The audio recordings used in this thesis consist of:

- **Washing Machine:** Recordings obtained from a controlled laboratory setup involving a washing machine, where anomalies were artificially induced to simulate abnormal operating conditions. Two recordings are used:
  - *normal_audio_wm*: A 19-minute and 49-second recording of a washing machine in normal operating mode.
  - *anomalous_audio_wm*: A 1-minute and 26-second recording of anomalous knocking sounds by introducing shoes instead of clothes into the washing machine.
- **Synthetic Industrial Machine:** This dataset consists of two recordings during normal operation mode of the hydropower plant, including synthetic anomalies in a controlled environment.
  - *normal_audio_synthetic*: A 12-minute and 46-second recording of the machine under normal operating conditions.
  - *anomalous_audio_synthetic*: A 3-minute and 29-second recording of artificially induced anomalies by hitting a hammer with a shovel, providing a similar sound to real-world anomalies.
- **Real Industrial Machine:** The main dataset consists of two recordings captured in a real pumped storage hydropower plant in operation. This set includes:
  - *normal_audio_real*: A 59-minute and 53-second recording of the machine working under routine conditions, providing a baseline for normal operation.
  - *anomalous_audio_real*: A 5-second recording containing real anomalies produced by the machine during operation.

The experimental setup, including hardware, software and dataset specifications, was chosen to align with the proposed framework that aims to address the central thesis goals: leveraging acoustic sensing for anomaly detection, investigating state-of-the-art ML techniques, and integrating these insights into a comprehensive framework for predictive maintenance in hydropower plants. The setup ensures reproducibility and demonstrates the applicability of the framework with distinct datasets, allowing to integrate spectral analysis techniques and evaluate the proposed approach extensively.

5.2 Presentation of Results

This section covers the details of the results obtained from both the EDA and the performance evaluation of the anomaly detection models.

5.2.1 Exploratory Data Analysis Outcomes

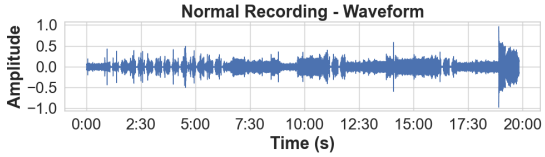
The EDA aims to uncover the unique sound and spectral characteristics of the acoustic signals in each dataset, providing insights to support the distinction needed for anomaly detection. The analysis includes visualizations in both the time and frequency domains, as well as the extraction of high-level acoustic features which provide a comprehensive overview of the audio recordings and their anomalies. For brevity, only the most relevant visualizations and insights are presented, specifically those that most clearly show the distinction between normal and anomalous recordings in each dataset. The complete results of the EDA can be found in the Appendix A.

5.2.1.1 Washing Machine Dataset

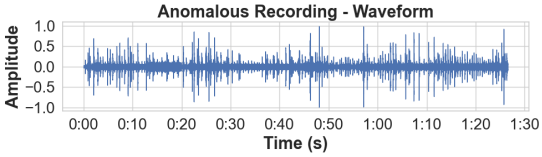
Visual analysis of the raw waveforms of the recordings can provide valuable insight into the acoustic characteristics of the recordings. Additionally, at this step, the audio recordings are already normalized, so it is easier to compare them. Figure 5.1 presents the waveforms of both normal and anomalous recordings. It can be seen in Figure 5.1a that the normal recording exhibits a relatively uniform amplitude distribution, indicating a relatively consistent signal strength throughout. The spikes exhibited in the normal recording, especially in the beginning, represent the different operation modes of a washing machine. In contrast, the anomalous waveform in Figure 5.1b displays much more abrupt spikes and irregular amplitude fluctuations throughout the recording. In comparison, the normal waveform is much more compact than the anomalous recording. The sudden amplitude changes in the anomalous recording are indicative of transient acoustic events, which likely correspond to the induced anomalies in the experiment.

In this case, the focus on amplitude as a key preliminary indicator is due to the nature of this experiment, where anomaly-induced events such as shoes colliding with the walls of the washing machine cause sharp and louder noises. Since amplitude strictly correlates to the signal strength of the recording, such irregularities are a strong signal of abnormal behavior in this context and thus help indicate anomalies.

Another relevant visualization of the audio recordings is the Mel-spectogram, which reveals time-frequency patterns in the recordings. The Mel-spectogram of both audio



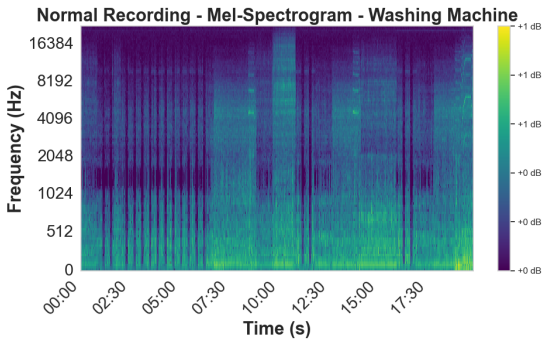
(a) Normal Washing Machine Recording — Raw Waveform



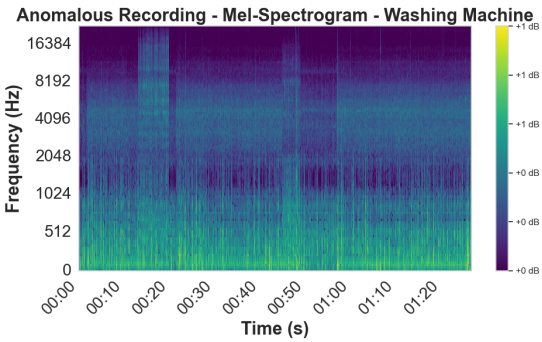
(b) Anomalous Washing Machine Recording — Raw Waveform

Figure 5.1: Comparison of Anomalous and Normal Recordings Raw Waveforms

recordings can be seen in Figure 5.2b. The normal Mel-spectogram in 5.2a reveals a cyclical pattern, especially in the first quarter of the recording, that suggests the operating cycle of a washing machine (*e.g.,* tumbling and spinning). For example, vertical striping patterns indicate periodic drum rotations. In contrast, the anomalous Mel-spectogram in Figure 5.2b shows no apparent cyclical pattern. It also can be seen that the energy distribution is not as uniform as in the normal recording; this can be detected in the thinner energy spikes in the anomalous Mel-spectogram, which can suggest the collisions caused by the shoe tumbling introduced into the washing machine.



(a) Normal Washing Machine Recording — Mel-Spectogram

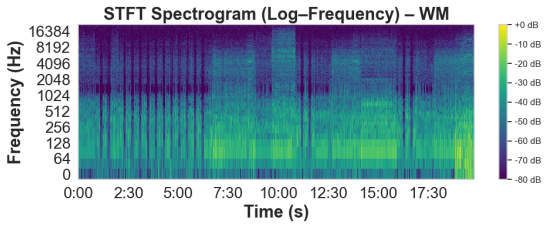


(b) Anomalous Washing Machine Recording — Mel-Spectogram

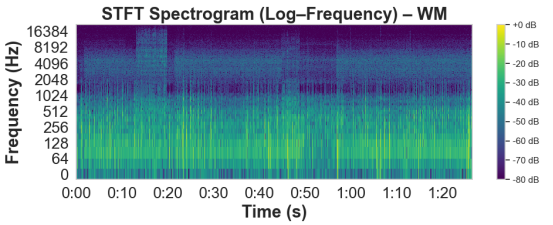
Figure 5.2: Comparison of Anomalous and Normal Recordings Mel-Spectograms

The smoother energy distribution in the normal recording compared to the anomalous recording can also be seen in the STFT-spectogram shown in Figure 5.3. This STFT-spectogram provides a complementary view of the audio signals by preserving more raw frequency information of the signals using the actual frequency content with linear/logarithmic scaling and without applying the Mel filter bank. Apart from the energy changes from the different washing operation modes seen in the normal recording (Figure 5.2a), the energy is well distributed which can be noticed at the smoother distribution of colors

in the STFT-spectogram, while in the anomalous recording these color shifts are much more abrupt and spiky, as can be recognized by the thinner energy stripes. This again, shows abnormal conditions in the anomalous recording, which can lead to deducing the anomalies in that recording.



(a) Normal Washing Machine Recording — STFT-Spectogram



(b) Anomalous Washing Machine Recording — STFT-Spectogram

Figure 5.3: Comparison of Anomalous and Normal Recordings STFT-Spectograms

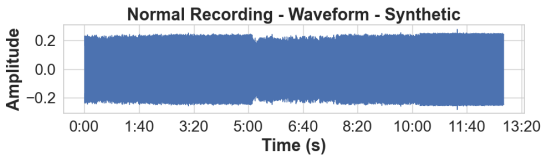
Furthermore, the EDA also captures features such as spectral centroid, spectral roll-off, and spectral contrast. The findings of these recordings are shown in Table 5.1. In this case, the mean centroid of the anomalous recording is lower than the one for the normal recording. This can suggest that the energy in the anomalous recording is more concentrated in the lower frequencies during anomalous events, for example, low-frequency sounds such as the shoes colliding with the washing machine walls. The spectral roll-off is also lower for the anomalous recording, indicating less high-frequency components during anomalies. Finally, the spectral contrast values lie relatively close. These slight variations may reflect subtle differences in the dynamic range of the frequency components. Interestingly, the zero crossing rate of the normal recording is also higher, at 0.070 compared to 0.047 in the anomalous recording. A higher zero crossing rate usually means more sudden or unusual changes from the baseline, which can potentially be flagged as anomalies. Although these findings are counterintuitive, it can be argued that due to the nature of this experiment the zero crossing rate is higher in the normal recording due to the shifting operation modes and high frequency vibrations of the machine under normal conditions. To justify it, it was also stated earlier with the mean spectral centroid that the anomalies took place in a lower frequency range, which can potentially yield longer periods between zero crossing and thus explain the lower zero crossing rate.

Metric	Anomalous	Normal
Spectral Centroid (Hz)	2945.91	3962.37
Spectral Rolloff (Hz)	6020.86	8502.82
Spectral Contrast	{ 14.62, 8.23, 12.23, 16.38, 14.08, 14.93, 20.17 }	{ 14.40, 8.30, 12.23, 16.40, 13.88, 15.50, 18.83 }
Zero Crossing Rate	0.047	0.070

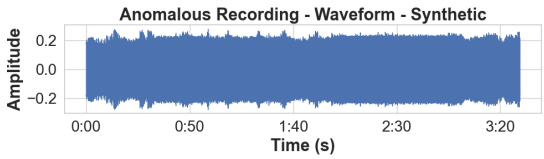
Table 5.1: Spectral Metrics for the Washing Machine Dataset

5.2.1.2 Synthetic Industrial Machine Dataset

Figure 5.4 shows the raw waveforms of the normal and anomalous recordings in the synthetic dataset. It is worth noting that compared to the washing machine dataset, these recordings are quite saturated, due to the amount of consistent background noise in the microphones. Nevertheless, similarly to the washing machine dataset, it can be seen here too, that the amplitude distribution of the normal recording in Figure 5.4a is much more compact compared to the amplitude distribution of the anomalous recording in Figure 5.4b. The normal recording exhibits the amplitude distribution of the pumped storage in the hydroelectric plant under normal operations. Thus the waveform is rather smooth without many changes in its amplitude except for shortly after the 5th minute of the recording which represents a changing operation mode. In contrast, the anomalous recording shows abrupt amplitude spikes throughout the recording that represent the high frequency sounds produced by the hits between the shovel and hammer, making the anomalies visible at plain sight.



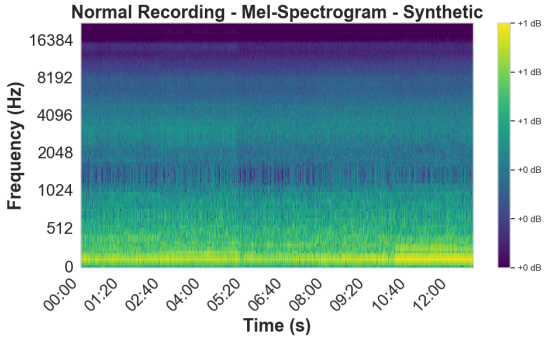
(a) Normal Synthetic Recording — Raw Waveform



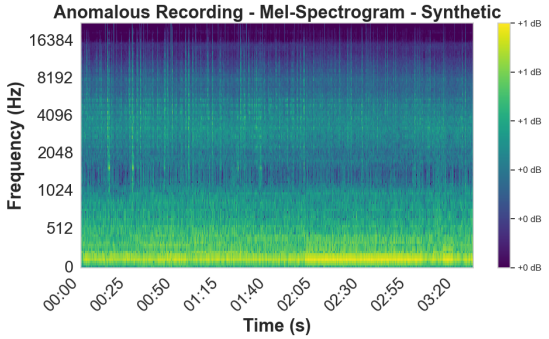
(b) Anomalous Synthetic Recording — Raw Waveform

Figure 5.4: Comparison of Anomalous and Normal Recordings Raw Waveforms

The Mel-spectograms of the synthetic anomaly and normal datasets are presented in Figure 5.5. Again, the Mel-spectogram of the normal recording 5.5a shows a smooth and constant energy distribution of the normal operating mode of the hydropower plant with even frequencies throughout the recording, whereas the anomalous Mel-spectogram in Figure 5.5b reveals multiple bursts of energy throughout the recording. This can be seen by the changing vertical energy stripes that represent the induced anomalies.



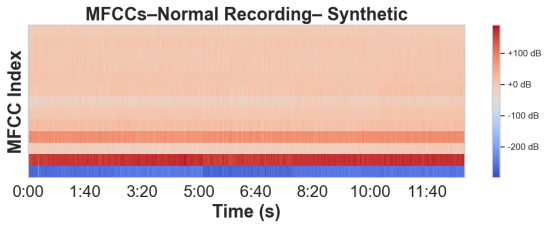
(a) Normal Synthetic Recording — Mel-Spectogram



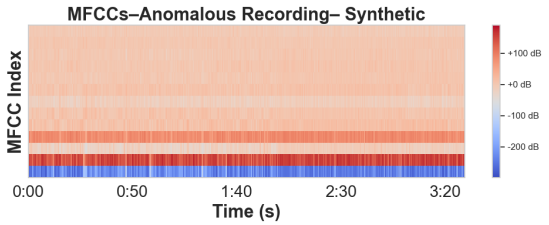
(b) Anomalous Synthetic Recording — Mel-Spectogram

Figure 5.5: Comparison of Anomalous and Normal Recordings Mel-Spectograms

Figure 5.6 shows the MFCC coefficients of the mel frequency for both recordings. The amplitude levels of the MFCC coefficients in the normal recording shown in Figure 5.6a remain relatively uniform across the entire recording. This suggests a stable spectral envelope with minimal short-term fluctuations compared to anomalous recording, where Figure 5.6b reveals more pronounced and sudden shifts in the MFCC coefficients over short time intervals, especially in coefficient 0 which represents the average log-energy of the signal. Overall though, the amplitude levels in both MFCC bands remain fairly similar between the two recordings, indicating similar sound intensity, which can be due to the background noise of the hydroelectric plant.



(a) Normal Recording MFCCs

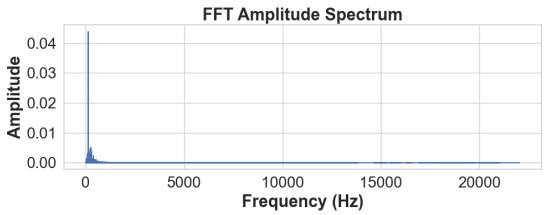


(b) Anomalous Recording MFCCs

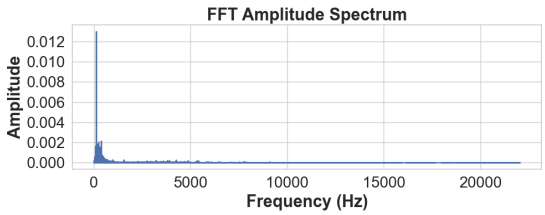
Figure 5.6: Comparison of Anomalous and Normal Recordings MFCCs

Furthermore, Figure 5.7 displays the FFT Amplitude spectrum, revealing the amplitude distribution across frequencies. The amplitude of the normal synthetic recording 5.7a shows a concentrated spectral profile in the lower frequency axis, while the anomalous recording in 5.7b exhibits a much wider energy distribution and distinct amplitude spikes in higher frequencies compared to the normal recording, corresponding to the high frequency acoustic signals introduced by the induced anomalies. This is yet another great

example of how anomalies can be manually spotted in a graph before model training, which reflects the substantial value of the EDA.



(a) Normal Synthetic Recording — FFT Amplitude



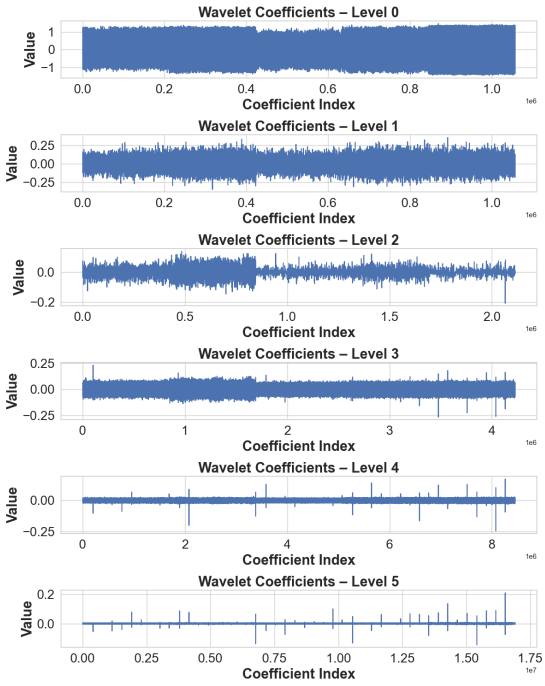
(b) Anomalous Synthetic Recording — FFT Amplitude

Figure 5.7: Comparison of Anomalous and Normal Recordings FFT Amplitudes

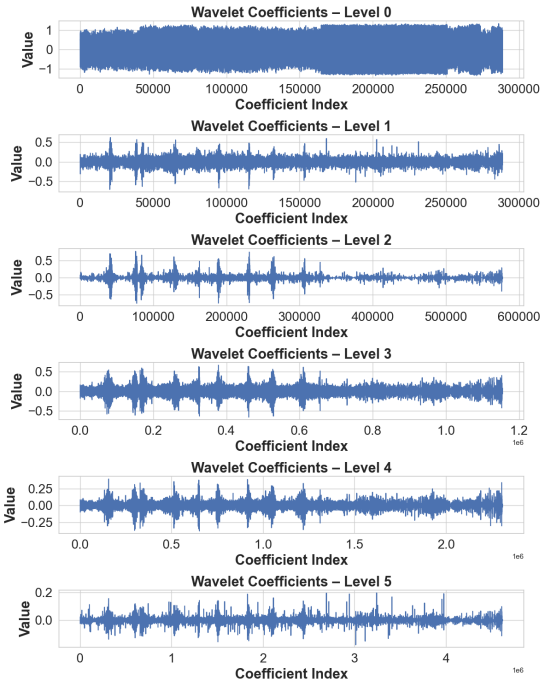
Lastly, Figure 5.8 compares the wavelet coefficients of the two recordings. It can be seen that the wavelet coefficients of the anomalous recording in Figure 5.8b exhibit several noticeable spikes, especially at mid to high decomposition levels. Conversely, the normal recording in Figure 5.8a shows relatively stable amplitudes within all coefficient levels and rather small values in higher coefficient levels compared to the anomalous recording. It is notable that at the higher levels (*e.g.,* Level 4 and Level 5), the wavelet coefficients remain relatively compact, indicating that the most energy is concentrated in the lower frequency band of the machine's operating noise. Thus, due to the high frequency of the induced anomalies, the anomalous recording presents sudden spikes and higher values in the higher coefficients.

5.2.1.3 Real Industrial Machine Dataset

So far, the EDA has been helpful in revealing the acoustic characteristics of each dataset, especially in finding indicators of anomalies or the anomalies themselves within the audio recordings. Most of the visualizations in the EDA help to understand the data and allow humans to spot irregularities manually, before training a ML model. This provides immense value for humans that could potentially detect anomalies by only looking at the datasets. The previous evaluations of the datasets have proved that the EDA is a step in the right direction to analyze data manually and not only supports anomaly detection but also provides crucial clues into the nature of the anomalies, especially when combined with domain knowledge. However, the EDA has only proved its usefulness using synthetically induced anomalies. Thus, an analysis of a real-world dataset needs to be done to validate the real capabilities of the notebook and to test the feasibility of spotting real-world and harder-to-spot anomalies.



(a) Normal Synthetic Recording — Wavelet coefficients



(b) Anomalous Synthetic Recording — Wavelet coefficients

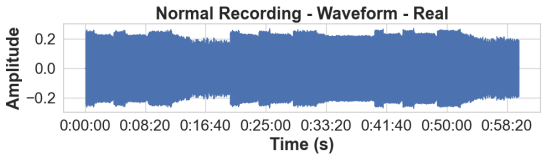
Figure 5.8: Comparison of Anomalous and Normal Recordings Wavelet Coefficients

In this subsection, the dataset of a real-world scenario of an operational hydropower plant, Rodundwerk II, is analyzed.

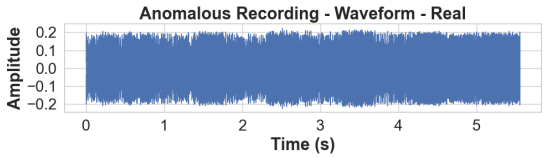
Figure 5.9 presents the results of the raw waveforms of the real recordings. For context, the normal recording consists of three separate recordings from three different microphones recording simultaneously and containing three different operation modes of the pumped storage. Also, three real anomalies were found in the recording which took place during the change of operating modes of the power plant but were removed manually in order to have a clean recording. The anomalous recording consists of the three anomalies concatenated into one dataset. Furthermore, similarly to the synthetic dataset, this dataset too has a high consistent background noise due to the high volume operations done by the pumped storage in the hydropower plant.

The operation cycles of the normal recording can be clearly seen in Figure 5.9a, where the three cycles of the operations are repeated three times. This figure exhibits a highly concentrated amplitude and compact waveform within each cycle. In contrast, the waveform of the anomalous recording shown in Figure 5.9b shows irregular acoustic amplitudes, with small and pronounced amplitude spikes. The lack of structure and

continuous variations in the anomalous waveform suggest a disruption in the regular operational pattern of the machine and thus hint towards anomalies or transient events. Of course, in this case, because the anomalous recording is short, the waveform does not necessarily present a clear representation of anomalies. Thus, at this stage it cannot be stated already that the irregularities are anomalies, which is why further acoustic features are presented.



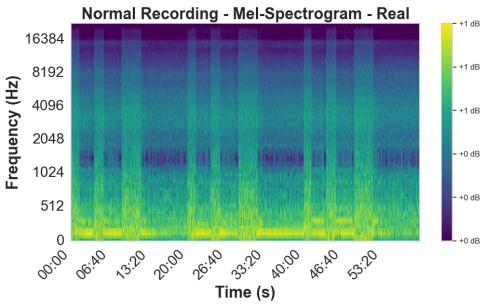
(a) Normal Real Recording — Raw Waveform



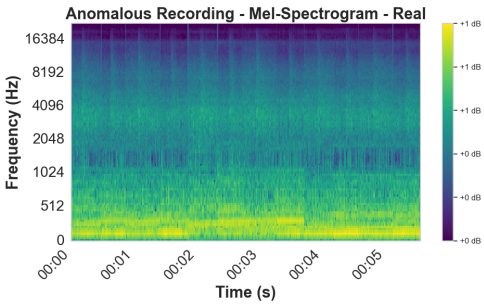
(b) Anomalous Real Recording — Raw Waveform

Figure 5.9: Comparison of Real Anomalous and Normal Recordings Raw Waveforms

The Mel-spectogram of the recordings in this dataset is presented in Figure 5.10. The three operating modes of the machine are clearly visible in Figure 5.10a where the operation modes are represented by the pronounced shift in energy stripes in the graph. It is important to note that even though the energy shifts are visible in the graph, this does not necessarily mean they would correspond to an anomaly. In this case for example, even when the machine changes from one operation mode to another, the changes are prolonged and the frequency distribution stays uniform. On the other hand, the frequency distribution of the anomalous recording is not as smooth, as can be seen in Figure 5.10b. It exhibits inconsistent frequency changes without any apparent pattern throughout the complete recording.



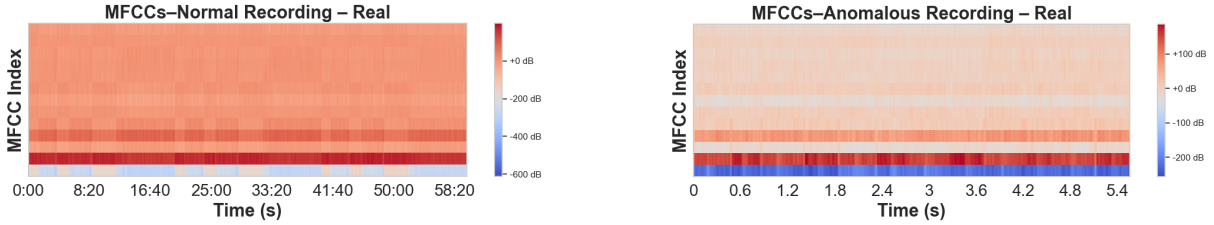
(a) Normal Real Recording — Mel-Spectogram



(b) Anomalous Real Recording — Mel-Spectogram

Figure 5.10: Comparison of Real Recordings Mel-Spectograms

The previous observations can also be seen in Figure 5.11 which shows the MFCCs of both recordings. Figure 5.11a shows consistent color bands in most coefficients in the normal recording indicating a stable spectral envelope. Apart from the evident the changes in operating modes in coefficient 0, each coefficient contains a stable amplitude level. In contrast, it can be seen from Figure 5.11a that the anomalous recording exhibits multiple short-term disruptions in the color bands of each coefficient with sudden amplitude deviations throughout the recording. Interestingly, coefficient 0 of the anomalous recording has much lower amplitude levels compared to the normal recording. This can indicate anomalies due to the fact that amplitude is inversely correlated to frequency, meaning, the anomalous recording has a higher frequency component which tends to suggest anomalies in the context of big hydropower plants.



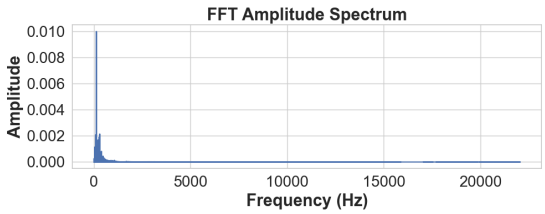
(a) Normal Real Recording — MFCCs

(b) Anomalous Real Recording — MFCCs

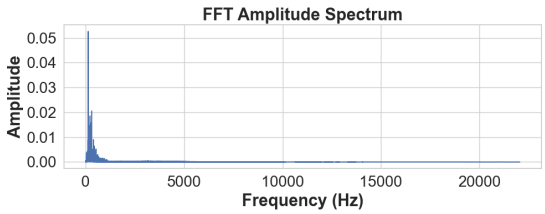
Figure 5.11: Comparison of Real Recordings MFCCs

Finally, Figure 5.12 shows a comparison of the FFT Amplitude spectrum of both recordings. The amplitude and frequency distribution shown in Figure 5.12b is significantly broader compared to the normal recording. The frequency distribution for the normal recording in Figure 5.12a is much more compact, representing normal operating conditions. Furthermore, the amplitude peak in the normal recording is 0.010dB compared to a higher peak in the anomalous recording of over 0.050dB. This is due to the longer audio file for the normal recording and more distributed acoustic signals in contrast to the short and noisy signals in the anomalous recording. Furthermore, this still shows that the anomalous recording contains a wider energy spectrum and more acoustic variations in the recordings which correspond to the real anomalies of the pumped machine.

The observations and visualizations from the real-world dataset in this EDA demonstrate that while anomalies are less immediately recognizable than in synthetic recordings, the combined insights from the acoustic waveforms, Mel-spectogram, MFCCs and FFT analyses still reveal meaningful deviations indicative of anomalous behavior. Unlike the



(a) Normal Real Recording — FFT Amplitude



(b) Anomalous Real Recording — FFT Amplitude

Figure 5.12: Comparison of Real Recordings FFT Amplitude

synthetic dataset, where anomalies were induced arbitrarily and were easily identifiable, real-world anomalies manifest themselves in a more subtle manner, such as irregular amplitude spikes, inconsistent frequency distributions, and abrupt spectral shifts. These results underline the value of EDA in providing a comprehensive and multifaceted perspective on acoustic signals, enabling the detection of anomalies even in complex and noisy environments.

5.2.2 Model Performance

This section presents the results of the model trainings on each dataset. The metrics discussed in Section 3.4 will be used to compare the results of the model trainings systematically.

5.2.2.1 Washing Machine Dataset

Table 5.2 shows the results of the models performance with the dataset from the washing machine. With a training time of 7.0023 seconds, the OC-SVM achieved the best performance on the washing machine dataset, reaching a high ROC AUC of 0.9659 and relatively high precision (0.8905) and recall (0.7679). Furthermore, Figure 5.13b shows that the model was able to successfully classify most of the anomalies (569 true negatives). This indicates that boundary learning anomaly detection is highly effective for this dataset where the anomalies are subtle low-frequency density sounds caused by the impacting shoes inside the machine.

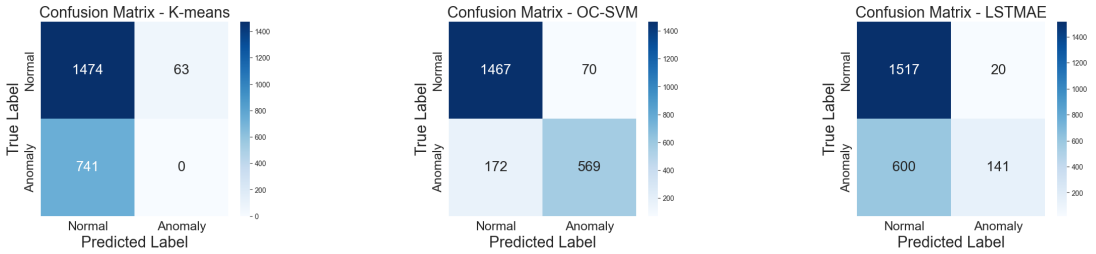
The K-means model was not able to detect any anomalies (5.13a), having a low ROC AUC of 0.5871 and 0.0000 of precision and recall. From its confusion matrix, it can also be seen that 741 normal frames were flagged as anomalous. This is most likely due to the fact that the model was not able to cluster the density variations in this specific dataset,

and indicating that clustering is not the best approach for the operational fluctuations and anomalies in the context of low-frequency anomalies in the washing machine dataset.

The LSTM AE achieved decent performance with a ROC AUC of 0.8885 but a low recall of 0.1943, and only detecting 141 anomalies (5.13c). This could be attributed to the fact that the AE struggled to reconstruct the subtle differences in the anomalies because they are similar to the normal training noise. Additionally, it needed significantly more training time than the other two models, approximately 48.31 seconds (6525%) more K-means and 42.04 seconds (601%) more than the OC-SVM. In terms of inference time, the OC-SVM needed the longest to classify an anomaly (2.0666s), but still remains as a small inference time in general.

Model	Train Time (s)	ROC AUC	Precision	Recall	F1 Score	Inference Time (s)
K-Means	0.7403	0.5871	0.0000	0.0000	0.0000	0.0270
OC-SVM	7.0023	0.9659	0.8905	0.7679	0.8246	1.9641
LSTM-AE	49.0513	0.8885	0.8521	0.1943	0.3165	0.3669

Table 5.2: Performance Comparison on the Washing Machine Dataset



(a) K-Means Confusion Matrix

(b) OC-SVM Confusion Matrix

(c) LSTM AE Confusion Matrix

Figure 5.13: Comparison of Confusion Matrices for Washing Machine Dataset

5.2.2.2 Synthetic Industrial Machine Dataset

The differences between anomalies and normal operating sound in this dataset are large, allowing the models to capture the differences in a clearer way. The results are summarized in Table 5.3.

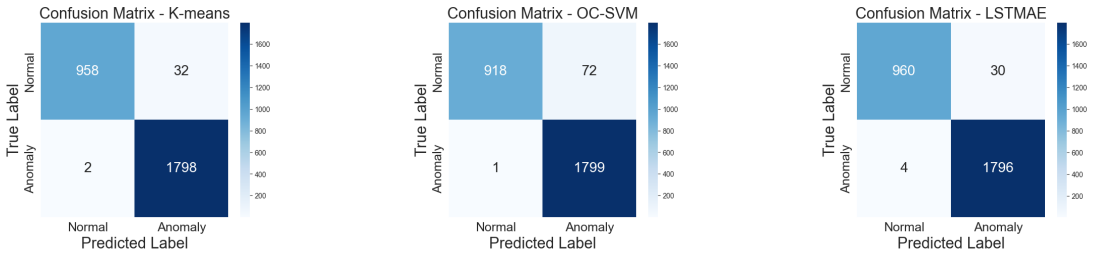
The K-means model achieved near perfect ROC AUC (0.9974) as well as recall (0.9989) in a short training time of 0.3700 seconds. Furthermore, the OC-SVM had an almost flawless recall of 0.9994 and almost perfect ROC AUC of 0.9977, demonstrating again the robustness of boundary decision for anomaly detection. In addition, the training time of the model was still relatively short at 2.8284 seconds. The LSTM AE had the best

ROC AUC (0.9995) of the three models and very similar results for precision (0.9814) and recall (0.9972), yet having the slowest training time of 32.8777 seconds. All the three models had excellent performance and only slight performance differences, but in general, taking into account all the evaluation metrics, the OC-SVM still remains one of the strongest models, despite having the slowest inference time (2.1469 seconds).

This dataset shows the capability of each model to learn high-frequency patterns and their true potential for detecting anomalies. Nevertheless, it would not be wise to assume that anomalies always contain high-frequency frames, as this does not always happen in the real world. This leads us to the next dataset, the real-world scenario for anomaly detection.

Model	Train Time (s)	ROC AUC	Precision	Recall	F1 Score	Inference Time (s)
K-Means	0.3700	0.9974	0.9825	0.9989	0.9906	0.0275
OC-SVM	2.8284	0.9977	0.9615	0.9994	0.9801	2.1469
LSTM-AE	32.8777	0.9995	0.9814	0.9972	0.9893	0.4765

Table 5.3: Performance Comparison on the Synthetic Industrial Machine Dataset



(a) K-Means Confusion Matrix

(b) OC-SVM Confusion Matrix

(c) LSTM AE Confusion Matrix

Figure 5.14: Comparison of Confusion Matrices for Synthetic Dataset

5.2.2.3 Real Industrial Machine Dataset

Finally, the real anomaly dataset represents a real-world scenario where the models could actually be used in production. Table 5.4 summarizes the results of each models training for this dataset. Due to the considerably larger dataset, all models required more time for training. K-means was the fastest at 5.2506 seconds, followed by the OC-SVM at 55.5363 seconds and the LSTM AE at 145.2456 seconds.

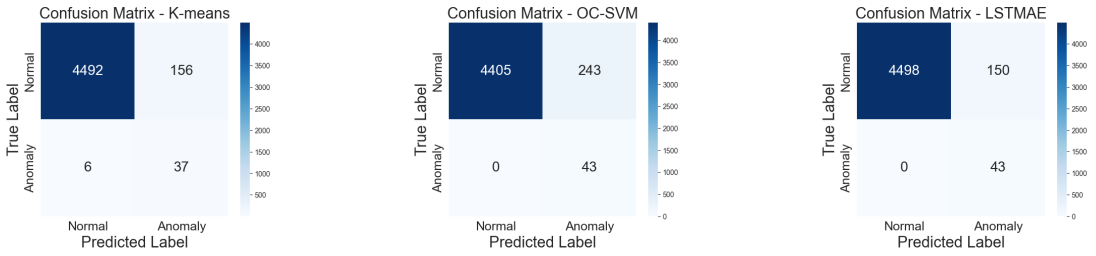
Both the OC-SVM and AE had great detection performance with ROC AUC of 0.9976 and 0.9974, respectively, and perfect recall of 1.0000, detecting all the anomalies in the dataset. K-means remained behind them but still exhibited great detection performance

with a ROC AUC of 0.9618 and 0.8605 recall. Additionally, its inference time (0.0430 seconds) is much shorter compared to the other models, especially to the OC-SVM (12.1680 seconds).

Model	Train Time (s)	ROC AUC	Precision	Recall	F1 Score	Inference Time (s)
K-Means	5.2506	0.9618	0.1917	0.8605	0.3136	0.0430
OC-SVM	55.5363	0.9976	0.1503	1.0000	0.2614	12.1680
LSTM-AE	145.2456	0.9974	0.2194	1.0000	0.3598	0.8667

Table 5.4: Performance Comparison on the Real Industrial Machine Dataset

In general, the precision of the models was low though; 0.1917 for K-means, 0.1503 for OC-SVM and 0.2194 for the LSTM AE. This can also be seen in Figure 5.15, where K-means had 156 (5.15a) falsely detected anomalies, the OC-SVM 243 (5.15b), and the LSTM AE 150 (5.15c). So in general, in can be seen that the models achieved high recall (especially the OC-SVM and LSTM AE) but at the cost of precision.



(a) K-Means Confusion Matrix

(b) OC-SVM Confusion Matrix

(c) LSTM AE Confusion Matrix

Figure 5.15: Comparison of Confusion Matrices for the Real Industrial Dataset

5.3 Discussion & Comparative Analysis

This section brings together the findings from the EDA and model evaluations, providing a comprehensive review of the performance of both the EDA and the ML models for detecting anomalies in a high-altitude hydroelectric plant. The analysis reveals critical insights into the models efficacy, computational advantages, and the role of the EDA in bridging acoustic theory with anomaly detection, a core contribution of this work. Furthermore, it establishes the significance of the findings as a further step into improving anomaly detection systems which will allow industrial companies, or in this case a high-altitude hydropower plant, to improve their PdM strategies and detect mal-functions to avoid overhead costs for unnecessary maintenance. Ultimately, this thesis presents a

comparative evaluation of the best found approaches for building an effective ADS in order to detect anomalies in industrial settings.

The EDA served as a foundational pillar for providing a thorough understanding of the datasets by visualizing the unique insights into the acoustic properties of each of them.

For the washing machine dataset, the identification of low-frequency anomalies through the Mel- and STFT-spectograms and the spectral centroid analysis 5.2 explained OC-SVM's superior performance (ROC AUC: 0.9659), as boundary-based methods excel at detecting subtle deviations in such distributions. Conversely, K-means' failure (precision: 0.000) aligned with the EDA's revelation of overlapping density patterns between normal and anomalous clusters, underscoring the limitations of centroid-based clustering in low-separation scenarios. Thus, the preliminary findings in the EDA can already give a decent idea of what is behind the recordings and what can be expected from the model trainings.

Furthermore, the EDA's clear visualization of high-frequency anomalies (Figure 5.7) in the synthetic dataset justified the near-perfect performance of all models (ROC AUC >0.9974) and anomalous energy shifts were also found in the wavelet coefficients of the audios, reinforcing the visibility of engineered anomalies introduced in the environment.

In terms of real-world scenarios, the EDA demonstrated a dual role in its usability: while Mel-spectrograms and MFCCs revealed subtle anomalies (Figure 5.10), they also highlighted challenges such as persistent background noise and overlapping spectral patterns. These are factors that contributed to the models degraded precision (0.15–0.21) despite high recall (0.86–1.00). This duality reinforces the thesis's emphasis on EDA being not only a diagnostic tool for anomaly characterization but also as a strategic guide for addressing real-world challenges, providing robustness to the proposed PdM framework.

The chosen evaluation metrics and results found in the performance comparison of the trained models - K-means, OC-SVM and LSTM AE - revealed significant trade-offs in detection accuracy and computational efficiency.

In the washing machine dataset, the OC-SVM achieved a good balance between high ROC AUC (0.9659), precision (0.8905), and recall (0.7679), aligning well with the EDA insights that indicated clear but subtle acoustic deviations. The K-Means model however, struggled to effectively cluster the small differences in density variations and ended up having poor anomaly detection performance (ROC AUC: 0.5871), not being able to

identify any of the anomalies (precision and recall: 0.0000). Furthermore, even though the LSTM AE reached decent ROC AUC (0.8885), the low recall (0.1943) suggests that the AE could not reconstruct the anomalies easily.

The synthetic dataset showed, that in optimal controlled environments, all models can perform extremely well. Although the anomalies were more clearly visible, as seen in the EDA, even lightweight models could achieve near-perfect model performance (ROC AUC>0.9974). However, in the real industrial machine dataset, where the length of the video, noise and complexity of the acoustic signals were more complex, it proved to be more challenging to detect anomalies due to broader energy distributions. This also leads to the observation that further hyperparameter tuning or hybrid modeling could potentially enhance precision without compromising sensitivity. In the end, the systematic comparison of the models performance provided in this thesis serves as a baseline for future works or companies to compare the distinct trade-offs that each model poses. In the synthetic environment, where all models had similar performance scores (ROC AUC >0.9974, Precision>0.9615, Recall>0.9972 and F1Score>0.9801) the training and inference time could be a deciding factor for a hydropower energy plant that needs real time and cheap computational models compared to another manufacturer who can allow himself more time but needs the models to be more accurate. The potential trade-offs in choosing a model can be summarized in the following points:

- **OC-SVM (Boundary Leaning):** The model dominated in scenarios that required sensitivity to subtle acoustic deviations, achieving the highest ROC AUC (0.9659-0.9976) in two datasets. In addition, its computational efficiency (training time: 2.8–55.5s) aligns with the thesis objectives of developing deployable solutions for industrial settings.

- **K-means (Clustering):** Performed well under synthetic conditions (ROC AUC: 0.9974) but failed with overlapping distributions (washing machine ROC AUC: 0.5871), underlining its niche applicability depending on the dataset. However, its fast inference (0.027–0.043s) could support real-time anomaly detection, but with limited robustness.

- **LSTM AE (Temporal pattern recognition):** Demonstrates resilience to noise in real-world data (ROC AUC: 0.9974) but suffers from high training overhead (145s). While the model can capture temporal dependencies which aligns with the goals of

this thesis to focus on robust ADSs, the models complexity and training overheads could hinder its scalability.

In summary, the proposed hybrid approach for an anomaly detection system highlights the significance of not only training a model to predict anomalies but also the process of understanding the audio samples and seeing why some models work better than others for a specific dataset. Although the LSTM AE was the most sophisticated model tested and probably the most common one used in the industry, it was possible to show that cheap computational models like the OC-SVM can perform even better than the LSTM AE in certain scenarios. The OC-SVM is proof that a hybrid approach for anomaly detection can be more beneficial to an analysis given its excellent performance and computational efficiency. In that sense, this framework successfully analyzes the applicability of acoustic sensing to detect anomalies in hydropower plants and achieved good results in both controlled environments and real-world scenarios. Furthermore, it also proved that a hybrid approach, where multiple models work alongside to detect anomalies, can bring more value to the final results and provide a collective impact on the performance of the anomaly detection system. This marks a huge step for the industry, given that anomaly detection systems can use lightweight computational models using real-time analysis to detect anomalies. This also reduces the complexity barrier for companies to develop their own solutions for anomaly detection systems, making a headway for optimized PdM in high-cost industrial environments like hydropower plants, where minutes of downtime can incur significant losses.

Chapter 6

Conclusions & Future work

6.1 Summary

This thesis explored the feasibility and efficacy of using acoustic sensing for anomaly detection in a high-altitude hydropower plant. The proposed framework was tested on a range of datasets, from synthetic to real-world acoustic recordings from an operational high-altitude power plant. Through the EDA and subsequent model training, it was shown that the anomalies could be successfully identified even in challenging environments such as in the real-world scenario.

A standout result was the performance of the OC-SVM, which outperformed K-Means and LSTM AE under both controlled and real-world conditions. This shows the effectiveness of boundary decision algorithms for anomaly detection, even with minimal training and inference times. In that sense, a contribution of this thesis is that simpler, computationally lightweight algorithms can also be used to detect anomalies and potentially outperform more sophisticated ones under certain conditions.

The LSTM AE also demonstrated good performance, especially in the synthetic and real industrial dataset, but needed much longer training times compared to the other models. K-Means showed strong performance in the synthetic and real industrial dataset too, but lacked robustness in more ambiguous scenarios such as the washing machine dataset.

The implementation and results of this thesis can be publicly accessed on Github: https://github.com/nico-keller/acoustic-sensing-anomaly-detection.

6.2 Final Considerations

The research carried out in this thesis validated the use of acoustic sensing for anomaly detection in a high-altitude hydropower plant, validating its utility in predictive mainte-

nance systems. The integration of the EDA into the framework served as an essential component not only for preprocessing the dataset, but also for revealing insights about their acoustic patterns and offering a holistic perspective on the datasets. Techniques such as visualizing the raw waveforms, Mel- and STFT-spectogramms, MFCCs, and FFT amplitude analyses proved to be highly relevant for identifying anomalies even before any machine learning was applied.

Through a systematic comparative evaluation of different ML models, it became clear once again that there is no universal best model for anomaly detection. In this thesis, the OC-SVM showed the best performance and computational efficiency all round. The LSTM AE was also effective for datasets with clearer temporal patterns, while the K-Means was most suitable for high-contrast synthetic anomalies but less so for noticing small subtleties as in the washing machine dataset. This can be seen as an opportunity for hydropower plants to integrate multiple ML models into their pipelines to detect anomalies and not having to rely on only one model.

The challenges encountered in this thesis are the scarcity and imbalance of anomaly data and the inherent noise in real-world recordings. On the one hand, having only a handful of anomalies to test the models limits their potential and precision, especially in the case of the real dataset. In addition, minor inconsistencies in data acquisition, such as background noise, also left a noticeable mark on the datasets. Although the preprocessing steps in the notebooks use noise reduction techniques to hinder the effects of background noise, it was still evident in the EDA. These findings highlight the importance of clean data acquisition and robust processing techniques to overcome these issues.

Another relevant finding is the demonstrated value of a hybrid approach for anomaly detection systems. By integrating a thorough EDA and multiple ML models with complementary strengths, the framework achieved more reliable results and error-tolerant detection capabilities, which is crucial for high-altitude hydropower plant. This further supports the idea that hybrid anomaly detection systems, particularly those that include lightweight computational models, can be effective and scalable in real-world scenarios for industrial deployment.

In conclusion, this thesis established a solid foundation for acoustic sensing in predictive maintenance. The findings demonstrate the high effectiveness and potential of classical ML algorithms in real-world scenarios and highlighted the value that a hybrid approach yields in such an anomaly detection system. Thus, this proposed framework serves as an effective and practical foundation for real-time anomaly detection in high-

altitude hydropower plants with the objective of improving predictive maintenance, where operational downtime translates into financial losses.

6.3 Future Work

Given that one of the biggest challenges encountered in this thesis was the scarcity of data, future research should focus on expanding the datasets to include a broader variety of operational scenarios and to avoid imbalanced datasets. This would enhance model training and validation, allowing for more generalizable results.

Another primary area for improvement would be to integrate the findings of the EDA to dynamically adjust the model hyperparameter and achieve even better training results by automatically adapting the models to each dataset. For instance, a model could adapt its hyperparameters depending on the noise level or frequency distributions of a given dataset revealed in the EDA. This would allow the anomaly detection system to automatically adapt itself to the dataset and thus increase performance and precision without human intervention. This could also pave the way for universal systems that could be introduced in any industrial setting for real-time monitoring.

Furthermore, noise-resilient approaches should be further investigated so that noisy environments are no longer limited when training ML models. Techniques such as adaptive filtering, transfer learning, or data augmentation could potentially improve real-world performance and lead to more robust anomaly detection frameworks.

Bibliography

[1] Zrar Kh. Abdul and Abdulbasit K. Al-Talabani. "Mel Frequency Cepstral Coefficient and its Applications: A Review". In: *IEEE Access* 10 (2022), pp. 122136–122158. DOI: 10.1109/ACCESS.2022.3223444.

[2] Raghdah Adnan Abdulrazzq, Nisreen Mustafa Sajid, and Marwan Sabah Hasan. "Artificial intelligence-driven predictive maintenance in IoT systems". In: *South Florida Journal of Development* 5.12 (2024), e4781. DOI: 10.46932/sfjdv5n12-030.

[3] Mounia Achouch, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhouib, Hussein Ibrahim, and Mehdi Adda. "On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges". In: *Applied Sciences* 12.16 (2022). ISSN: 2076-3417. DOI: 10.3390/app12168081.

[4] Naman Agrawal. *Decoding the Symphony of Sound: Audio Signal Processing for Musical Engineering*. Accessed: 2025-02-13. 2023. URL: https://towardsdatascience.com/decoding-the-symphony-of-sound-audio-signal-processing-for-musical-engineering-c66f09a4d0f5/.

[5] Hyojung Ahn and Inchoon Yeo. "Deep-Learning-Based Approach to Anomaly Detection Techniques for Large Acoustic Data in Machine Operation". In: *Sensors* 21.5446 (2021). DOI: 10.3390/s21165446.

[6] Amazon Web Services, Inc. *What is Predictive Maintenance?* Accessed: 2025-02-13. 2025. URL: https://aws.amazon.com/what-is/predictive-maintenance/.

[7] Chantal Amrhein and Barry Haddow. "Don't Discard Fixed-Window Audio Segmentation in Speech-to-Text Translation". In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. 2022, pp. 203–219.

[8] Artesis. *Key Predictive Maintenance Trends for 2023 and Beyond*. Accessed: 2025-02-13. 2021. URL: https://artesis.com/key-predictive-maintenance-trends-for-2023-and-beyond/.

[9] NTi Audio. *Fast Fourier Transform (FFT)*. Accessed: 2025-02-15. 2025. URL: https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft.

[10] Eldho Babu, Jebin Francis, Esther Thomas, Rahul Cherian, and Sudarsana S. Sunandhan. "Review on Various Signal Processing Techniques for Predictive Maintenance". In: *2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and Its Control (PARC)*. IEEE. 2022, pp. 1–8. DOI: 10.1109/PARC52418.2022.9726618.

[11] Baris Bayram, Taha Berkay Duman, and Gökhan Ince. "Real time detection of acoustic anomalies in industrial processes using sequential autoencoders". In: *Expert Systems* 38 (2021). DOI: 10.1111/exsy.12564.

[12] E. O. Brigham and R. E. Morrow. "The fast Fourier transform". In: *IEEE Spectrum* 4.12 (1967), pp. 63–70. DOI: 10.1109/MSPEC.1967.5217220.

[13] Guangyi Chen, Wen-Fang Xie, and Zhao Yongjia. "Wavelet-based denoising: A brief review". In: 2013, pp. 570–574. ISBN: 978-1-4673-6248-1. DOI: 10.1109/ICICIP.2013.6568140.

[14] Zeki Murat Çınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0". In: *Sustainability* 12.19 (2020). ISSN: 2071-1050. DOI: 10.3390/su12198211.

[15] Cloud Software Group, Inc. *What is Predictive Maintenance?* Accessed: 2025-02-13. 2025. URL: https://www.spotfire.com/glossary/what-is-predictive-maintenance.

[16] W.T. Cochran, J.W. Cooley, D.L. Favin, H.D. Helms, R.A. Kaenel, W.W. Lang, G.C. Maling, D.E. Nelson, C.M. Rader, and P.D. Welch. "What is the fast Fourier transform?" In: *Proceedings of the IEEE* 55.10 (1967), pp. 1664–1674. DOI: 10.1109/PROC.1967.5957.

[17] Codecademy Team. *Normalization*. Accessed: 2025-02-13. 2025. URL: https://www.codecademy.com/article/normalization.

[18] Gabriel Coelho, Luís Miguel Matos, Pedro José Pereira, André Ferreira, André Pilastri, and Paulo Cortez. "Deep Autoencoders for Acoustic Anomaly Detection: Experiments with Working Machine and In-Vehicle Audio". In: *RepositóriUM* (2022).

[19] Jovani Dalzochio, Rafael Kunst, Edison Pignaton, Alecio Binotto, Srijnan Sanyal, Jose Favilla, and Jorge Barbosa. "Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges". In: *Computers in Industry* 123 (2020), p. 103298. ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2020.103298.

[20] S. Daultrey. *Principal Components Analysis*. 1976. ISBN: 9780902246560.

[21] Emmanuel Deruty. *Intuitive Understanding of MFCCs*. Accessed: 2025-02-13. 2022. URL: https://medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779.

[22] Google Developers. *Accuracy, Precision, and Recall - Machine Learning Crash Course*. Accessed: 2025-03-04. 2025. URL: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall.

[23] Emanuele Di Fiore, Antonino Ferraro, Antonio Galli, Vincenzo Moscato, and Giancarlo Sperlì. "An anomalous sound detection methodology for predictive maintenance". In: *Expert Systems with Applications* 209 (2022), p. 118324. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.118324.

[24] Ray Milton Dolby. "An Audio Noise Reduction System". In: *Journal of The Audio Engineering Society* 15 (1967), pp. 383–388.

[25] Taha Berkay Duman, Baris Bayram, and Gökhan Ince. "Acoustic Anomaly Detection Using Convolutional Autoencoders in Industrial Processes". In: *Advances in Intelligent Systems and Computing*. 2020. DOI: 10.1007/978-3-030-20055-8_41.

[26] École Polytechnique Fédérale de Lausanne. *Signal Processing (COM-202)*. Accessed: 2025-02-13. 2025. URL: https://edu.epfl.ch/coursebook/en/signal-processing-COM-202.

[27] Edge Impulse. *Audio Feature Extraction*. Accessed: 2025-02-13. 2025. URL: https://docs.edgeimpulse.com/docs/concepts/data-engineering/audio-feature-extraction.

[28] Issam El Naqa and Martin J. Murphy. "What Is Machine Learning?" In: *Machine Learning in Radiation Oncology: Theory and Applications*. 2015, pp. 3–11. ISBN: 978-3-319-18305-3. DOI: 10.1007/978-3-319-18305-3_1.

[29] Sarvarbek Erniyazov, Yongmin Kim, M. Jaleel, and Chang Gyoon Lim. "Comprehensive Analysis and Improved Techniques for Anomaly Detection in Time Series Data with Autoencoder Models". In: *International Journal on Advanced Science, Engineering and Information Technology* 14.6 (2024), pp. 1861–1867. DOI: 10.18517/ijaseit.14.6.20451.

[30] Mattia Fanan, Claudio Baron, Ruggero Carli, Marc-Aurèle Divernois, Jean-Christophe Marongiu, and Gian Antonio Susto. "Anomaly Detection for Hydroelectric Power Plants: a Machine Learning-based Approach". In: 2023, pp. 1–6. DOI: 10.1109/INDIN51400.2023.10218027.

[31] Zhiyan Feng, Zengtao Zhao, Honghao Chen, Bowen Dou, and Liehao Hu. "Power Plant Production Equipment Sound Recognition Method Combined with Attention Mechanism". In: 2022, pp. 185–188. DOI: 10.1109/ICPICS55264.2022.9873701.

[32] A. Ferraro, A. Galli, V.L. Gatta, V. Moscato, M. Postiglione, G. Sperlì, and F. Moscato. "Unsupervised Anomaly Detection in Predictive Maintenance using Sound Data". In: *CEUR Workshop Proceedings* 3478 (2023), pp. 449–458.

[33] GeeksforGeeks. *Least Mean Squares Filter in Signal Processing*. Accessed: 2025-02-13. 2024. URL: https://www.geeksforgeeks.org/least-mean-squares-filter-in-signal-processing/.

[34] GeeksforGeeks. *Mel-Frequency Cepstral Coefficients (MFCC) for Speech Recognition*. Accessed: 2025-02-13. 2024. URL: https://www.geeksforgeeks.org/mel-frequency-cepstral-coefficients-mfcc-for-speech-recognition/.

[35] GeeksforGeeks. *Preprocessing the Audio Dataset*. Accessed: 2025-02-13. 2023. URL: https://www.geeksforgeeks.org/preprocessing-the-audio-dataset/.

[36] GeeksforGeeks. *What is Digital Signal Processing?* Accessed: 2025-02-13. 2024. URL: https://www.geeksforgeeks.org/what-is-digital-signal-processing/.

[37] Cihun-Siyong Alex Gong, Huang-Chang Lee, Yu-Chieh Chuang, Tien-Hua Li, Chih-Hui Simon Su, Lung-Hsien Huang, Chih-Wei Hsu, Yih-Shiou Hwang, Jiann-Der Lee, and Chih-Hsiung Chang. "Design and Implementation of Acoustic Sensing System for Online Early Fault Detection in Industrial Fans". In: *Journal of Sensors* 2018 (2018), pp. 1–15. DOI: 10.1155/2018/4105208.

[38] Deepam Goyal and B.S. Pabla. "Condition based maintenance of machine tools—A review". In: *CIRP Journal of Manufacturing Science and Technology* 10 (2015), pp. 24–35. ISSN: 1755-5817. DOI: https://doi.org/10.1016/j.cirpj.2015.05.004.

[39] Jian Guan, Feiyang Xiao, Youde Liu, Qiaoxi Zhu, and Wenwu Wang. "Anomalous Sound Detection Using Audio Representation with Machine ID Based Contrastive Learning Pretraining". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096054.

[40] Gökşel Gündüz. *Fundamental Terms of Signal Processing*. https://medium.com/@goksselgunduz/fundamental-terms-of-signal-processing-2826a1b5543d. Accessed: 2025-04-12. 2021.

[41] S. Haykin. *Adaptive Filter Theory: International Edition*. 2014. ISBN: 9780273775720.

[42] Guosheng Huang, Jinchuan Chen, and Lei Liu. "One-Class SVM Model-Based Tunnel Personnel Safety Detection Technology". In: *Applied Sciences* 13.3 (2023). ISSN: 2076-3417. DOI: 10.3390/app13031734.

[43] Valentin Huber and Stefan Krummenacher. "A Case Study of Acoustic-based Anomaly Detection for Industrial Predictive Maintenance". In: *School of Computer Science, University of St. Gallen* (2023).

[44] Hugging Face. *Introduction to Audio Data*. Accessed: 2025-02-13. 2025. URL: https://huggingface.co/learn/audio-course/en/chapter1/audio_data.

[45] IBM. *Supervised vs. Unsupervised Learning: Key Differences and Examples*. Accessed: 2025-02-16. 2025. URL: https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning.

[46] IBM Corporation. *What is Predictive Maintenance?* Accessed: 2025-02-13. 2023. URL: https://www.ibm.com/think/topics/predictive-maintenance.

[47] Infraspeak Team. *Is Predictive Maintenance Really Cost-Effective?* Accessed: 2025-02-13. 2021. URL: https://blog.infraspeak.com/predictive-maintenance-cost-effective/.

[48] Aryan Jadon, Avinash Patil, and Shruti Jadon. "A Comprehensive Survey of Regression-Based Loss Functions for Time Series Forecasting". In: *Data Management, Analytics and Innovation*. 2024, pp. 117–147. ISBN: 978-981-97-3245-6.

[49] Andrew Jardine, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance". In: *Mechanical Systems and Signal Processing* 20 (2006), pp. 1483–1510. DOI: 10.1016/j.ymssp.2005.09.012.

[50] Ian Jolliffe. "Principal Component Analysis". In: *Encyclopedia of Statistics in Behavioral Science*. 2005. ISBN: 9780470013199. DOI: https://doi.org/10.1002/0470013192.bsa501.

[51] Gbanaibolou Jombo and Yu Zhang. "Acoustic-Based Machine Condition Monitoring—Methods and Challenges". In: *Eng* 4.1 (2023), pp. 47–79. ISSN: 2673-4117. DOI: 10.3390/eng4010004.

[52] S.A. Keenan, O. Carrillo, and H. Casseres. "Electroencephalography". In: *Encyclopedia of Sleep*. 2013, pp. 66–70. ISBN: 978-0-12-378611-1. DOI: https://doi.org/10.1016/B978-0-12-378610-4.00140-6.

[53] M. Khanjari, A. Azarfar, M. H. Abardeh, and et al. "Anomalous Sound Detection for Machine Condition Monitoring Using 3D Tensor Representation of Sound and 3D Deep Convolutional Neural Network". In: *Multimedia Tools and Applications* 83 (2024), pp. 44101–44119. DOI: 10.1007/s11042-023-17043-9.

[54] Barış Kopuz and Nihan Kahraman. "Comparison and Analysis of LSTM-Capsule Networks and 3DConv-LSTM Autoencoder in Ambient Anomaly Detection". In: (2024), pp. 1–5. DOI: 10.1109/eleco64362.2024.10847263.

[55] Kun-Lun Li, Hou-Kuan Huang, Sheng-Feng Tian, and Wei Xu. "Improving one-class SVM for anomaly detection". In: *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*. Vol. 5. 2003, 3077–3081 Vol.5. DOI: 10.1109/ICMLC.2003.1260106.

[56] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[57] Claudia Linnhoff-Popien, Steffen Illium, Fabian Ritz, and Robert Müller. "Acoustic Anomaly Detection for Machine Sounds based on Image Transfer Learning". In: *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (Volume 2)*. 2021, pp. 49–56.

[58] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[59] Bin Lu, David B. Durocher, and Peter Stemper. "Predictive maintenance techniques". In: *IEEE Industry Applications Magazine* 15.6 (2009), pp. 52–60. DOI: 10.1109/MIAS.2009.934444.

[60] W. Ma, M. Ma, Z. Zhang, J. Ma, R. Zhang, and J. Wang. "Anomaly Detection of Mountain Photovoltaic Power Plant Based on Spectral Clustering". In: *IEEE Journal of Photovoltaics* 13.4 (2023), pp. 621–631. DOI: 10.1109/JPHOTOV.2023.3267222.

[61] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. 3rd. 2008.

[62] Sílvia Margarit Jaile. "Anomaly detection using audio signals". PhD thesis. UPC, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, Departament de Teoria del Senyal i Comunicacions, 2020.

[63] C. Mateo and J. A. Talavera. "Bridging the gap between the short-time Fourier transform (STFT), wavelets, the constant-Q transform and multi-resolution STFT". In: *Signal, Image and Video Processing* 14 (2020), pp. 1535–1543. DOI: 10.1007/s11760-020-01701-8.

[64] Muhammad Aiman Md Zuki, Nazlena Mohamad Ali, and Jun Kit Chaw. "Reinforcement learning: methods and recent applications". In: *Journal of Information System and Technology Management* 9.36 (2024), pp. 67–89. DOI: 10.35631/jistm.936005.

[65] M. Meire and P. Karsmakers. "Comparison of deep autoencoder architectures for real-time acoustic based anomaly detection in assets". In: *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. Vol. 2. 2019, pp. 786–790. DOI: 10.1109/IDAACS.2019.8924301.

[66] Umberto Michelucci. *An Introduction to Autoencoders*. 2022. arXiv: 2201.03898 [cs.LG]. URL: https://arxiv.org/abs/2201.03898.

[67] R. Keith Mobley. "1 - Impact of Maintenance". In: *An Introduction to Predictive Maintenance (Second Edition)*. Second Edition. 2002, pp. 1–22. ISBN: 978-0-7506-7531-4. DOI: https://doi.org/10.1016/B978-075067531-4/50001-4.

[68] Mohammadamin Moradi, Shirin Panahi, Erik M. Bollt, and Ying–Cheng Lai. "Kolmogorov-Arnold Network Autoencoders". In: (2024). DOI: 10.48550/arxiv.2410.02077.

[69] Samreen Naeem, Aqib Ali, Sania Anam, and Munawar Ahmed. "An Unsupervised Machine Learning Algorithms: Comprehensive Review". In: *IJCDS Journal* 13 (2023), pp. 911–921. DOI: 10.12785/ijcds/130172.

[70] Vladimir Nasteski. "An overview of the supervised machine learning methods". In: *HORIZONS.B* 4 (2017), pp. 51–62. DOI: 10.20544/HORIZONS.B.04.1.17.P05.

[71] E. C. Nunes. "Anomalous sound detection with machine learning: A systematic review". In: *arXiv preprint* (2021). DOI: 10.48550/arXiv.2102.07820.

[72] Henri J. Nussbaumer. "The Fast Fourier Transform". In: *Fast Fourier Transform and Convolution Algorithms*. 1982, pp. 80–111. ISBN: 978-3-642-81897-4. DOI: 10.1007/978-3-642-81897-4_4.

[73] Antero Ollila and Markku Malmipuro. "Maintenance has a role in quality". In: *The TQM Magazine* 11.1 (1999), pp. 17–21. DOI: 10.1108/09544789910246589.

[74] Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni, and Jelena Loncarski. "Machine Learning approach for Predictive Maintenance in Industry 4.0". In: *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. 2018, pp. 1–6. DOI: 10.1109/MESA.2018.8449150.

[75] Yumin Peng, Zengtao Zhao, Fanqi Huang, and Liehao Hu. "Anomalous sound detection for hydroelectric plant equipment based on self-encoder and weakly supervised learning". In: *International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2023)*. Vol. 12645. 2023, 126452M, p. 126452M. DOI: 10.1117/12.2681050.

[76] Matthias Plaue. "Supervised machine learning". In: *Data Science: An Introduction to Statistics and Machine Learning*. 2023, pp. 185–248. ISBN: 978-3-662-67882-4. DOI: 10.1007/978-3-662-67882-4_6.

[77] Petru Potrimba. "What is an Autoencoder?" In: *Roboflow Blog* (2022).

[78] Vibration Research. *Fast Fourier Transform (FFT) Analysis*. Accessed: 2025-02-15. 2025. URL: https://vibrationresearch.com/blog/fast-fourier-transform-fft-analysis/.

[79] Mark Richardson. "Principal component analysis". In: *URL: http://people. maths. ox. ac. uk/richardsonm/SignalProcPCA. pdf (last access: 3.5. 2013). Aleš Hladnik Dr., Ass. Prof., Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales. hladnik@ ntf. uni-lj. si* 6.16 (2009), p. 4.

[80] Leland Roberts. *Understanding the Mel Spectrogram*. Accessed: 2025-02-13. 2018. URL: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

[81] Douglas Romesburg and Charles E. Williams. "Normalizing the loudness of audio recordings". WO2006055058A1. 2006.

[82] Aradhna Saini, Gaurav Dhuriya, Ayush Jain, and Amit Mishra. "Machine Learning Algorithms and Applications". In: (2024), pp. 1–31. DOI: 10.1201/9781003504900-1.

[83] Ali Osman Mohammed Salih. "Audio Noise Reduction Using Low Pass Filters". In: *Open Access Library Journal* 4.11 (2017), pp. 1–7. DOI: 10.4236/oalib.1103709.

[84] SAP SE. *What is Predictive Maintenance? A Complete Overview*. Accessed: 2025-02-13. 2025. URL: https://www.sap.com/products/scm/apm/what-is-predictive-maintenance.html.

[85] ScienceDirect. *Audio Feature Extraction*. Accessed: 2025-02-13. 2025. URL: https://www.sciencedirect.com/topics/engineering/audio-feature.

[86] ScienceDirect. *Audio Segmentation*. Accessed: 2025-02-13. 2025. URL: https://www.sciencedirect.com/topics/engineering/audio-segmentation.

[87] ScienceDirect. *Discrete Wavelet Transform*. Accessed: 2025-02-13. 2025. URL: https://www.sciencedirect.com/topics/mathematics/discrete-wavelet-transform.

[88] ScienceDirect. *Signal Processing*. Accessed: 2025-02-13. 2025. URL: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/signal-processing.

[89] Sensemore. *Cost Savings through Predictive Maintenance*. Accessed: 2025-02-13. 2024. URL: https://sensemore.io/cost-savings-through-predictive-maintenance/.

[90] Serkan Celik, Huawei Developers. *Basics of Audio Processing*. Accessed: 2025-02-13. 2023. URL: https://medium.com/huawei-developers/basics-of-audio-processing-e69efce7765f.

[91] Shell Oil Company. *The Benefits of Predictive Maintenance*. Accessed: 2025-02-13. 2025. URL: https://www.shell.us/business/fuels-and-lubricants/lubricants-for-business/lubricants-services/industry-articles/the-benefits-of-predictive-maintenance.html.

[92] D.S. Shete and Prof Patil. "Zero crossing rate and Energy of the Speech Signal of Devanagari Script". In: *IOSR journal of VLSI and Signal Processing* 4 (2014), pp. 01–05. DOI: 10.9790/4200-04110105.

[93] Pratham Shimpi. "Reinforcement Learning in Real Life Applications". In: *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT* 09 (2025), pp. 1–9. DOI: 10.55041/IJSREM40881.

[94] Siemens. *The True Cost of Downtime: Identify, Avoid, and Overcome*. Report. Siemens, Digital Industries, Customer Services, 2023.

[95] Nebojsa Simic and Ana Gavrovska. "Normalization of audio signals for the needs of machine learning". In: *2023 31st Telecommunications Forum (TELFOR)*. 2023, pp. 1–4. DOI: 10.1109/TELFOR59449.2023.10372705.

[96] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. "A review of supervised machine learning algorithms". In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. 2016, pp. 1310–1315.

[97] Julius O. Smith. *Choice of Hop Size*. Accessed: 2025-02-13. 2011. URL: https://www.dsprelated.com/freebooks/sasp/Choice_Hop_Size.html.

[98] Petre Stoica and Randolph Moses. *Spectral Analysis of Signals*. 2005. ISBN: 0-13-113956-8.

[99] Niroop Sugunaraj and Prakash Ranganathan. "Applications for Autoencoders in Power Systems". In: (2024), pp. 1–7. DOI: 10.1109/naps61145.2024.10741685.

[100] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2018. ISBN: 0262039249.

[101] Yuki Tagawa, Rytis Maskeliūnas, and Robertas Damaševičius. "Acoustic Anomaly Detection of Mechanical Failures in Noisy Real-Life Factory Environments". In: *Electronics* 10.19 (2021), p. 2329. DOI: 10.3390/electronics10192329.

[102] Theodoros Theodorou, Iosif Mporas, and Nikos Fakotakis. "An Overview of Automatic Audio Segmentation". In: *International Journal of Information Technology and Computer Science* 6 (2014), pp. 1–9. DOI: 10.5815/ijitcs.2014.11.01.

[103] Pythoneers Thomas Konstantinovsky. *Wavelet Transform: A Practical Approach to Time-Frequency Analysis*. Accessed: 2025-02-15. 2024. URL: https://medium.com/pythoneers/wavelet-transform-a-practical-approach-to-time-frequency-analysis-662bdadeb08b.

[104] İrem Üstek, Miguel Arana-Catania, Alexander J. Farr, and Ivan Petrunin. "Deep Autoencoders for Unsupervised Anomaly Detection in Wildfire Prediction". In: (2024). DOI: 10.48550/arxiv.2411.09844.

[105] M. Venkata Sudhakar, M. Prabhu Charan, G. Naga Pranai, L. Harika, and P. Yamini. "Audio signal noise cancellation with adaptive filter techniques". In: *Materials Today: Proceedings* 80 (2023), pp. 2956–2963. ISSN: 2214-7853. DOI: https://doi.org/10.1016/j.matpr.2021.07.080.

[106] Shangfei Wang. "Isolation Forest Anomaly Detection Algorithm Based On Multi-level Sub-subspace Partition". In: *International Journal of Computer Science and Information Technology* (2024). DOI: 10.62051/ijcsit.v4n2.20.

[107] William W. S. Wei. "Spectral Analysis". In: (2008).

[108] Terry Wireman. *Total productive maintenance*. 2004.

[109] Dazhong Wu, Connor Jennings, Janis Terpenny, and Soundar Kumara. "Cloud-based machine learning for predictive analytics: Tool wear prediction in milling". In: *2016 IEEE International Conference on Big Data (Big Data)*. 2016, pp. 2062–2069. DOI: 10.1109/BigData.2016.7840831.

[110] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. "Deep Isolation Forest for Anomaly Detection". In: *IEEE Transactions on Knowledge and Data Engineering* 35.12 (2023), pp. 12591–12604. DOI: 10.1109/TKDE.2023.3270293.

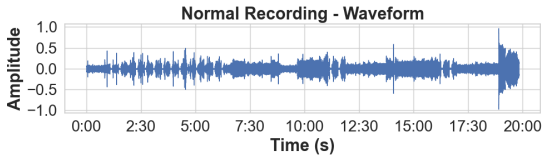
[111] Yufeng Zhang, Zhenyu Guo, Weilian Wang, Side He, Ting Lee, and Murray Loew. "A comparison of the wavelet and short-time fourier transforms for Doppler spectral analysis". In: *Medical Engineering Physics* 25.7 (2003), pp. 547–557. ISSN: 1350-4533. DOI: https://doi.org/10.1016/S1350-4533(03)00052-3.

[112] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. "Predictive maintenance in the Industry 4.0: A systematic literature review". In: *Computers Industrial Engineering* 150 (2020), p. 106889. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106889.

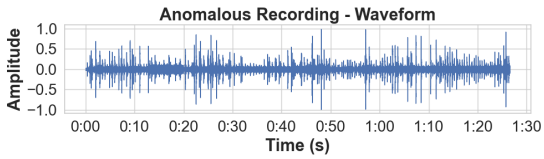
Appendix A
Exploratory Data Analysis Results

A.1 Washing Machine Dataset

This section presents the EDA results for the Washing Machine dataset.

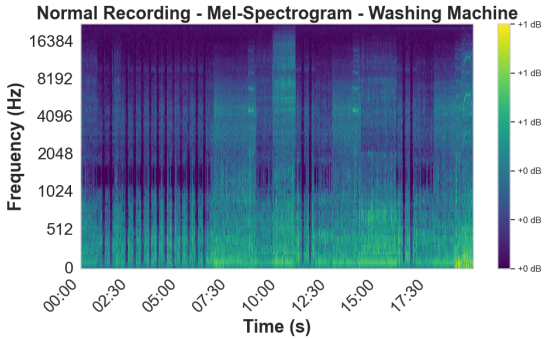


(a) Normal Washing Machine Recording — Raw Waveform

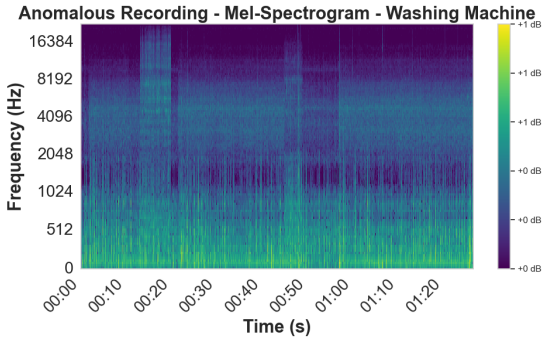


(b) Anomalous Washing Machine Recording — Raw Waveform

Figure A.1: Comparison of Anomalous and Normal Recordings Raw Waveforms



(a) Normal Washing Machine Recording — Mel-Spectogram

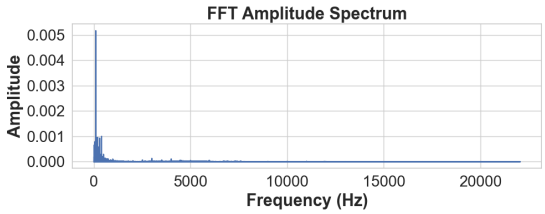


(b) Anomalous Washing Machine Recording — Mel-Spectogram

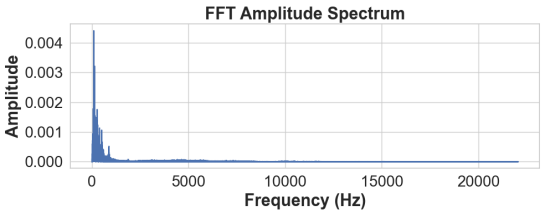
Figure A.2: Comparison of Anomalous and Normal Recordings Mel-Spectograms

Metric	Anomalous	Normal
Spectral Centroid (Hz)	2945.91	3962.37
Spectral Rolloff (Hz)	6020.86	8502.82
Spectral Contrast	{ 14.62, 8.23, 12.23, 16.38, 14.08, 14.93, 20.17 }	{ 14.40, 8.30, 12.23, 16.40, 13.88, 15.50, 18.83 }
Zero Crossing Rate	0.047	0.070

Table A.1: Spectral Metrics for the Washing Machine Dataset

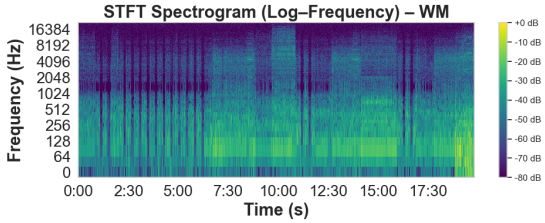


(a) Normal WM Recording — FFT Amplitude

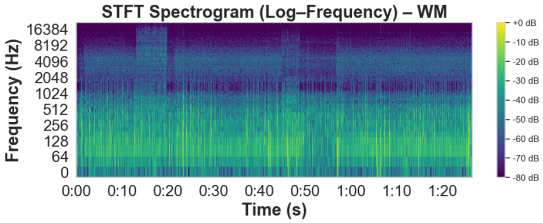


(b) Anomalous WM Recording — FFT Amplitude

Figure A.4: Comparison of WM Recordings FFT Amplitude

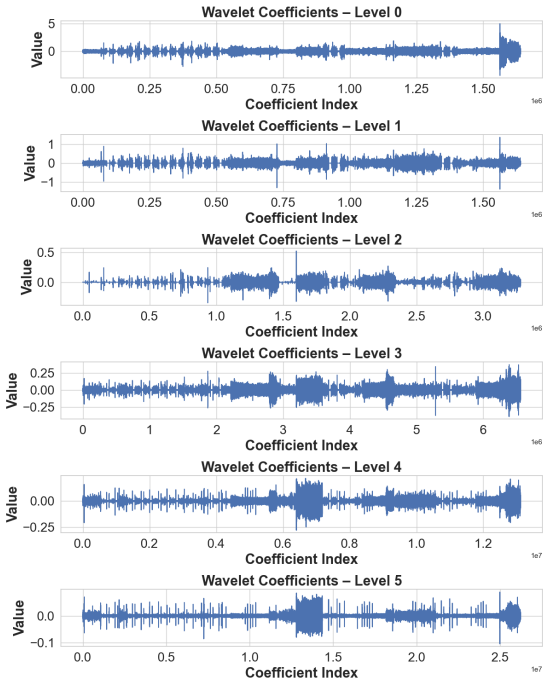


(a) Normal WM Recording — STFT Mel-spectogram

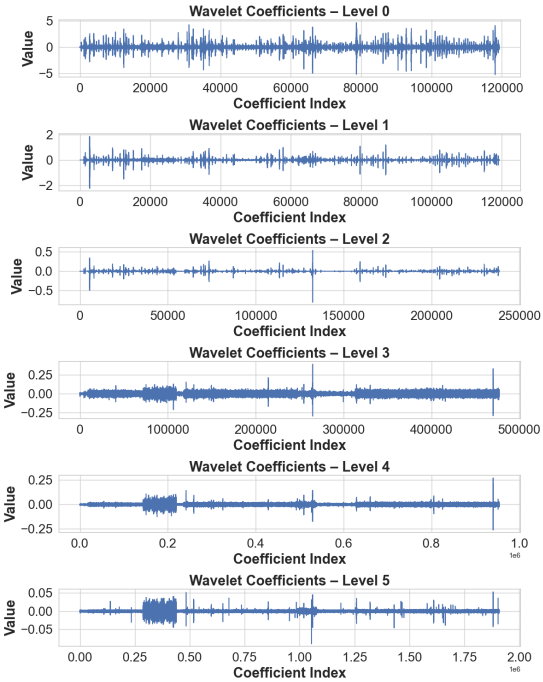


(b) Anomalous WM Recording — STFT Mel-spectogram

Figure A.5: Comparison of WM Recordings STFT-spectograms



(a) Normal WM Recording — Wavelet coefficients

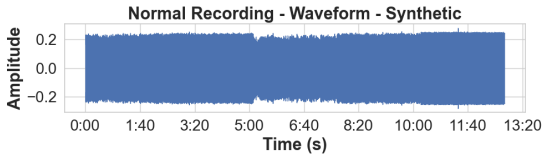


(b) Anomalous WM Recording — Wavelet coefficients

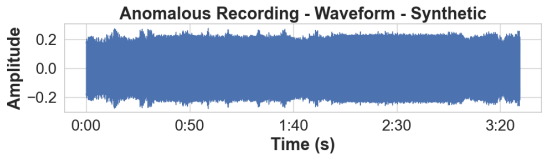
Figure A.6: Comparison of Anomalous and Normal Recordings Wavelet Coefficients

A.2 Synthetic Dataset

This section presents the EDA results for the Synthetic dataset.

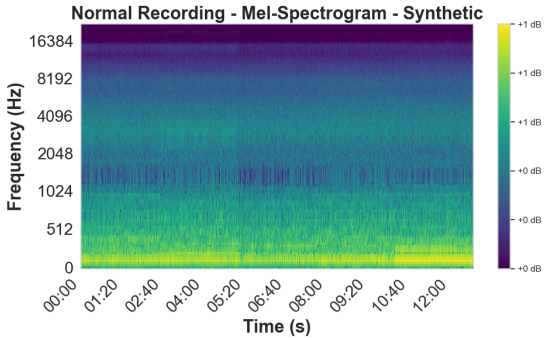


(a) Normal Synthetic Recording — Raw Waveform

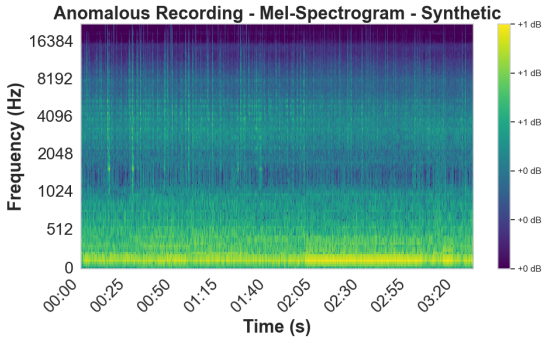


(b) Anomalous Synthetic Recording — Raw Waveform

Figure A.7: Comparison of Anomalous and Normal Recordings Raw Waveforms



(a) Normal Synthetic Recording — Mel-Spectogram

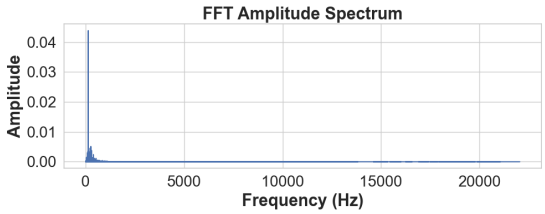


(b) Anomalous Synthetic Recording — Mel-Spectogram

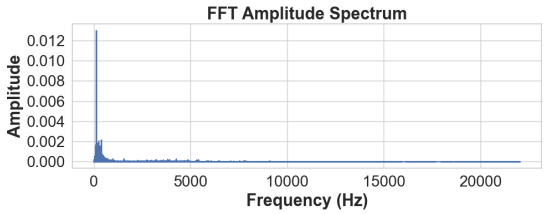
Figure A.8: Comparison of Anomalous and Normal Recordings Mel-Spectograms

Metric	Anomalous	Normal
Spectral Centroid (Hz)	2194.10	1595.73
Spectral Rolloff (Hz)	4890.26	3675.63
Spectral Contrast	{ 20.04, 8.14, 12.43, 15.52, 14.22, 17.46, 27.91 }	{ 20.78, 8.73, 12.10, 15.74, 13.65, 16.13, 28.39 }
Zero Crossing Rate	0.029	0.016

Table A.2: Spectral Metrics for the Synthetic Dataset

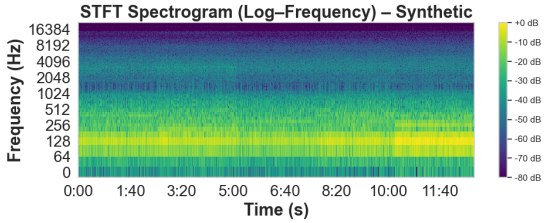


(a) Normal Synthetic Recording — FFT Amplitude

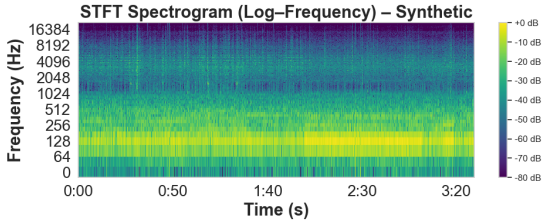


(b) Anomalous Synthetic Recording — FFT Amplitude

Figure A.10: Comparison of Synthetic Recordings FFT Amplitude

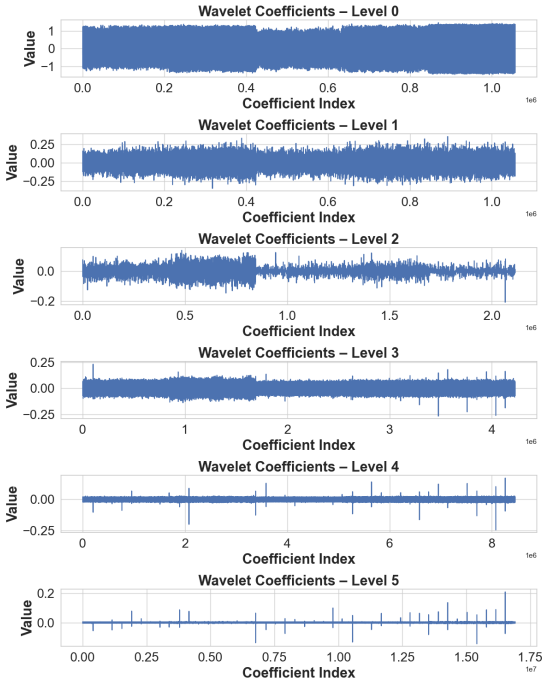


(a) Normal Synthetic Recording — STFT Mel-spectogram

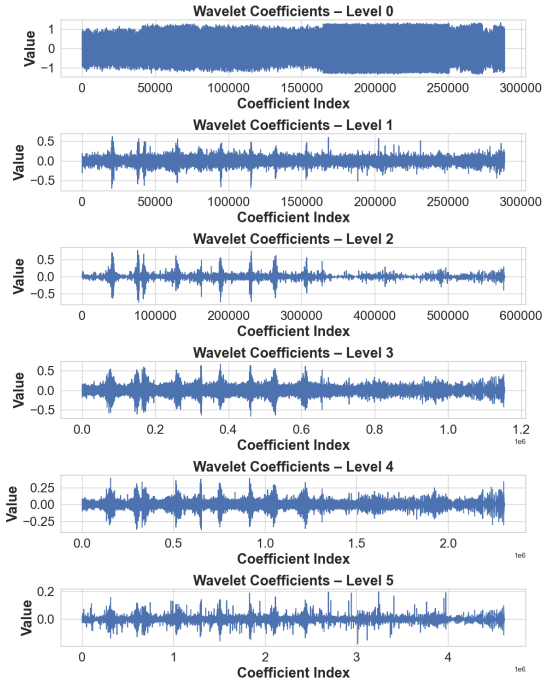


(b) Anomalous Synthetic Recording — STFT Mel-spectogram

Figure A.11: Comparison of Synthetic Recordings STFT-spectograms



(a) Normal Synthetic Recording — Wavelet coefficients

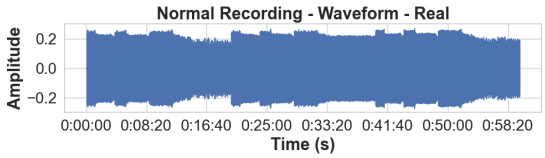


(b) Anomalous Synthetic Recording — Wavelet coefficients

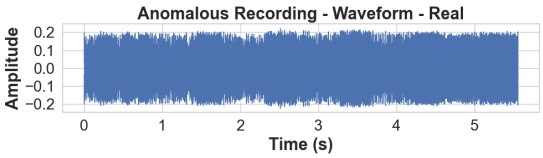
Figure A.12: Comparison of Anomalous and Normal Recordings Wavelet Coefficients

A.3 Real Industrial Machine Dataset

This section presents the EDA results for the Real Industrial Machine dataset.

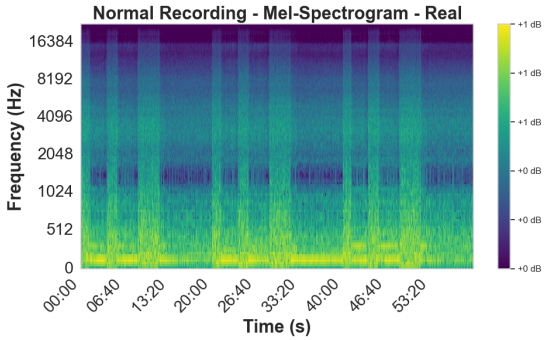


(a) Normal Real Recording — Raw Waveform

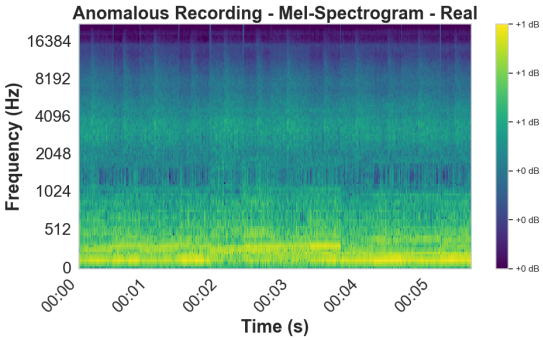


(b) Anomalous Real Recording — Raw Waveform

Figure A.13: Comparison of Anomalous and Normal Recordings Raw Waveforms

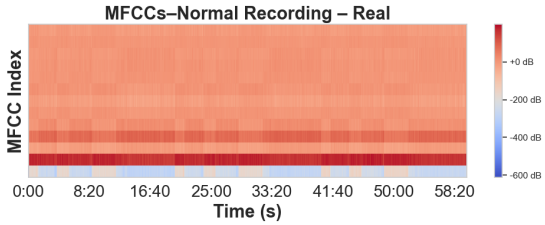


(a) Normal Real Recording — Mel-Spectogram

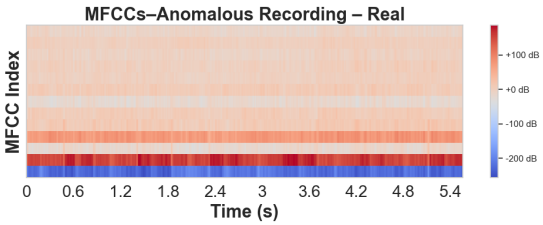


(b) Anomalous Real Recording — Mel-Spectogram

Figure A.14: Comparison of Anomalous and Normal Recordings Mel-Spectograms



(a) Normal Real Recording — MFCCs

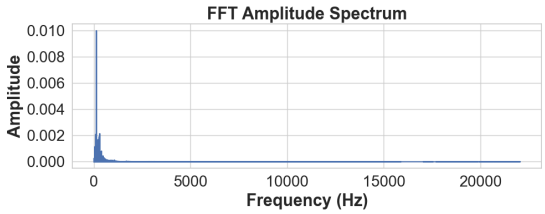


(b) Anomalous Real Recording — MFCCs

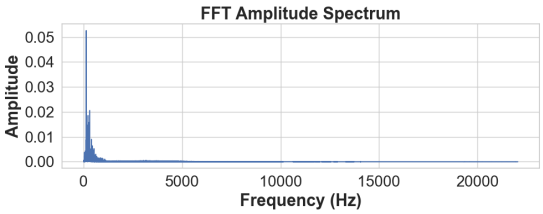
Figure A.15: Comparison of Anomalous and Normal Recordings MFCCs

Metric	Anomalous	Normal
Spectral Centroid (Hz)	1953.99518	1758.28663
Spectral Rolloff (Hz)	4321.02606	3948.34216
Spectral Contrast	{ 18.88061, 9.75598, 12.37520, 15.49265, 13.50399, 16.23916, 26.45347 }	{ 17.49419, 8.52598, 11.99904, 16.88300, 13.46328, 15.95403, 27.69915 }
Zero Crossing Rate	0.02382	0.02373

Table A.3: Spectral Metrics for the Real Industrial Dataset

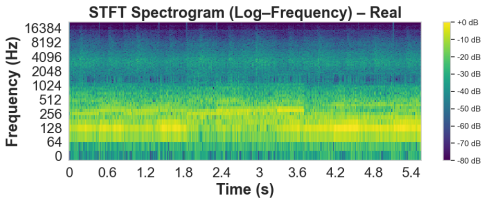


(a) Normal Real Recording — FFT Amplitude

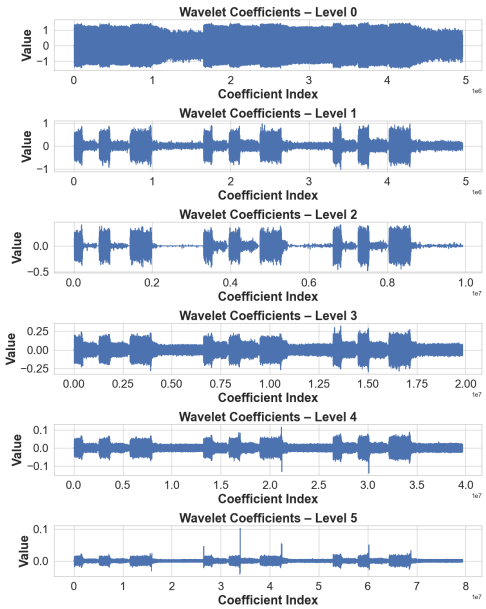


(b) Anomalous Real Recording — FFT Amplitude

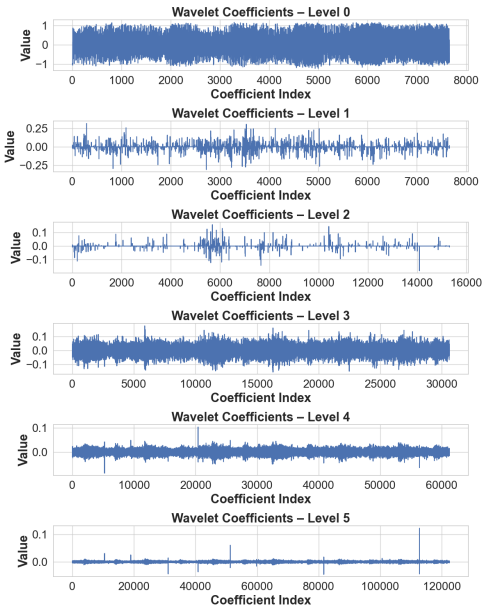
Figure A.16: Comparison of Real Recordings FFT Amplitude



(a) Anomalous Real Recording — STFT Mel-spectogram



(a) Normal Real Recording — Wavelet coefficients



(b) Anomalous Real Recording — Wavelet coefficients

Figure A.18: Comparison of Anomalous and Normal Recordings Wavelet Coefficients

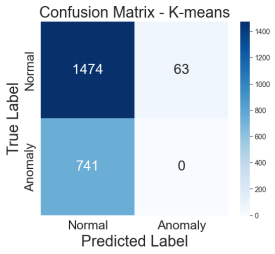
Appendix B
Model Performance Results

B.1 Washing Machine Dataset

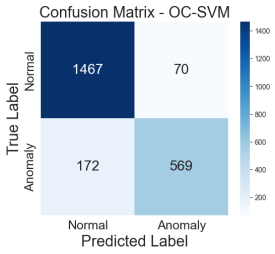
This section presents the results of the model training performance for the Washing Machine dataset.

Model	Train Time (s)	ROC AUC	Precision	Recall	F1 Score	Inference Time (s)
K-Means	0.7403	0.5871	0.0000	0.0000	0.0000	0.0270
OC-SVM	7.0023	0.9659	0.8905	0.7679	0.8246	1.9641
LSTM-AE	49.0513	0.8885	0.8521	0.1943	0.3165	0.3669

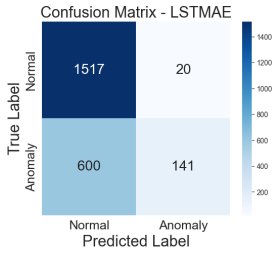
Table B.1: Performance Comparison on the Washing Machine Dataset



(a) K-Means Confusion Matrix

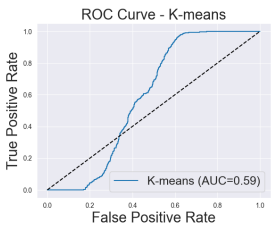


(b) OC-SVM Confusion Matrix

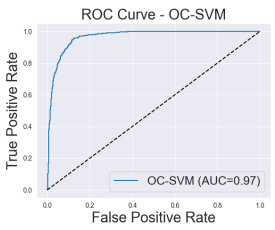


(c) LSTM AE Confusion Matrix

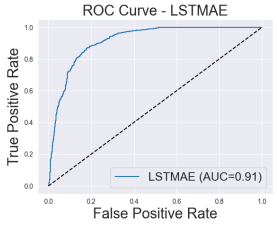
Figure B.1: Comparison of Confusion Matrices for Washing Machine Dataset



(a) K-Means ROC AUC



(b) OC-SVM ROC AUC



(c) LSTM AE ROC AUC

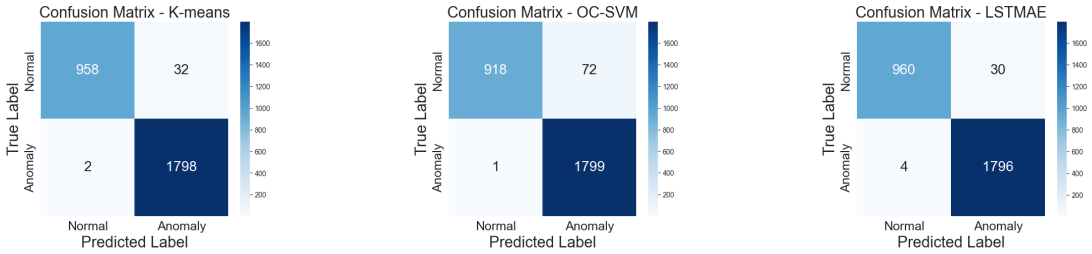
Figure B.2: Comparison of ROC AUCs for Washing Machine Dataset

B.2 Synthetic Dataset

This section presents the results of the model training performance for the Synthetic dataset.

Model	Train Time (s)	ROC AUC	Precision	Recall	F1 Score	Inference Time (s)
K-Means	0.3700	0.9974	0.9825	0.9989	0.9906	0.0275
OC-SVM	2.8284	0.9977	0.9615	0.9994	0.9801	2.1469
LSTM-AE	32.8777	0.9995	0.9814	0.9972	0.9893	0.4765

Table B.2: Performance Comparison on the Synthetic Industrial Machine Dataset

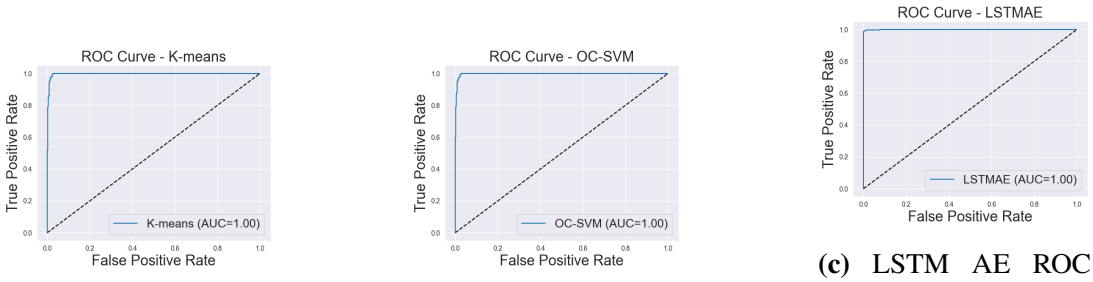


(a) K-Means Confusion Matrix

(b) OC-SVM Confusion Matrix

(c) LSTM AE Confusion Matrix

Figure B.3: Comparison of Confusion Matrices for Synthetic Dataset



(a) K-Means ROC AUC

(b) OC-SVM ROC AUC

(c) LSTM AE ROC AUC

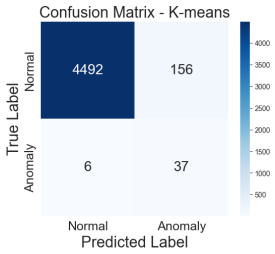
Figure B.4: Comparison of ROC AUCs for Synthetic Dataset

B.3 Real Industrial Machine Dataset

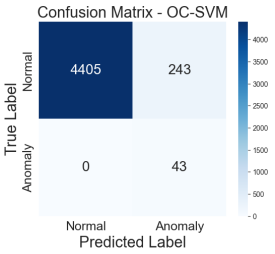
This section presents the results of the model training performance for the Real Industrial Machine dataset.

Model	Train Time (s)	ROC AUC	Precision	Recall	F1 Score	Inference Time (s)
K-Means	5.2506	0.9618	0.1917	0.8605	0.3136	0.0430
OC-SVM	55.5363	0.9976	0.1503	1.0000	0.2614	12.1680
LSTM-AE	145.2456	0.9974	0.2194	1.0000	0.3598	0.8667

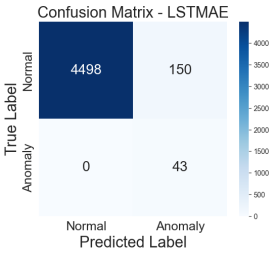
Table B.3: Performance Comparison on the Real Industrial Machine Dataset



(a) K-Means Confusion Matrix

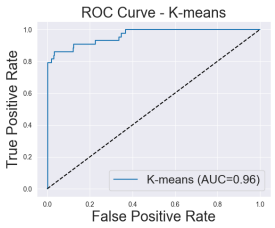


(b) OC-SVM Confusion Matrix

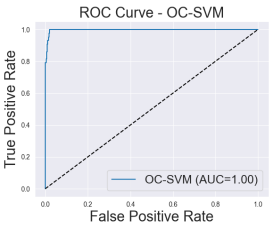


(c) LSTM AE Confusion Matrix

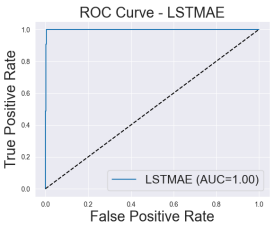
Figure B.5: Comparison of Confusion Matrices for Real Industrial Machine Dataset



(a) K-Means ROC AUC



(b) OC-SVM ROC AUC



(c) LSTM AE ROC AUC

Figure B.6: Comparison of ROC AUCs for Real Industrial Machine Dataset

Declaration of Authorship

‘I hereby declare,

- that I have written this thesis independently;
- that I have written the articles and contributions using only the aids listed in the index;
- that all parts of the thesis produced with the help of aids (incl. AI-Tools) have been precisely declared;
- that I have mentioned all sources used and cited them correctly according to established academic citation rules; this also includes the comprehensible disclosure of all personal publications;
- that I have acquired all immaterial rights to any materials I may have used, such as images or graphics, or that these materials were originally created by myself;
- that I am aware of the legal provisions regarding the publication and dissemination of parts or the entire thesis and that I comply with them accordingly;
- that I am aware that the University will prosecute a violation of this Declaration of Authorship and that disciplinary as well as criminal consequences may result, which may lead to expulsion from the University or to the withdrawal of my title.’

By submitting this thesis, I confirm through my conclusive action that I am submitting the statutory declaration, that I have read and understood it, and that it is true.

St. Gallen, May 19, 2025 Signature: Nicolas Keller

Declaration of Auxiliary Aids

The creation of this dissertation involved a lot of third-party software, including AI-based tools. I declare all uses of third-party software in the dissertation text that contributed non-trivially to the dissertation's content and are non-standard in such research. Further, by the University of St. Gallen's decree II.B.1.20 section 5.3, I explicitly declare the following uses:

The following tools were used in the completion of this thesis:

Aid	**Usage**	**Affected Parts**
GPT-4o GitHub Copilot Claude AI	Coding assistance	Code
GPT-4o Google Translate Writefull Grammarly	Spell and grammar checking, translations, and rewordings	Complete paper
SciSpace Perplexity AI	Finding Academic Sources	Research work