A Case Study of Acoustic-based Anomaly Detection for Industrial Predictive Maintenance

Valentin Huber, Stefan Krummenacher
School of Computer Science SCS, University of St. Gallen HSG
Rosenbergstrasse 30, 9000, St. Gallen, Switzerland
E-mail: [valentin.huber, stefan.krummenacher]@student.unisg.ch

***Abstract*—Unplanned downtime in industrial machinery can lead to significant financial losses, underscoring the critical need for effective predictive maintenance strategies. Recent advancements in acoustic anomaly detection have facilitated the development of such systems. However, domain shifts and the transition from controlled synthetic environments to real-world applications presents challenges due to variability in noise and overlapping acoustic patterns. To address these challenges, we propose an acoustic anomaly detection framework for industrial machinery. This approach is designed to evaluate anomalies in both synthetic and real-world datasets by leveraging a combination of signal processing techniques with Autoencoder (AE) and Local Outlier Factor (LOF) algorithms. Additionally, Isolation Forest (IF) and Principal Component Analysis (PCA) are incorporated as comparative benchmarks to evaluate the relative performance of density-based and reconstruction-based methods. Experimental results demonstrate that LOF achieves high precision and recall under noisy, variable conditions, while AE performs comparably in structured, low-noise settings with reduced computational overhead once trained. These findings highlight the complementary strengths of AE and LOF, suggesting that hybrid approaches could further enhance anomaly detection reliability in complex industrial scenarios.**

***Index Terms*—Industrial Machinery, Acoustic Anomaly Detection, Predictive Maintenance, Autoencoder (AE), Local Outlier Factor (LOF)**

I. Introduction

Maintenance is a crucial activity in industrial environments, as unexpected equipment failures result in operational downtime. According to recent industry estimates [1], unplanned downtime can cost an average of $129 million per year. Fortune Global 500 firms collectively face downtime losses of approximately $1.5 trillion annually (nearly 11% of their total revenues), marking a 70% increase from previous periods. Consequently, determining the ideal time to perform maintenance is crucial: intervening too late risks catastrophic failures, whereas scheduling maintenance too early leads to unnecessary production halts. In response, predictive maintenance has gained traction for its potential to identify an optimal maintenance window, reducing costs and boosting system reliability [2], [3], [4], [5], [6].

Among the many approaches to predictive maintenance, acoustic-based anomaly detection stands out for its ability to capture latent faults in machinery. By analyzing deviations in acoustic signals, one can detect early signs of wear or incipient failures before they escalate into major breakdowns [7], [8], [9]. However, real-world industrial environments often feature high noise, overlapping sound sources, and limited labeled anomaly data [10], [9], [11], making reliable fault detection particularly challenging.

Traditional strategies frequently rely on signal processing methods, such as the Short-Time Fourier Transform (STFT) and Mel-spectrogram representations, to capture relevant time-frequency patterns [11], [3], [10]. Yet, these methods may not fully capture nuanced or high-dimensional phenomena under noisy conditions [10]. Density-based algorithms like the Local Outlier Factor (LOF) have shown promise in identifying outliers based on local density deviations [12], [13], [14], though they can lead to high computational overhead [15].

Deep learning approaches, especially those involving Autoencoders (AEs), have substantially enhanced acoustic anomaly detection. AEs are trained on normal data to learn representative features, then flag potential anomalies when reconstruction errors surpass typical levels [5], [16], [8], [17], [9]. Despite their success, issues related to noise resilience, scalability, and computational overhead persist, highlighting the need for robust preprocessing and domain-specific adaptations [10], [11].

To address these challenges, this study proposes an acoustic anomaly detection framework tailored to industrial machinery where downtime is particularly costly. The framework integrates Mel-spectrogram preprocessing, an AE for reconstruction-based anomaly detection, and LOF for density-based benchmarking. By validating the approach on datasets from both controlled laboratory settings and real-world recordings, we demonstrate its effectiveness under noisy, variable industrial conditions, showing that due to the complementary strengths of AE and LOF, a hybrid approach could enhance reliability. Furthermore, we showcase the practical applicability of acoustic anomaly detection for an industrial machine use case.

The remainder of this paper is organized as follows: Section II reviews existing literature on acoustic anomaly detection, focusing on both traditional signal processing and machine learning approaches. Section III presents the proposed framework, detailing its preprocessing pipeline and model architectures. Section IV discusses experimental results and provides a comparative evaluation on lab-based and real-world datasets. Finally, Section V offers concluding remarks and outlines directions for future research.

II. Related Work

Anomaly detection is a key component of predictive maintenance, enabling the early identification of faults in industrial systems [8], [7]. Anomalies, characterized as deviations from normal operational behavior, are detected through measurable differences in data patterns [12]. Current techniques for anomaly detection in acoustic systems can be broadly categorized into signal processing-based methods, machine learning-based approaches, and deep learning frameworks. This section reviews state-of-the-art techniques and identifies gaps that motivate further research, with an emphasis on acoustic anomaly detection.

A. Signal Processing-Based Methods

Signal processing techniques are foundational for extracting meaningful features from noisy industrial data. Methods such as the Short-Time Fourier Transform (STFT) and Mel-scale transformation are widely used for representing acoustic signals in the frequency domain [17], [3], [10]. These methods form the basis of many frameworks, converting raw audio signals into spectrograms to provide a time-frequency representation of the data.

[11] employed spectral coefficients of acoustic signals to identify faults in rotating machinery, demonstrating how frequency-domain features are critical for detecting deviations indicative of mechanical failures. Similarly, [7] applied advanced preprocessing techniques, such as 3D tensor representations of sound, to enhance the detection of anomalies under noisy conditions. Their approach emphasized the value of combining temporal and spectral characteristics to improve the robustness of anomaly detection models.

Despite their effectiveness, signal processing-based methods often fail to capture complex relationships in high-dimensional data, particularly in environments with significant noise interference. These limitations have driven the adoption of more sophisticated machine learning and deep learning approaches, which excel at modeling intricate patterns.

B. Machine Learning-Based Methods

Machine learning techniques, particularly density- and clustering-based methods, have been used for anomaly detection. One prominent example is the Local Outlier Factor (LOF), a density-based algorithm that identifies anomalies by comparing the local density of a data point with its neighbors [12]. LOF has been successfully applied in industrial settings, such as estimating operation, maintenance, and avoidance of unplanned downtime of rotating machinery [13] and the measurement error evaluation of power metering equipment for accurate metering of electric energy [14].

A significant advantage of LOF lies in its unsupervised nature, which allows it to detect anomalies without requiring labeled datasets. This capability makes it particularly suitable for industrial applications where anomalous data is scarce. However, LOF is constrained by its reliance on explicit distance and density measures, which may not effectively capture latent representations in high-dimensional data. Additionally, LOF is not commonly used for audio data but excels at high-dimensional spectral data [18], [19].

[9] extended machine learning techniques to classify acoustic emission events for wear monitoring in sliding bearing systems. Their work highlighted how clustering and classification models can enhance anomaly detection by leveraging domain-specific acoustic features. Despite these advancements, traditional machine learning models often struggle to scale to the complexity of modern industrial data, necessitating the adoption of deep learning approaches.

C. Deep Learning-Based Methods

Deep learning has emerged as a cornerstone for modern anomaly detection, particularly in high-dimensional domains such as acoustic data. Autoencoders (AEs) [20] are one of the most widely used architectures in this context [5], [16], [10], [11]. These models are trained to reconstruct normal data patterns, identifying anomalies based on elevated reconstruction errors. This makes AEs particularly effective for capturing the subtle deviations present in acoustic signals.

[16] demonstrated the efficacy of convolutional autoencoders (CAEs) for processing spectrogram inputs, achieving high accuracy in detecting anomalies during industrial processes. Similarly, [17] explored convolutional LSTM autoencoders to model temporal dependencies in sequential data, emphasizing the benefits of combining temporal and spatial features for anomaly detection in manufacturing systems.

Recent advancements have enhanced the capabilities of AEs. For example, [21] introduced a self-supervised learning framework leveraging contrastive learning to improve the robustness of AEs in scenarios with limited labeled data. [7] extended AE frameworks by integrating 3D CNNs to process tensor representations of sound, enabling precise detection of anomalies even in noisy industrial environments.

[22] compared various AE architectures for real-time acoustic anomaly detection, highlighting the superiority of deep convolutional designs in terms of both detection accuracy and computational efficiency. These studies collectively demonstrate that AEs excel at modeling high-dimensional data, particularly when used in conjunction with preprocessing techniques such as Mel-scale transformations.

D. Applications of Acoustic Anomaly Detection

Acoustic anomaly detection is particularly challenging in noisy environments due to overlapping sound sources and variable operating conditions. Addressing these challenges often requires robust preprocessing pipelines. [10] applied preprocessing techniques such as normalization and synchronization to deploy lightweight models on edge devices for real-time anomaly detection. Their work underscored the importance of preprocessing in mitigating noise and ensuring reliable performance.

[23] focused on detecting abnormal sounds in power plant equipment using self-supervised learning. By combining spectral analysis with a self-supervised pretraining phase, their framework achieved significant improvements in detecting anomalies in dynamic industrial environments. Similarly, [24] integrated spectrogram analysis with MEMS sensors and PyTorch implementations to enhance sound recognition in noisy conditions.

These studies highlight the importance of designing models that are both noise-resilient and computationally efficient. While preprocessing techniques and deep learning architectures address many of these challenges, integrating domain-specific knowledge into these models remains an open area of research.

E. Integrating Signal Processing, Machine Learning, and Deep Learning

The reviewed studies illustrate the progression of anomaly detection techniques, from signal processing-based approaches to machine learning and deep learning frameworks. While signal processing provides foundational tools for feature extraction, the scalability and flexibility of deep learning models have expanded the capabilities of anomaly detection systems. The integration of autoencoders and density-based methods, such as LOF, offers a promising avenue for addressing noise resilience, scalability, in industrial acoustic environments. Building on these advancements, we aim to develop a robust and scalable framework for anomaly detection in domain specific industrial machinery.

III. Design and Implementation

Our goal in this Section is to demonstrate that a deep-learning framework can successfully detect anomalous events in the specific domain of industrial machinery using only a minimal amount of representative data. We incorporate an Autoencoder (AE) and benchmark it against Local Outlier Factor (LOF), Isolation Forest (IF) and Principal Component Analysis (PCA) algortihms. Building on the methodology proposed by [8] for anomalous sound detection, we adopt a simplified pipeline better suited to our limited datasets and adapt the AE architecture from [19]. While in an ideal scenario we would include more comprehensive recordings, the aim here is to illustrate the feasibility of applying this approach to specific industrial machinery, rather than to exhaustively characterize all forms of anomalous behavior.

An overview of the proposed framework for acoustic anomaly detection is illustrated in Figure 1. The framework integrates preprocessing, feature extraction, AE training, and evaluation using reconstruction-based methods. Subsequently, the framework is benchmarked against LOF, IF, and PCA to assess comparative performance.

The design of this pipeline reflects key considerations for handling real-world industrial acoustic-data, including noise resilience, efficient anomaly representation, and scalable deployment in both lab-controlled and real industrial environments.

A. Data Pipeline Overview

The pipeline comprises four main stages:

1) Data Acquisition: Audio recordings of machine operations in controlled and real-world environments.
2) Preprocessing: Transformation of raw audio into normalized Mel-spectrograms.
3) Training: Development of an AE, LOF, and IF trained exclusively on normal data.
4) Anomaly Detection and Evaluation: Identification of anomalies using reconstruction error and comparing to the benchmarks.

A detailed explanation of each stage is provided below.

B. Data Acquisition

For this study, three distinct datasets were collected to investigate the feasibility of acoustic anomaly detection in specific industrial machinery. This was achieved by recording machine sounds, cutting them into separate normal and anomalous *.wav* files, which could be used for data preprocessing. As the primary objective was to demonstrate the applicability of our method rather than conduct a comprehensive survey of all operational scenarios, no further datasets were recorded or added. Despite differences in the datasets, the methodologies reported in this paper remain identical across all three datasets, highlighting the transferability of our approach.

C. Data Preprocessing

Prior to modelling, raw audio signals must be converted into representations that are both stable and discriminative in the context of acoustic anomaly detection. In this work, we use a Short-Time Fourier Transform (STFT) followed by a Mel-spectrogram conversion and a frame-based segmentation procedure. These steps are essential to transform the original time domain signals into a form more suited to machine learning methods such as AE, LOF, IF or PCA.

1) Short-Time Fourier Transform (STFT): We apply a standard discrete Short-Time Fourier Transform (STFT) [25], [26], [27] from the *librosa* library with a 1024-sample window and a hop length of 512. By decomposing the audio into short, overlapping time segments, we obtain local frequency-domain information, enabling the detection of short-duration events (e.g., knocks or impacts). This time-frequency representation is critical for capturing the subtle temporal patterns that characterize anomalies in industrial or mechanical systems. Mathematically, the STFT is defined in Equation 1, where we sum over windowed signal frames and apply the discrete Fourier transform.

$$\mathrm{STFT}(n,k) = \sum_{m=0}^{L-1} x(n+m)\, w(m)\, e^{-j2\pi \frac{k}{N} m} \tag{1}$$

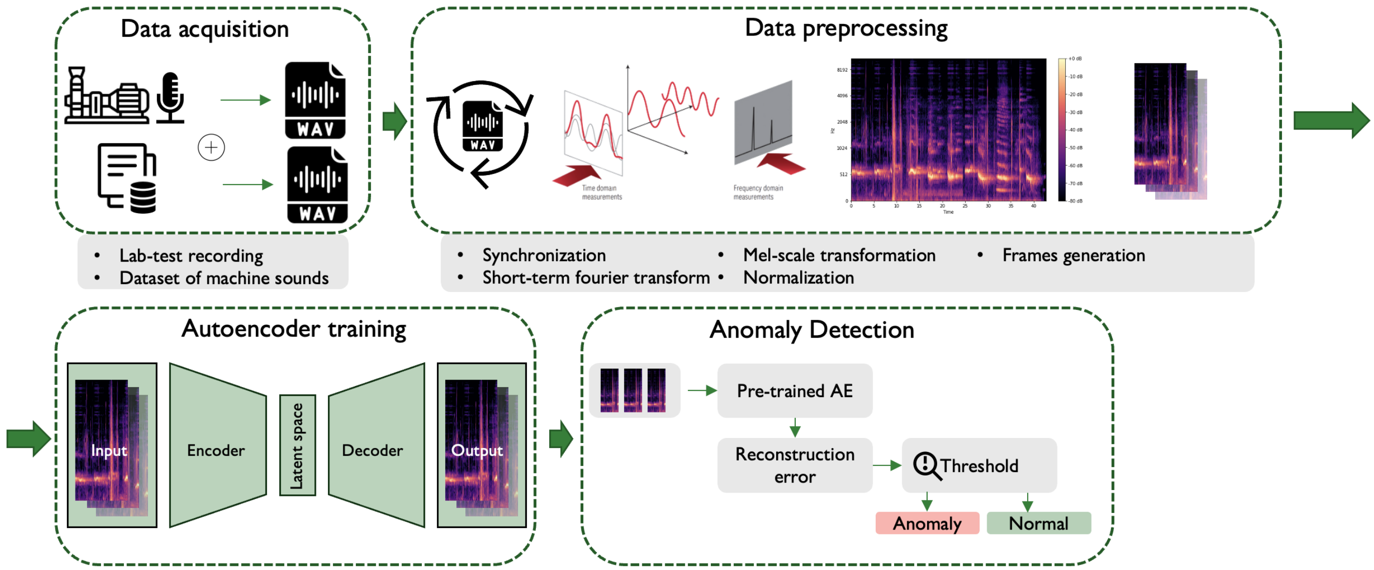


Fig. 1: Overview of the proposed pipeline for acoustic anomaly detection, including data acquisition, preprocessing, training, and anomaly detection.

2) Mel-Spectrogram Conversion: From the STFT magnitude, we derive a 128-band Mel-spectrogram [28], [26], [29]. The Mel scale aligns more closely with human auditory perception, ensuring that frequency regions vital to detecting anomalies (*e.g.*, sudden high-frequency bursts) are not overshadowed by less relevant areas of the spectrum. By compressing the frequency axis in this perceptual manner, the model is better able to focus on critical frequency components that differentiate normal from abnormal operations.

$$M(m,n) = \sum_{k=0}^{K-1} H_m(k)\,|STFT(n,k)|^2 \tag{2}$$

In Equation 2, each Mel bin is computed by summing the STFT power according to a triangular filter bank H_m.

$$M_{DB}(m,n) \;=\; 10\log_{10}\!\left(\frac{M(m,n)}{\max_{m,n}(M(m,n))}\right) \tag{3}$$

Then, the Mel-spectrogram is converted to decibels using Equation 3, normalizing each cell by the global maximum and applying $10\log_{10}$.

3) Normalization: Each Mel-spectrogram is min-max normalized [30] to a $[0,1]$ range as shown in Equation 4, which subtracts the minimum and divides by the overall range:

$$x_{\text{norm}} = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{4}$$

This standardization step makes training more robust by reducing amplitude-related variability and ensuring uniform feature scales across the dataset. As a result, the model is not biased by unusually loud or soft recordings, improving generalization to new, unseen data.

```python
time_per_frame = 0.6
hop_ratio      = 0.2
hop_length     = 512

def generate_mel_spectrogram(audio_path):
    audio, sr = librosa.load(audio_path, sr=None)
    stft = librosa.stft(audio, n_fft=1024,
        ↪ hop_length=hop_length)
    mel  = librosa.feature.melspectrogram(S=np.
        ↪ abs(stft)**2, sr=sr, n_mels=128)
    mel_db = librosa.power_to_db(mel, ref=np.max)
    mel_db_norm = (mel_db - mel_db.min()) / (
        ↪ mel_db.max() - mel_db.min())
    return mel_db_norm, sr
```

Listing 1: Code to Generate and Normalize Mel-Spectrograms

The Python code in Listing 1 illustrates how the first three steps are implemented in practice, including loading the audio data, applying an STFT, generating and normalizing Mel-spectrograms.

4) Frame Segmentation: After computing the Mel-spectrogram, we segment each spectrogram into overlapping frames. In our setup:

- Windowing: A frame length of 0.512 s is chosen to capture transient acoustic events while retaining sufficient frequency resolution.
- Hop Ratio: A 20 % overlap is maintained between consecutive frames (*i.e.*, a hop ratio of 0.2). This partial overlap preserves temporal continuity and guards against losing information at frame boundaries—particularly important for detecting short, impulsive anomalies.

By transforming each spectrogram into a series of frames, we produce localized snapshots of the acoustic signal. These frames serve as input units for both training and evaluation, making the subsequent anomaly detection task more fine-grained and sensitive to local transient events.

```python
def generate_frames(mel_spectrogram, frame_size, hop_size):

    num_frames = (mel_spectrogram.shape[1] - frame_size) // hop_size + 1
    frames = np.zeros((num_frames, mel_spectrogram.shape[0], frame_size))
    for i in range(num_frames):
        start = i * hop_size
        frames[i] = mel_spectrogram[:, start: start + frame_size]
    return frames
```

Listing 2: Code to Generate Overlapping Frames from a Mel-Spectrogram

The Python code in Listing 2 illustrates how step 4, generating the frames, is implemented in practice.

Each set of frames is then saved in `.npy` format for subsequent training and evaluation. By separating the raw audio loading from the downstream tasks, we ensure a clear workflow in which all models operate on the same preprocessed feature set.

D. Autoencoder Architecture

The encoder-decoder structure of the AE is designed to compress and reconstruct input data, enabling the identification of anomalies based on reconstruction errors [20]. We adopt a fully connected AE design, following [19]. The AE is trained only on normal frames, so that anomalous frames typically exhibit higher reconstruction errors.

```python
class Autoencoder(nn.Module):
    def __init__(self, input_shape):
        super(Autoencoder, self).__init__()
        flattened_size = input_shape[0] * input_shape[1]
        self.encoder = nn.Sequential(
            nn.Flatten(),
            nn.Linear(flattened_size, 64),
            nn.ReLU(),
            nn.Linear(64, 32),
            nn.ReLU(),
            nn.Linear(32, 16),
            nn.ReLU(),
            nn.Linear(16, 8))
        self.decoder = nn.Sequential(
            nn.Linear(8, 16),
            nn.ReLU(),
            nn.Linear(16, 32),
            nn.ReLU(),
            nn.Linear(32, 64),
            nn.ReLU(),
            nn.Linear(64, flattened_size),
            nn.Sigmoid(),
            nn.Unflatten(1, input_shape))

    def forward(self, x):
        encoded = self.encoder(x)
        decoded = self.decoder(encoded)
        return decoded
```

Listing 3: Code for the AE architecture

1) *Encoder:*

- Multiple fully connected (FC) layers sequentially compress each input frame—flattened from the (n_mels, frame_width) shape—down to a low-dimensional latent space (8–16 neurons).
- Each FC layer is followed by a ReLU activation to introduce nonlinearity.

2) *Decoder:*

- Mirrors the encoder by progressively expanding from the latent space to the original input size.
- A Sigmoid activation in the final layer normalizes reconstructed spectrograms to the range [0, 1], matching the normalization of the input Mel-spectrogram frames.

Listing 3 displays our PyTorch implementation of the AE.

3) Loss Function: The Mean Squared Error (MSE) between input and reconstructed frames is minimized, encouraging the network to learn normal patterns while assigning higher errors to anomalous inputs [31]. The MSE loss function is defined as 5:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{5}$$

where x_i is the original input, $\hat{x}_i$ is the reconstructed output, and n is the number of data points.

4) Optimizer: We use the Adam optimizer [32] with a learning rate of 1×10^{-4}, chosen for its stable convergence and efficient training. The Adam optimizer updates the network parameters using adaptive estimates of lower-order moments.

E. Benchmarks

To benchmark the AE, we implemented two standard outlier-detection algorithms LOF 6 [12] and IF 7 [33], with an optional PCA, using the same normal-only training frames. This ensures direct comparability under uniform feature extraction and thresholding.

$$LOF_{MinPts}(p) = \frac{1}{|N_{MinPts}(p)|}\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)} \tag{6}$$

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}} \tag{7}$$

1) Flattening Spectrogram Frames and Label Conversion: Because *scikit-learn* expects 2D arrays, we flatten each (n_mels, frame_width) spectrogram frame to a single vector and convert labels $\{0,1\}$ to $\{1,-1\}$ (normal vs. anomaly) as illustrated in Listing 4:

```python
X_train = X_train_3d.reshape(len(X_train_3d), -1)
X_val = X_val_3d.reshape(len(X_val_3d), -1)
X_test = X_test_3d.reshape(len(X_test_3d), -1)

y_val = np.zeros(len(X_val_3d))
y_val = np.where(y_val == 0, 1, -1)
y_test= np.where(y_test_raw == 0, 1, -1)
```

Listing 4: Flattening Data and Converting Labels

2) Thresholding and Evaluation: We compute each method's outlier score on the validation set (normal-only) to pick a percentile-based threshold. Frames in the test set whose scores exceed this threshold are labeled anomalies. Listing 5 illustrates how we systematically search for an optimal threshold by trying multiple cutoffs:

```python
def find_best_thresh(scores, y_true, metric="f1",
    step_percentiles=(5, 95), num_steps=30):
    low_val  = np.percentile(scores[y_true==1], step_percentiles[0])
    high_val = np.percentile(scores[y_true==1], step_percentiles[1])
    thresholds = np.linspace(low_val, high_val, num_steps)

    best_thr = thresholds[0]
    best_val = -np.inf

    for thr in thresholds:
        preds = np.where(scores >= thr, 1, -1)
        rec   = recall_score(y_true,  preds, pos_label=-1)
        prec  = prec_score(y_true, preds, pos_label=-1)
        f1    = f1_score(y_true,       preds, pos_label=-1)

        if metric == "f1":        this_val = f1
        elif metric == "prec": this_val = prec
        elif metric == "recall":  this_val = rec
        else:                     this_val = f1

        if this_val > best_val:
            best_val = this_val
            best_thr = thr

    return best_thr, best_val
```

Listing 5: Threshold Selection and Evaluation

3) Integrating PCA and Manual Grid Search: Optionally,a PCA is applied (*e.g.*, 80–95% explained variance) before LOF or IF. A simple grid search adapted from [34], [35], illustrated in Listing 6, iterates over parameter sets (*e.g.*, number of neighbors for LOF) to find the highest validation F1. The best combination is then tested on the test data for final performance metrics (AUC, recall, precision, F1).

```python
def manual_grid_search(X_train, X_val, y_val, pca_ratios, model_type="LOF"):
    param_grid = {"n_neighbors": [5, 10]} if model_type == "LOF" else {"n_estimators": [50, 100]}

    for pca_ratio in pca_ratios:
        for params in _iterate_params(param_grid):
            steps = [("scaler", scaler)]
            if pca_ratio:
                steps.append(("pca", pca_ratio))
            steps.append(("classifier", create_model(model_type, params)))
            pipeline = Pipeline(steps).fit(X_train)

            val_scores = pipeline[-1].decision_function(pipeline[:-1].transform(X_val))
            thr, f1, _ = find_best_threshold(val_scores, y_val, metric="f1")

    return f1, thr
```

Listing 6: Simplified Grid Search with PCA

This pipeline allows classical methods to be directly compared with our AE's reconstruction-based approach, under the same preprocessing, validation-based thresholding, and test-set evaluation.

F. Anomaly Detection

Anomalies are identified by comparing test-sample scores (reconstruction errors or classical outlier-scores) against a percentile-based threshold derived from a normal-only validation set.

1) Reconstruction-Based Detection (AE):

- A fully connected AE is trained solely on normal frames, minimizing MSE between each input spectrogram frame and its reconstruction [20].
- For each test frame, the reconstruction error (MSE) is computed.
- A threshold, set between the 80^{th} and 97^{th} percentile of the validation reconstruction errors, determines whether a frame is anomalous.
- Frames exceeding this threshold are flagged as anomalies.

2) Outlier Methods (LOF, IF):

- In parallel, Local Outlier Factor (LOF) and Isolation Forest (IF) are trained on the same normal data.
- Each algorithm produces outlier scores for test frames; again, a percentile threshold on validation scores delineates normal vs. anomalous.
- These density-based or subspace methods provide a complementary perspective, facilitating direct benchmarking against the learned AE model.

G. Overview

The proposed methodology integrates advanced preprocessing, an Autoencoder architecture, and a density-based benchmark for robust anomaly detection. By leveraging datasets from both controlled and real-world environments, the framework addresses noise resilience, scalability, and generalizability. This approach lays the groundwork for scalable deployment in industrial settings.

IV. Experimental Evaluation

This Section presents the results of the proposed acoustic anomaly detection framework, evaluated on the washing machine, synthetic and real industrial machine datasets. All audio was captured with a *Samson Meteor MIC* at a 16 kHz sampling rate and stored in `.wav` format to ensure consistency and high-fidelity signal representation. Regarding the traditional machine learning approaches, PCA-based methods did not achieve competitive performance compared to LOF, potentially due to the loss of critical feature information during dimensionality reduction. Therefore, they are excluded from detailed discussion. Consequently, the following subsections will focus on the results achieved by AE, LOF, and IF.

A. Evaluation Metrics

To measure the performance of the framework and the benchmark, we applied the following evaluation metrics for each dataset:

1) ROC AUC Score [36]: The Area Under the Receiver Operating Characteristic (ROC) Curve 8 quantifies the ability of the model to distinguish between normal and anomalous data.

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}\big(\mathrm{FPR}^{-1}(\alpha)\big)\, d\alpha. \tag{8}$$

2) Precision, Recall, and F1-Score [37], [38]: Precision 9 measures the proportion of detected anomalies that are correct, while recall 10 captures the fraction of true anomalies detected. F1 11 is their harmonic mean, offering a balanced view of performance.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{9}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{10}$$

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{11}$$

3) Confusion Matrix [39]: Visualizes the performance of the classification, including true positives, false positives, true negatives, and false negatives to analyze trade-offs between missed and mistaken detections.

B. Washing Machine Anomaly Dataset

This dataset consists of washing-machine recordings under two conditions:

- *Normal Operation (19 minutes 49 seconds)*: The machine operates with clothes, producing acoustic signatures representative of standard machine behavior.
- *Induced Anomalies (1 minute 26 seconds)*: Irregular knocking sounds are simulated by operating the machine with shoes instead of clothes.

The controlled nature and clear labeling of these experiments makes this dataset an ideal starting point for developing and evaluating the proposed methodology.

1) Mel-Spectrogram Analysis: The Mel-spectrogram of the washing machine dataset as well as the normal and anomalous frames are shown in Figure 2. The complete spectrogram (a) reveals varying energy distribution, reflecting washing cycles. The anomalous frame (c) shows short high-energy bands in a low frequency range of below 200 HZ, which correspond to the knocking sounds caused by shoes inside the drum. Conversely, the normal frame (b) exhibits a constant energy band in the similar frequency range of below 500 HZ. However, the normal operation noise in this similar low-frequency range can complicate purely visual discrimination between normal and anomalous events.

2) Evaluation Metrics: The evaluation of AE, LOF, and IF on the washing machine dataset is summarized in Table I. The ROC AUC, precision, recall, and F1-score are presented for all three methods. The confusion matrices in Figure 3 provide a detailed breakdown of classification results for normal and anomalous frames.

TABLE I: Evaluation metrics for the washing machine dataset.

Method	ROC AUC	Precision	Recall	F1-Score
AE	0.731	0.536	0.494	0.514
LOF	0.988	0.912	0.980	0.945
IF	0.706	0.000	0.000	0.000

3) Observations: LOF achieves a high ROC AUC of 0.988 alongside excellent precision (0.912) and recall (0.980). This indicates that density-based anomaly detection is highly effective for the washing machine dataset, where anomalous events create local deviations from normal operating conditions. The AE attains a lower ROC AUC (0.731) and comparatively balanced but modest precision and recall. Reconstruction-based methods can be more sensitive to normal operating variations, leading to a higher rate of misclassification when anomalies mimic normal acoustic patterns. Isolation Forest (IF) underperforms significantly, achieving a ROC AUC of 0.706 but failing to detect any anomalies (precision and recall of 0.000). This suggests that IF is not suitable for this dataset, possibly due to its inability to capture the specific density variations introduced by the anomalous events in the washing machine data. For AE (Figure 3 a), there are notable misclassifications: a relatively large number of normal frames are flagged as anomalous (528), and many anomalous frames are predicted as normal (627). For LOF (Figure 3 b), both false positives and false negatives are drastically reduced, illustrating that LOF captures the subtle density shifts caused by the anomalous shoe "knocking" without over-detecting normal operational fluctuations. Despite the controlled nature of the washing machine dataset, where the anomalous sounds stand out in the low-frequency range, normal operation noise partly overlaps. LOF's ability to distinguish local density variations makes it more robust against this overlap, while the AE can struggle to reconstruct subtle differences if the anomaly partially resembles normal noise.

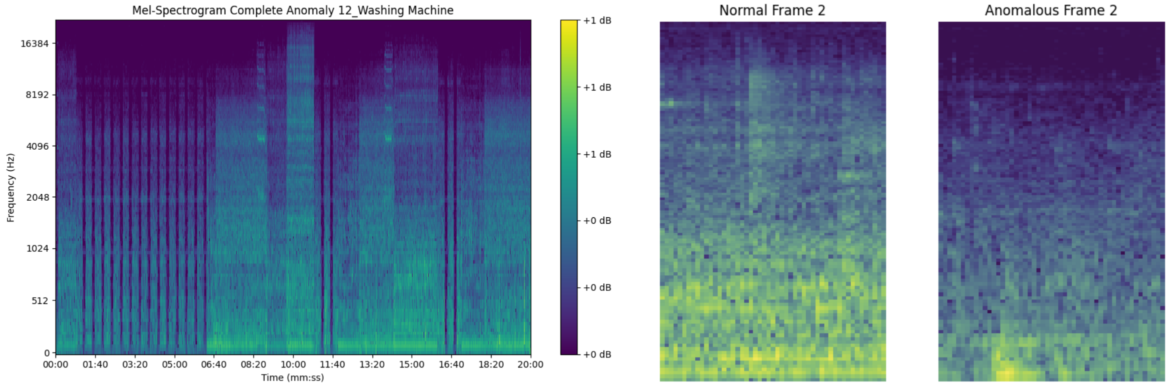
C. Industrial Environment with Simulated Anomalies

A second dataset was recorded in an operational industrial machine and comprises:

- *Normal Operation (12 minutes 46 seconds)*: Equipment functioning under routine conditions, reflecting regular machine activity.
- *Induced Anomalies (3 minutes 29 seconds)*: Artificial mechanical faults introduced via controlled impacts from a hammer striking a shovel, selected to resemble real-world structural issues yet remain safe and repeatable.

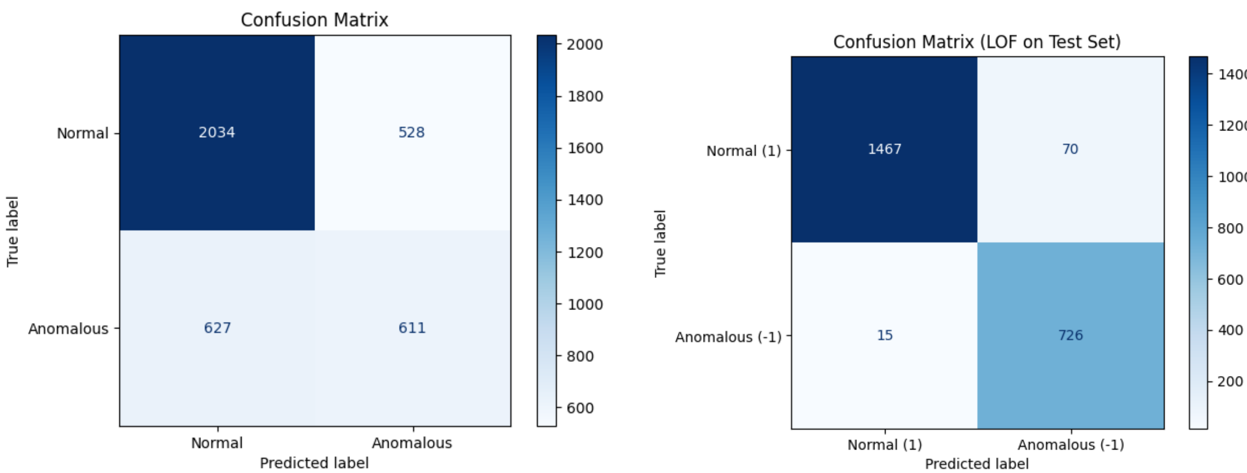
These synthetic knocks allow replication of realistic fault phenomena and thus form the basis for testing the framework under near–real-world conditions while preserving consistency in labeling.

1) Mel-Spectrogram Analysis: The Mel-spectrogram of the synthetic anomaly dataset as well as the normal and anomalous frames are shown in Figure 4. The complete spectrogram (a) reveals a constant energy distribution,



(a) Complete Washing Machine Spectrogram (b) Normal Frame (c) Anomaly Frame

Fig. 2: Mel-spectrograms of the washing machine dataset and comparison of normal vs. anomalous frame



(a) Autoencoder (AE) (b) Local Outlier Factor (LOF)

Fig. 3: Confusion matrices for the washing machine dataset. (a) AE. (b) LOF.

reflecting one mode of operation of the industrial machine. The normal frame (b) exhibits a constant energy band in a lower frequency range of mostly below 2000 HZ. Conversely, the anomalous frame (c) shows short high-energy bands in a high frequency range of above 2000 HZ, which correspond to the knocking sounds caused by hammer hitting a shovel. Therefore, the normal operation noise is clearly visually distinguishable from the anomalous sound.

2) Evaluation Metrics: Table II presents the evaluation metrics for AE, LOF, and IF on the synthetic anomaly dataset. Both AE and LOF achieved near-perfect results, as reflected in the ROC AUC, precision, recall, and F1-scores. IF also performed admirably, though slightly below AE and LOF. The confusion matrices in Figure 5 further highlight the strong classification performance.

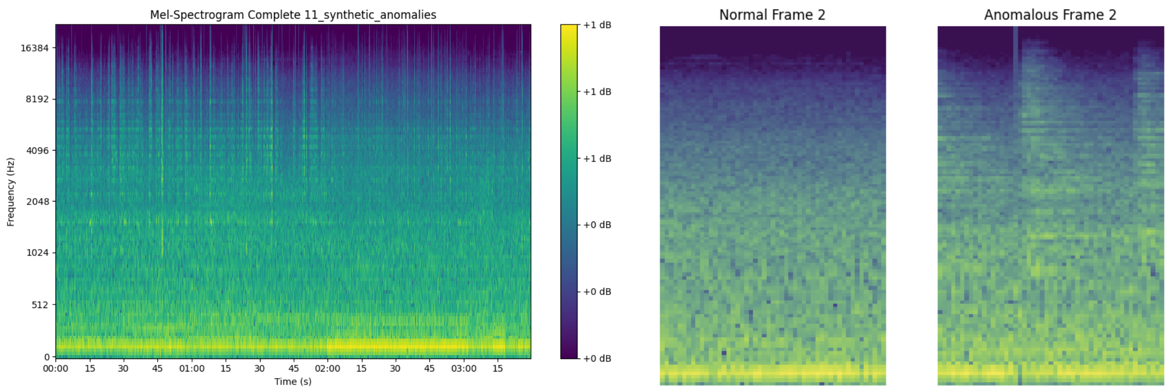
TABLE II: Evaluation metrics for the synthetic anomaly dataset.

Method	ROC AUC	Precision	Recall	F1-Score
AE	0.999	0.973	0.997	0.985
LOF	1.000	0.969	1.000	0.984
IF	0.994	0.971	0.977	0.974

3) Observations: The synthetic dataset, being highly structured and noise-free, allowed both AE and LOF to achieve near-perfect classification results. LOF demonstrated a slight edge in recall (1.000 vs. 0.997), whereas AE exhibited marginally higher precision (0.973 vs. 0.969) and F1-score (0.985 vs. 0.984). The differences, however, are minimal. Isolation Forest (IF) also performed exceptionally well with a ROC AUC of 0.994 and high precision (0.971) and recall (0.977). Although slightly below AE and LOF, IF remains a strong performer on this dataset, indicating its capability to detect high-frequency anomalies in a controlled environment. The controlled nature of the synthetic anomalies highlights the strengths of both methods in detecting high-frequency patterns but may not fully reflect real-world challenges.

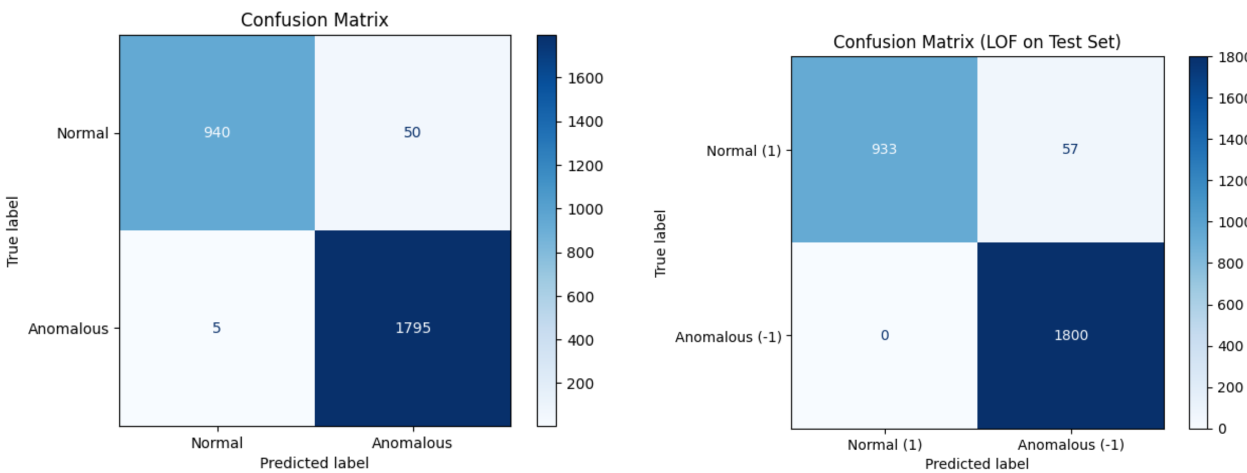
D. Industrial Environment with Real Anomalies

A third dataset was recorded in an operational industrial machine and comprises:



(a) Synthetic Anomaly Spectrogram (b) Normal Frame (c) Anomaly Frame

Fig. 4: Mel-spectrograms of synthetic anomaly dataset and normal vs anomalous frames.



(a) Autoencoder (AE) (b) Local Outlier Factor (LOF)

Fig. 5: Confusion matrices for the synthetic anomaly dataset. (a) AE. (b) LOF.

- *Normal Operation (59 minutes 53 seconds)*: Equipment functioning under routine conditions, reflecting regular machine activity.
- *Real Anomalies (5 seconds)*: Real anomalies produced by the machine.

These real knocks form the basis for testing the framework under real-world conditions whilst introducing inconsistencies in labeling.

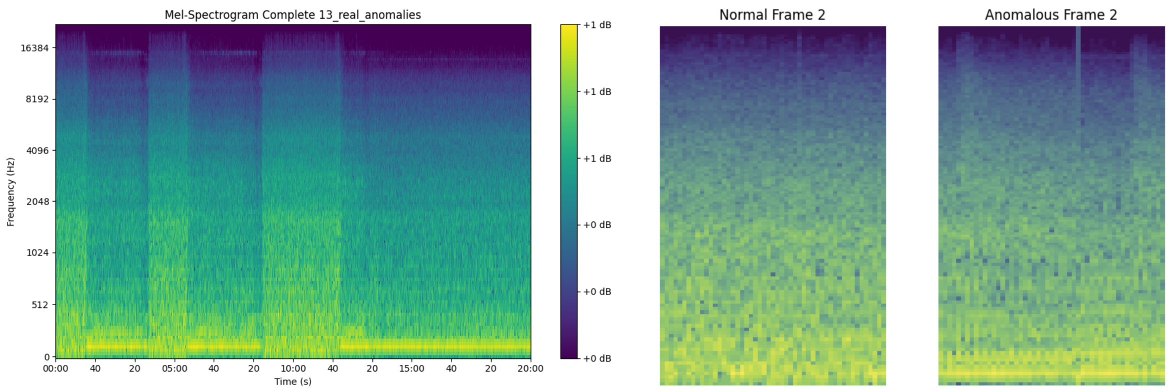
1) Mel-Spectrogram Analysis: The Mel-spectrogram of the real anomaly dataset as well as the normal and anomalous frames are shown in Figure 6. The complete spectrogram (a) shows different operation modes of the industrial machine which is reflected in the changing energy distribution. The normal frame (b) exhibits a constant energy band in a lower frequency range of mostly below 2000 HZ. Conversely, the anomalous frame (c) shows short high-energy bands in a high frequency range of above 2000 HZ, which correspond to the knocking sounds caused by a real anomaly. The high frequency energy pattern is similar to the synthetic anomaly in Figure 4. However, the intensity of the real anomaly is less pronounced compared to the synthetic anomaly.

TABLE III: Evaluation metrics for the real anomaly dataset.

Method	ROC AUC	Precision	Recall	F1-Score
AE	0.992	0.218	0.977	0.356
LOF	1.000	0.162	1.000	0.278
IF	0.856	0.049	0.279	0.083

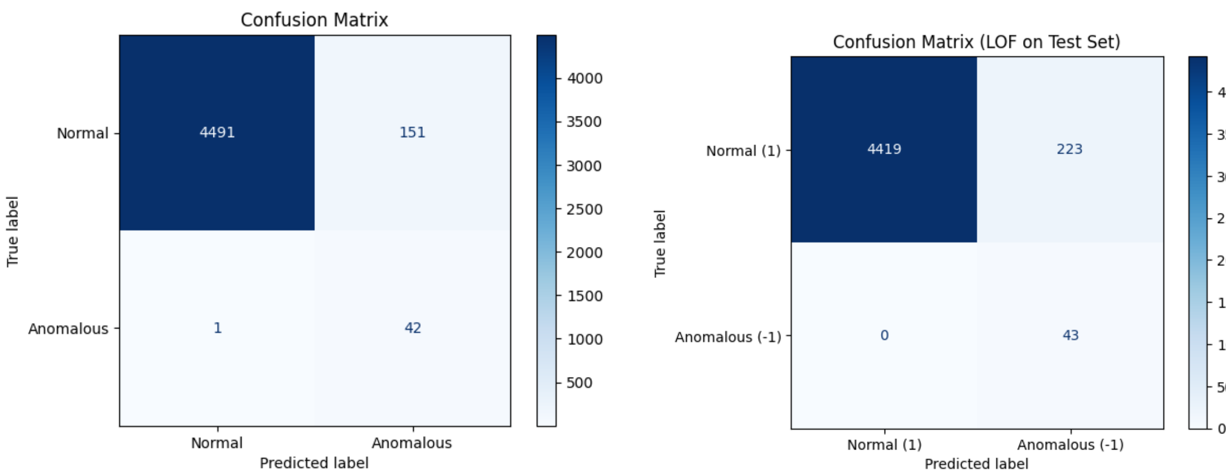
2) Evaluation Metrics: Table III presents the evaluation metrics for AE, LOF, and IF on the real anomaly dataset. Both AE and LOF achieve high performance, reflected in the ROC AUC and recall. IF, however, shows considerably lower performance compared to AE and LOF. The confusion matrices in Figure 7 illustrate these results in more detail.

3) Observations: Both AE and LOF show strong detection capabilities on the real anomaly dataset, with near-perfect recall (AE at 0.977, LOF at 1.000) and high ROC AUC (0.992 for AE and 1.000 for LOF). Isolation Forest (IF) exhibits significantly lower performance, achieving a ROC AUC of 0.856 with minimal precision (0.049) and low recall (0.279). This indicates that IF struggles to effectively detect real



(a) Real Anomaly Spectrogram (b) Normal Frame (c) Anomaly Frame

Fig. 6: Mel-spectrograms of real anomaly dataset and normal vs anomalous frames.



(a) Autoencoder (AE) (b) Local Outlier Factor (LOF)

Fig. 7: Confusion matrices for the real anomaly dataset. (a) AE. (b) LOF.

anomalies in this dataset, likely due to the subtlety and reduced intensity of the anomalous signals compared to synthetic anomalies. AE achieves higher precision (0.218) than LOF (0.162), which contributes to a higher F1-score (0.356 vs. 0.278). LOF's perfect recall indicates it detects all anomalies but at the expense of more false positives (lower precision), while AE occasionally misses an anomaly but makes fewer incorrect anomaly predictions overall. The real anomaly's high-frequency signatures are detected reliably by both methods, though reduced signal intensity compared to the synthetic case highlights the practical importance of microphone placement and noise mitigation strategies.

E. Comparative Insights

Both AE and LOF achieved near-perfect scores on the synthetic anomaly dataset, reflecting its noise-free nature and clearly distinguishable high-frequency bursts. In contrast, the washing machine dataset posed more challenges due to overlapping normal and anomalous sounds. Here, LOF outperformed AE (ROC AUC of 0.988 vs. 0.731), indicating that LOF is more adept at capturing subtle deviations under real-world acoustic variability, whereas AE is more prone to misclassifications when anomalies closely resemble normal conditions.

On the real anomaly dataset, both methods again demonstrated strong detection capabilities, yet their trade-offs became clearer. LOF achieved perfect recall but at the cost of more false positives (lower precision), while AE missed a small number of anomalies but incurred fewer false alarms. Isolation Forest (IF), however, showed limited effectiveness on this dataset, with significantly lower ROC AUC, precision, and recall compared to AE and LOF. Consequently, LOF's high recall is useful for applications where missing an anomaly is unacceptable, whereas AE's higher precision may reduce the burden of investigating false alarms. This distinction parallels their behavior in the washing machine dataset, though with overall higher precision and recall in the real anomaly setting compared to the more complex washing machine noise patterns.

Across all three datasets, the gap between AE and LOF is smallest on the relatively clean synthetic data. For noisier scenarios like the washing machine data and to some extent the real anomaly data LOF appears more robust to overlapping frequencies and operational variability. Isolation Forest consistently underperforms in these more complex and noisy environments, highlighting its limitations in handling subtle and less pronounced anomalies. Meanwhile, AE excels in more structured or less noisy environments, which results in simpler interpretability of reconstruction errors.

Finally, the consistent high-frequency signature of the synthetic anomalies somewhat equals the real anomalies produced in the industrial machine. However, because the real anomalies are less pronounced, microphone placement and noise reduction prove crucial. Overall, the complementary strengths of reconstruction-based (AE) and density-based (LOF) approaches suggest that a hybrid system combining both methods could enhance reliability and performance in complex industrial environments with varying noise levels.

V. Summary, Conclusions, and Future Work

A. Summary

We introduced a comprehensive framework for acoustic anomaly detection aimed at industrial machinery, utilizing Mel-spectrogram normalization alongside a fully connected Autoencoder (AE) and a density-based Local Outlier Factor (LOF) algorithm. Additionally, we incorporated Isolation Forest (IF) and Principal Component Analysis (PCA) as comparative benchmark. The framework was evaluated on three distinct datasets: a washing machine anomaly dataset, a synthetic industrial machinery anomaly dataset, and a real-world industrial machinery anomaly dataset. Our experiments demonstrated that LOF consistently outperformed IF and PCA in traditional machine learning approaches, particularly excelling in scenarios with overlapping normal and anomalous acoustic patterns. Notably, both AE and LOF achieved near-perfect classification metrics on the synthetic dataset. In contrast, LOF maintained superior performance on the washing machine dataset (F1 score: AE 0.514 vs. LOF 0.945) and AE showed stronger performance on the real industrial dataset (F1 score: AE 0.356 vs. LOF 0.278). These results underscore the effectiveness of density-based methods in handling diverse and noisy real-world environments, as well as the complementary strengths of reconstruction-based approaches in structured settings. The implementations of our proposed framework and experimental evaluations are publicly available on GitHub: https://github.com/frodo-hu/Acoustic-Anomaly-Detection.

B. Conclusions

Small and unbalanced datasets can hinder model performance and require careful planning or data augmentation. There is a performance gap when transitioning from synthetic data to real-world recordings, highlighting the need for robust preprocessing and model architectures tailored to variable operating conditions. Even minor variations in recording conditions or machine states can lead to misclassifications. Hence, precise data cleaning is essential for maximizing anomaly detection performance. Overall, in the structured, low-noise setting of the synthetic dataset, both AE and LOF achieved near-perfect classification metrics, underscoring their suitability for detecting clear-cut anomalies. IF consistently underperformed across all datasets, highlighting its limitations in detecting subtle anomalies among variable operational noise. In the washing machine dataset, which is relatively small and imbalanced, LOF significantly outperformed AE, highlighting LOF's robustness to overlapping and variable acoustic conditions. Conversely, in real industrial machine scenarios with short, high-frequency burst anomalies that are more clearly distinguishable from normal operation sounds, AE performed stronger than LOF. An additional advantage of the AE approach is its potential to be streamlined for real-time detection. In scenarios, especially where anomalies are more distinctly separated from normal conditions, such as high-frequency bursts in industrial machinery, a pre-trained AE can quickly flag anomalies without excessive computational overhead compared to LOF. These findings emphasize the complementary strengths of AE and LOF: AE excels in cleaner or more distinct acoustic conditions, whereas LOF remains highly effective across diverse and noisier settings. This complementary nature suggests that a hybrid approach combining AE and LOF could leverage the high precision of AE and the high recall of LOF, thereby enhancing overall anomaly detection reliability in diverse industrial environments.

C. Future Work

Future work could focus on enhancing scalability and noise resilience by exploring hybrid learning strategies that integrate reconstruction- and density-based approaches. Additionally, expanding the framework to utilize larger and more diverse real-world datasets will help validate and refine the models. Investigating advanced feature extraction techniques, such as convolutional neural networks for Mel-spectrograms, could further improve anomaly detection accuracy. Finally, the integration of this framework into real-time on-site monitoring systems could strengthen predictive maintenance by enabling immediate anomaly alerts resulting in reduced downtime and less failures in industrial settings.

Acknowledgments

This work was supported by the School of Computer Science at the University of St. Gallen. We extend our special thanks to our supervisor, Professor Bruno Rodrigues, for his continual support and guidance throughout this project.

References

[1] Siemens, "The true cost of downtime: Identify, avoid, and overcome," Erlangen, Germany, 2023, digital Industries, Customer Services. [Online]: https://assets.new.siemens.com/siemens/assets/api/uuid:3d606495-dbe0-43e4-80b1-d04e27ada920/dics-b10153-00-7600truecostofdowntime2022-144.pdf

[2] R. Mobley, "An introduction to predictive maintenance," *Elsevier Science google schola*, Vol. 2, pp. 485–520, 2002.

[3] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers Industrial Engineering*, Vol. 150, p. 106889, 2020. [Online]: https://www.sciencedirect.com/science/article/pii/S0360835220305787

[4] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers Industrial Engineering*, Vol. 137, p. 106024, 2019. [Online]: https://www.sciencedirect.com/science/article/pii/S0360835219304838

[5] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.

[6] M. Achouch, M. Dimitrova, K. Ziane, S. Sattarpanah Karganroudi, R. Dhouib, H. Ibrahim, and M. Adda, "On predictive maintenance in industry 4.0: Overview, models, and challenges," *Applied Sciences*, Vol. 12, No. 16, 2022. [Online]: https://www.mdpi.com/2076-3417/12/16/8081

[7] M. Khanjari, A. Azarfar, M. H. Abardeh, and E. Alibeiki, "Anomalous sound detection for machine condition monitoring using 3d tensor representation of sound and 3d deep convolutional neural network," *Multimedia Tools and Applications*, Vol. 83, No. 15, pp. 44 101–44 119, 2024. [Online]: https://link.springer.com/article/10.1007/s11042-023-17043-9

[8] E. Di Fiore, A. Ferraro, A. Galli, V. Moscato, and G. Sperlì, "An anomalous sound detection methodology for predictive maintenance," *Expert Systems with Applications*, Vol. 209, p. 118324, 2022. [Online]: https://www.sciencedirect.com/science/article/pii/S0957417422014518

[9] F. König, C. Sous, A. Ouald Chaib, and G. Jacobs, "Machine learning-based anomaly detection and classification of acoustic emission events for wear monitoring in sliding bearing systems," *Tribology International*, Vol. 155, p. 106811, 2021. [Online]: https://www.sciencedirect.com/science/article/pii/S0301679X20306368

[10] F. Lo Scudo, E. Ritacco, L. Caroprese, and G. Manco, "Audio-based anomaly detection on edge devices via self-supervision and spectral analysis," *Journal of Intelligent Information Systems*, Vol. 61, No. 3, pp. 765–793, 2023. [Online]: https://link.springer.com/article/10.1007/s10844-023-00792-2

[11] Y. Tagawa, R. Maskeliūnas, and R. Damaševičius, "Acoustic anomaly detection of mechanical failures in noisy real-life factory environments," *Electronics*, Vol. 10, No. 19, 2021. [Online]: https://www.mdpi.com/2079-9292/10/19/2329

[12] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *ACM SIGMOD Record*, Vol. 29, No. 2, pp. 93–104, 2000. [Online]: https://dl.acm.org/doi/10.1145/335191.335388

[13] H. Wang, D. Wang, H. Liu, and G. Tang, "A predictive sliding local outlier correction method with adaptive state change rate determining for bearing remaining useful life estimation," *Reliability Engineering System Safety*, Vol. 225, p. 108601, 2022. [Online]: https://www.sciencedirect.com/science/article/pii/S0951832022002460

[14] J. Ma, Z. Teng, Q. Tang, W. Qiu, Y. Yang, and J. Duan, "Measurement error prediction of power metering equipment using improved local outlier factor and kernel support vector regression," *IEEE Transactions on Industrial Electronics*, Vol. 69, No. 9, pp. 9575–9585, 2022.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: A survey," *ACM Computing Surveys (CSUR)*, Vol. 41, No. 3, pp. 1–58, 2009.

[16] T. B. Duman, B. Bayram, and G. İnce, "Acoustic anomaly detection using convolutional autoencoders in industrial processes," *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019)*, F. Martínez Álvarez, A. Troncoso Lora, J. A. Sáez Muñoz, H. Quintián, and E. Corchado, Eds. Springer International Publishing, 2020, pp. 432–442. [Online]: https://link.springer.com/chapter/10.1007/978-3-030-20055-8_41

[17] B. Bayram, T. B. Duman, and G. Ince, "Real time detection of acoustic anomalies in industrial processes using sequential autoencoders," *Expert Systems*, Vol. 38, No. 1, p. e12564, 2021. [Online]: https://doi.org/10.1111/exsy.12564

[18] S. Yu, X. Li, L. Zhao, and J. Wang, "Hyperspectral anomaly detection based on low-rank representation using local outlier factor," *IEEE Geoscience and Remote Sensing Letters*, Vol. 18, No. 7, pp. 1279–1283, 2021.

[19] A. Hussein, "Exploring-aad: Audio anomaly detection in machine sounds using chroma and mel frequency features," https://github.com/AHruler/Exploring-AAD, 2021, gitHub repository.

[20] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, Vol. 72, No. 2011, pp. 1–19, 2011.

[21] H. Hojjati and N. Armanfard, "Self-supervised acoustic anomaly detection via contrastive learning," *ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3253–3257. [Online]: https://ieeexplore.ieee.org/document/9746207

[22] M. Meire and P. Karsmakers, "Comparison of deep autoencoder architectures for real-time acoustic based anomaly detection in assets," *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, Vol. 2, 2019, pp. 786–790. [Online]: https://ieeexplore.ieee.org/document/8924301

[23] Y. Peng, X. Zhong, X. Yang, and L. Hu, "Detection of abnormal sound of power plant equipment fault based on self-supervised learning," *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2022, pp. 174–178. [Online]: https://ieeexplore.ieee.org/document/9873544

[24] A. Spournias, N. Nanos, E. Faliagka, C. Antonopoulos, N. Voros, and G. Keramidas, "Enhanced sound recognition and classification through spectrogram analysis, mems sensors, and pytorch: A comprehensive approach," *Collaborative Computing: Networking, Applications and Worksharing*, H. Gao, X. Wang, and N. Voros, Eds. Springer Nature Switzerland, 2024, pp. 3–17. [Online]: https://link.springer.com/chapter/10.1007/978-3-031-54521-4_1

[25] M. Portnoff, "Time-frequency representation of digital signals and systems based on short-time fourier analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 1, pp. 55–69, 1980.

[26] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python." *SciPy*, 2015, pp. 18–24.

[27] L. Cohen, *Time-frequency analysis*. Prentice Hall PTR New Jersey, 1995, Vol. 778.

[28] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 4, pp. 357–366, 1980.

[29] S. Malcolm, "Auditory toolbox version 2," *https://engineering. purdue. edu/˜ malcolm/interval/1998-010/*, 1998.

[30] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the influence of normalization/transformation process on the accuracy of supervised classification," *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 729–735.

[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]: http://www.deeplearningbook.org

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. [Online]: https://arxiv.org/abs/1412.6980

[33] M. Tokovarov and P. Karczmarek, "A probabilistic generalization of isolation forest," *Information Sciences*, Vol. 584, pp. 433–449, 2022. [Online]: https://www.sciencedirect.com/science/article/pii/S0020025521010999

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, Vol. 12, p. 2825–2830, 2011. [Online]: http://jmlr.org/papers/v12/pedregosa11a.html

[35] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomputing*, Vol. 415, pp. 295–316, 2020. [Online]: https://www.sciencedirect.com/science/article/pii/S0925231220311693

[36] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, Vol. 27, No. 8, pp. 861–874, 2006.

[37] Y. Yang, “An evaluation of statistical approaches to text categorization,” *Information Retrieval*, Vol. 1, No. 1-2, pp. 69–90, 1999.

[38] D. M. Powers, “Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation,” *Journal of Machine Learning Technologies*, Vol. 2, No. 1, pp. 37–63, 2011.

[39] Yanminsun, A. Wong, and M. S. Kamel, “Classification of imbalanced data: a review,” *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 23, 11 2011.

Appendix

Code Implementation

The implementations of our proposed framework and experimental evaluations are publicly available on GitHub: https://github.com/frodo-hu/Acoustic-Anomaly-Detection.

Contribution Statement

Both authors contributed equally to the research, design, execution, and writing of this research.

Aids

Aid	**Usage**	**Affected Part**
ChatGPT	Text reduction, Grammar correction	Complete Paper
Zotero	Compilation of the literature index	Literature Index

Declaration of Authorship

“We hereby declare,

- that we have written this thesis independently,
- that we have written the thesis using only the aids specified in the index;
- that all parts of the thesis produced with the help of aids have been declared;
- that we have handled both input and output responsibly when using AI. we confirm that we have therefore only read in public data or data released with consent and that we have checked, declared and comprehensibly referenced all results and/or other forms of AI assistance in the required form and that we are aware that we are responsible if incorrect content, violations of data protection law, copyright law or scientific misconduct (e.g. plagiarism) have also occurred unintentionally;
- that we have mentioned all sources used and cited them correctly according to established academic citation rules;
- that we have acquired all immaterial rights to any materials we may have used, such as images or graphics, or that these materials were created by us;
- that the topic, the thesis or parts of it have not already been the object of any work or examination of another course, unless this has been expressly agreed with the faculty member in advance and is stated as such in the thesis;
- that we are aware of the legal provisions regarding the publication and dissemination of parts or the entire thesis and that we comply with them accordingly;
- that we are aware that my thesis can be electronically checked for plagiarism and for third-party authorship of human or technical origin and that we hereby grant the University of St.Gallen the copyright according to the Examination Regulations as far as it is necessary for the administrative actions;
- that we are 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 our title.”

By submitting this thesis, we confirm through our conclusive action that we are submitting the Declaration of Authorship, that we have read and understood it, and that it is true.