Sound-Based Fault Detection for Machines

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by

Subhamkumar Sharma	**20204210**
Raman Jeenager	**20204163**
Sarthak Penta	**20204178**
Sohan Gond	**200204206**

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD PRAYAGRAJ
December, 2023

UNDERTAKING

I declare that the work presented in this report titled "*Sound-Based Fault Detection for Machines*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

December, 2023
Allahabad

Subhamkumar Sharma	20204210
Raman Jeenager	20204163
Sarthak Penta	20204178
Sohan Gond	200204206

CERTIFICATE

Certified that the work contained in the report titled "*Sound-Based Fault Detection for Machines*", by Subhamkumar Sharma 20204210, Raman Jeengar 20204163, Sarthak Penta 20204178, Sohan Gond 200204206, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Anil Kumar Singh)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

December, 2023

Acknowledgements

I would like to extend my heartfelt gratitude to the individuals and organizations whose significant contributions have been integral to the successful completion of this project.

Foremost among them is my project supervisor, Professor Anil Kumar Singh. I am deeply appreciative of their unwavering guidance, unparalleled expertise, and invaluable feedback, which have played a crucial role in shaping the direction and enhancing the overall quality of this work. Their patience, consistent support, and encouragement have been instrumental throughout the project, for which I am truly thankful.

Contents

Abstract

In recent years, there has been notable momentum in utilizing machine learning for identifying faults in machinery, particularly by analyzing sound. This approach capitalizes on the inherent acoustic signatures produced during machinery operation, utilizing machine learning algorithms to discern patterns indicative of potential faults. By extracting meaningful features from sound data and training models to recognize abnormalities, this method enables early fault detection, contributing to improved maintenance strategies and reduced downtime. This project explores the integration of machine learning and sound analysis, emphasizing its applications, benefits, and challenges in the context of industrial machinery fault detection.

Chapter 1

Introduction

The current approach to identifying machine faults relies heavily on manual inspections and periodic maintenance routines. While these methods have been fundamental in routine checks, they exhibit significant limitations, particularly in detecting subtle or early signs of faults. This deficiency can lead to unforeseen and major breakdowns, posing operational challenges for manufacturing processes. The inherent delay in identifying faults in the current approach underscores the need for a more proactive strategy.

The limitations of the current approach become apparent when waiting until a problem becomes evident before taking corrective action. This reactive stance can result in increased downtime, affecting overall production efficiency, and may incur higher repair costs due to the severity of the identified issues. To address these shortcomings, there is a growing recognition of the potential of sound as a diagnostic tool in assessing machine health.

Recognizing the unique sounds machines produce provides an innovative and efficient means of identifying anomalies. This sound-based diagnostic approach offers the advantage of early fault detection, allowing for a proactive response to prevent more serious problems. The ability to discern abnormal sounds before they escalate into critical issues can significantly improve maintenance planning and operational continuity.

Furthermore, the integration of machine learning techniques in fault detection processes adds a layer of sophistication to the system. Machine learning excels in recognizing complex patterns and abnormalities in sound that may be challenging for traditional methods. By leveraging machine learning algorithms, the diagnostic capabilities are enhanced, leading to a more advanced and precise fault detection system. This incorporation of technology not only improves the overall efficiency of fault detection but also contributes to the evolution of predictive maintenance strategies in the industrial landscape. As industries increasingly embrace the digitization of processes, the combination of sound-based diagnostics and machine learning emerges as a promising avenue for enhancing machinery reliability and reducing unplanned downtime.

Chapter 2

Dataset Info

We are using MIMII dataset which can be served as a valuable data resource for analysis of malfunctioning of industrial machine. We have used three different types of machine valves, pumps and fans. A solenoid valve is an electromechanical device controlling fluid flow. It utilizes an electrically operated solenoid to actuate a valve, regulating the passage of liquids or gases. When energized, the solenoid opens or closes the valve, making it essential in automation, refrigeration, and other applications requiring precise fluid control. A pump is a mechanical apparatus designed to propel fluids, such as liquids or gases, by creating either flow or pressure. A fan is a mechanical device with rotating blades that creates airflow, providing ventilation and cooling in various applications, from homes to industrial settings. The sounds produced by fans and pumps often exhibit a relatively constant and steady-state pattern over time, making them more characteristic of stationary sounds. These devices typically generate continuous, repetitive noises as they operate. Valves, on the other hand, can produce non-stationary sounds, meaning the sound characteristics change over time. This might be due to the dynamic nature of the valve operation, such as the opening and closing of the valve, leading to variations in sound patterns.

The test dataset consisted of all identified anomalous segments, while an equal number of normal segments were randomly chosen to form another part of the test dataset. The remaining normal segments were designated as the training dataset.

Machine Type	**Model ID**	**Segments**		**Total**
		Normal Condition	**Anomalous Condition**	
Fan	00	1000	400	1400
	01	1000	340	1340
	02	1000	350	1350
Pump	00	1000	140	1140
	01	700	100	800
	02	1000	110	1110
Valve	00	950	110	1060
	01	1000	120	1120
	02	700	120	820
Total		**8350**	**1790**	10140

Table 1: Dataset content details

Chapter 3

Proposed Method

The proposed methodology comprises three pivotal components: preprocessing, the generation of Mel-scaled spectrograms, and the subsequent classification process as shown in figure 1.

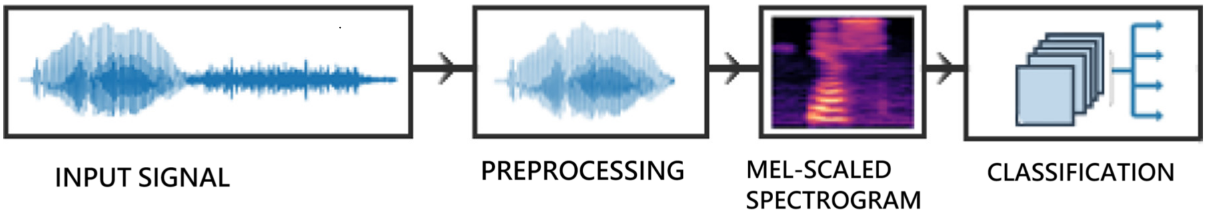

Figure 1: Project workflow overview

3.1 Preprocessing

Audio preprocessing is a fundamental stage in refining raw audio data for subsequent analysis. Within our project, this phase involves several critical operations. Firstly, the loading and initial processing of .wav files are conducted to ensure the data is in a suitable format for downstream tasks. Handling multi-channel audio data is a key aspect of this preprocessing step, where the objective is to extract specific channels effectively. This extraction process, similar to demultiplexing, allows for the isolation of individual sources within the audio data. The significance lies in enabling targeted analysis and enhancing computational efficiency. Throughout this preprocessing, mechanisms for error handling are implemented to maintain the integrity of the data. By focusing on the extraction and refinement of mono data, this stage establishes a solid foundation for subsequent feature extraction and in-depth analysis

```
def demux_wav(wav_name, channel=0):
    """
    demux .wav file.

    wav_name : str
        target .wav file
    channel : int
        target channel number

    return : numpy.array( float )
        demuxed mono data

    Enabled to read multiple sampling rates.

    Enabled even one channel.
    """
    try:
        multi_channel_data, sr = file_load(wav_name)
        if multi_channel_data.ndim <= 1:
            return sr, multi_channel_data

        return sr, numpy.array(multi_channel_data)[channel, :]

    except ValueError as msg:
        logger.warning(f'{msg}')
```

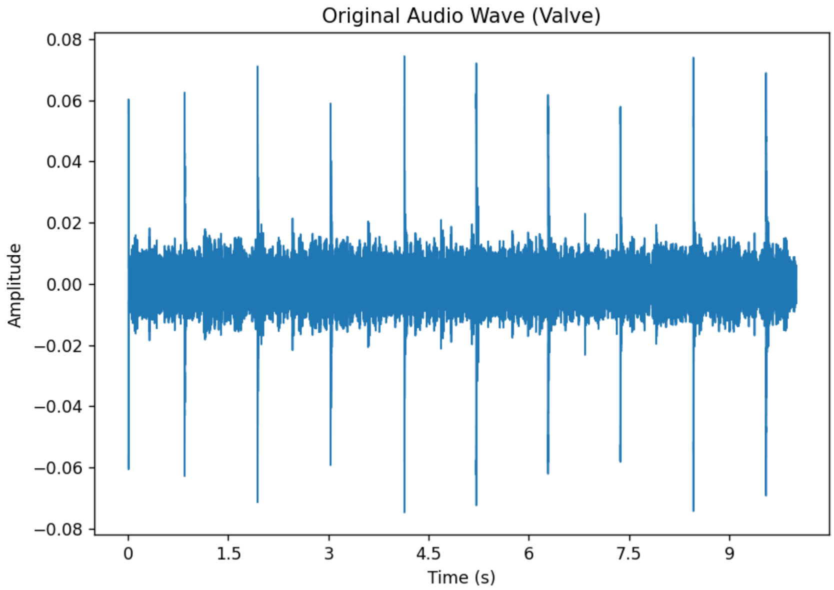

Figure 2: Before Preprocessing

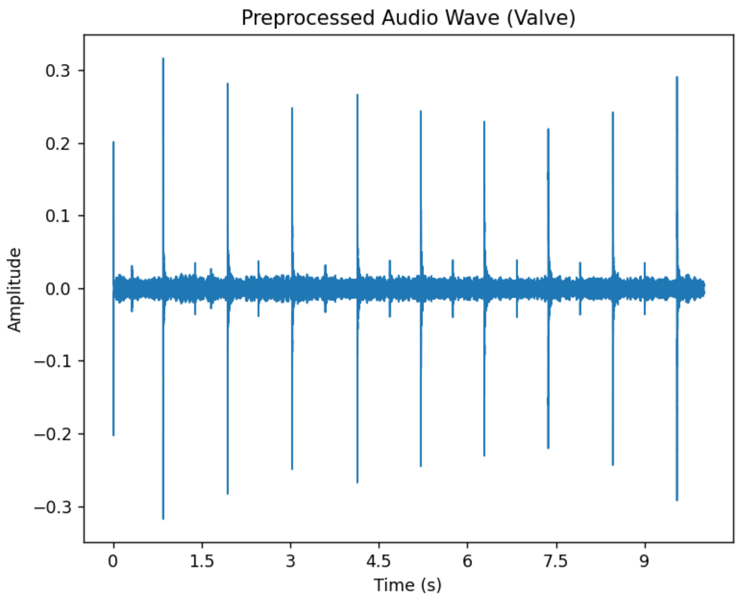

Figure 3: After Preprocessing

3.2 Spectrogram generation

In the subsequent phase of our project, we transform preprocessed audio data into a structured format by utilizing a log-Mel spectrogram. Initially, the raw audio waveform is converted into a two-dimensional image called a spectrogram, representing frequencies over time. Understanding that the human ear perceives frequencies logarithmically, we shift the frequency scale to mel-scale, transforming the regular spectrogram into a mel-spectrogram. This careful process involves specific parameters, such as a frame size of 1024, a hop size of 512, and 64 mel filters, with the assistance of tools like the librosa library. The resulting mel spectrogram, aligned with human auditory perception, is then subjected to a logarithmic transformation. The final step involves creating feature vectors, grouping frames of the log mel spectrogram, producing a flexible dataset that encapsulates the unique qualities of the audio signals. This adaptable approach allows us to fine-tune parameters for various audio datasets and specific analysis needs, culminating in a well-organized dataset for insightful audio analysis in our project analysis.

```
def file_to_vector_array(file_name, n_mels=64,
frames=5, n_fft=1024, hop_length=512, power=2.0):
    """
    convert file_name to a vector array.
    file_name : str
        target .wav file
    return : numpy.array( numpy.array( float ) )
        vector array
        * dataset.shape = (dataset_size, fearture_vector_length)
    """
    # 01 calculate the number of dimensions
    dims = n_mels * frames
    # 02 generate melspectrogram using librosa
    (**kwargs == param["librosa"])
    sr, y = demux_wav(file_name)
    mel_spectrogram = librosa.feature.melspectrogram(y=y,
    sr=sr,  n_fft=n_fft, hop_length=hop_length,
    n_mels=n_mels,  power=power)
    # 03 convert melspectrogram to log mel energy
    log_mel_spectrogram = 20.0 / power *
    numpy.log10(mel_spectrogram + sys.float_info.epsilon)
    # 04 calculate total vector size
    vectorarray_size=len(log_mel_spectrogram[0, :])- frames + 1
    # 05 skip too short clips
    if vectorarray_size < 1:
        return numpy.empty((0, dims), float)
    # 06 generate feature vectors by concatenating multi_frames
    vectorarray = numpy.zeros((vectorarray_size, dims), float)
    for t in range(frames):
        vectorarray[:, n_mels * t: n_mels * (t + 1)]
        = log_mel_spectrogram[:, t: t + vectorarray_size].T
    return vectorarray
```

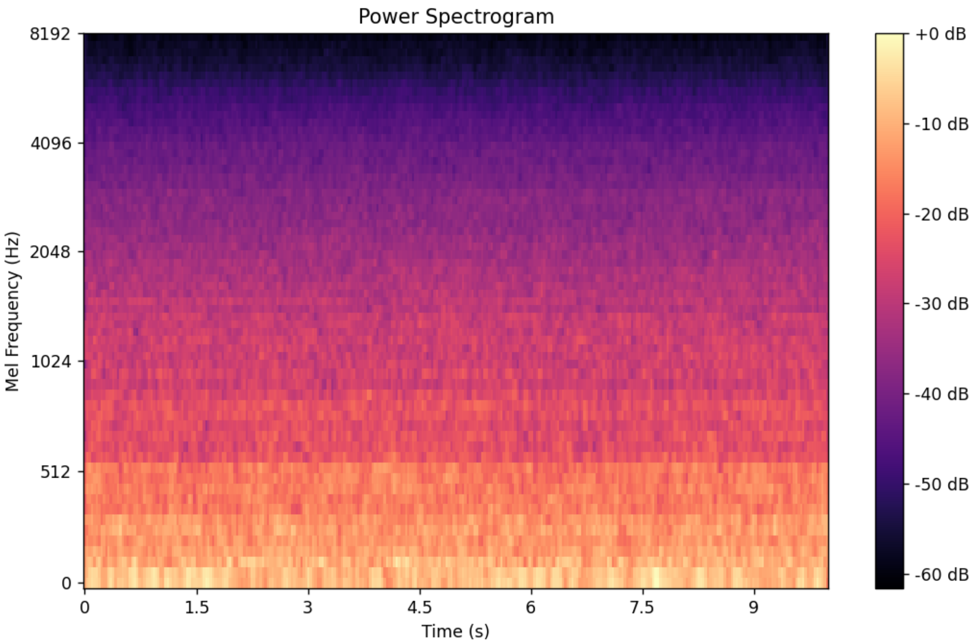

Figure 4: Fan

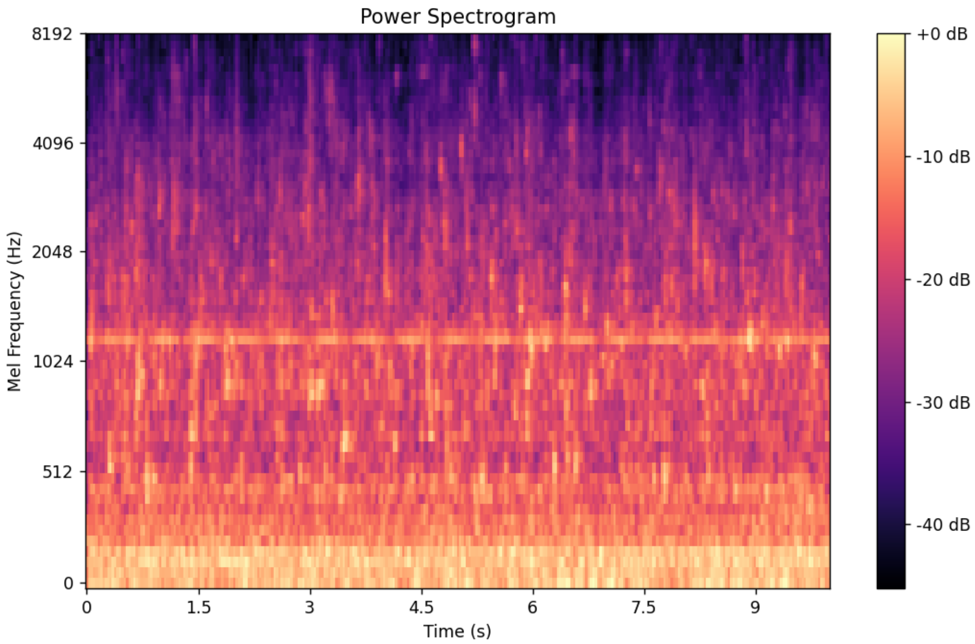

Figure 5: Pump

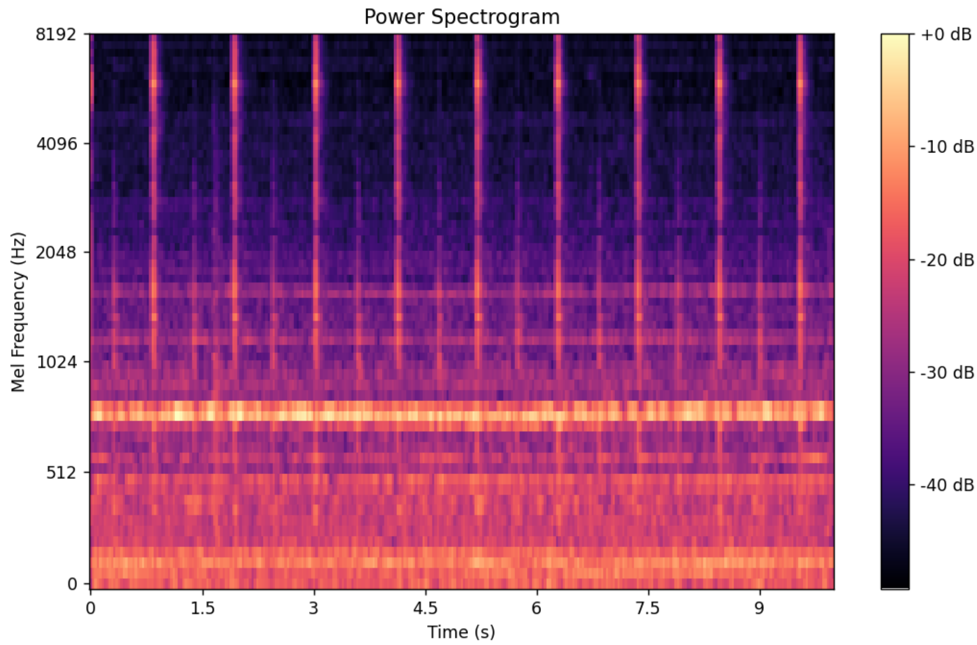

Figure 6: Valve

3.3 Classification Using Autoencoder

We have used an autoencoder to classify the abnormal sound from normal sound. An autoencoder represents a category of artificial neural networks employed in unsupervised learning. Its primary purpose is to learn efficient representations of data, typically by encoding the input into a lower-dimensional space and then decoding it back to the original data. The structure comprises both an encoder and a decoder. We have implemented autoencoder model using the Keras library, with a specific architectural structure. The model comprises an input layer with a shape defined by the input dimension (inputDim). The encoder section consists of three dense layers: the first with 64 neurons and a Rectified Linear Unit (ReLU) activation function, the second with 64 neurons and ReLU activation, and the third with 8 neurons and ReLU activation. Similarly, the decoder section mirrors the encoder with three dense layers: two with 64 neurons and ReLU activation, and the final layer with neurons equal to the input dimension, featuring a linear (no activation) function. The output layer maintains the same dimension as the input layer. In essence, this symmetric autoencoder aims to learn a condensed representation of the input data in an 8-dimensional space, subsequently reconstructing the input based on this representation.

The Keras autoencoder model implemented in our project is instrumental in the realm of fault detection within machine sounds. The architecture of the model follows a simple dense autoencoder structure with distinct layers, and it is specifically tailored to process Mel spectrograms obtained from audio data. In the training phase, the model is provided with a dataset that consists of typical machine sounds encountered during regular operations. Anomalies or faults are deliberately excluded from this dataset. The objective is for the autoencoder to learn a compressed representation of these normal sounds. The trained model is then employed for anomaly detection using a separate dataset that includes instances of machine sounds with faults. The autoencoder serves as a feature extractor, with the compressed layer acting as a bottleneck. Anomalies are identified by assessing the reconstruction error, which is the disparity between the input and its reconstructed version. A predetermined threshold is used to determine whether an input is classified as anomalous based on this error.

```
def keras_model(inputDim):
"""
define the keras model
the model based on the simple dense auto encoder (64*64*8*64*64)
"""
inputLayer = Input(shape=(inputDim,))
h = Dense(64, activation="relu")(inputLayer)
h = Dense(64, activation="relu")(h)
h = Dense(8, activation="relu")(h)
h = Dense(64, activation="relu")(h)
h = Dense(64, activation="relu")(h)
h = Dense(inputDim, activation=None)(h)
return Model(inputs=inputLayer, outputs=h)
```

Chapter 4

Results Analysis

For every machine type and model ID, the segments were divided into a training dataset and a test dataset. The test dataset comprised all the anomalous segments, and an equivalent number of randomly chosen normal segments were also included in the test dataset. The remaining normal segments were used as the training dataset. Separate autoencoders were trained for each machine type and model ID using the training dataset consisting only of normal segments. Anomaly detection was conducted for each segment by setting a threshold on the averaged reconstruction error over ten seconds. The area under the curve (AUC) values were calculated for the test dataset of each machine type and model ID. Additionally, different levels of signal-to-noise ratio (SNR) were taken into account, such as 6 dB, 0 dB, and -6 dB.

It is evident from the analysis that the AUC values for valves are consistently lower compared to the other machines. This disparity can be attributed to the nature of the sound signals emitted by valves, which are nonstationary, impulsive, and sparse in time. As a result, the averaged reconstruction error tends to be small, making it challenging to accurately detect anomalies for valves. On the other hand, detecting anomalies for fans is relatively easier due to the stationary nature of their sound signals.

Furthermore, it is worth noting that for certain machine models, the AUC values decrease rapidly as the noise level increases. This observation highlights the need to address the degradation caused by non-stationarity and noise in order to improve the performance of unsupervised anomalous sound detection.

Machine Type	**Model ID**	**Input SNR (-6 dB)**	**Input SNR (0 dB)**	**Input SNR (6 dB)**
Valve	00	0.62	0.51	0.65
	01	0.54	0.62	0.60
	02	0.55	0.58	0.64
Avg.		0.57	0.57	0.63
Pump	00	0.57	0.61	0.79
	01	0.90	0.96	0.98
	02	0.55	0.49	0.45
Avg.		0.67	0.69	0.74
Fan	00	0.57	0.64	0.82
	01	0.59	0.72	0.90
	02	0.65	0.82	0.99
Avg.		0.60	0.73	0.90

Table 2: AUCs for All Machines

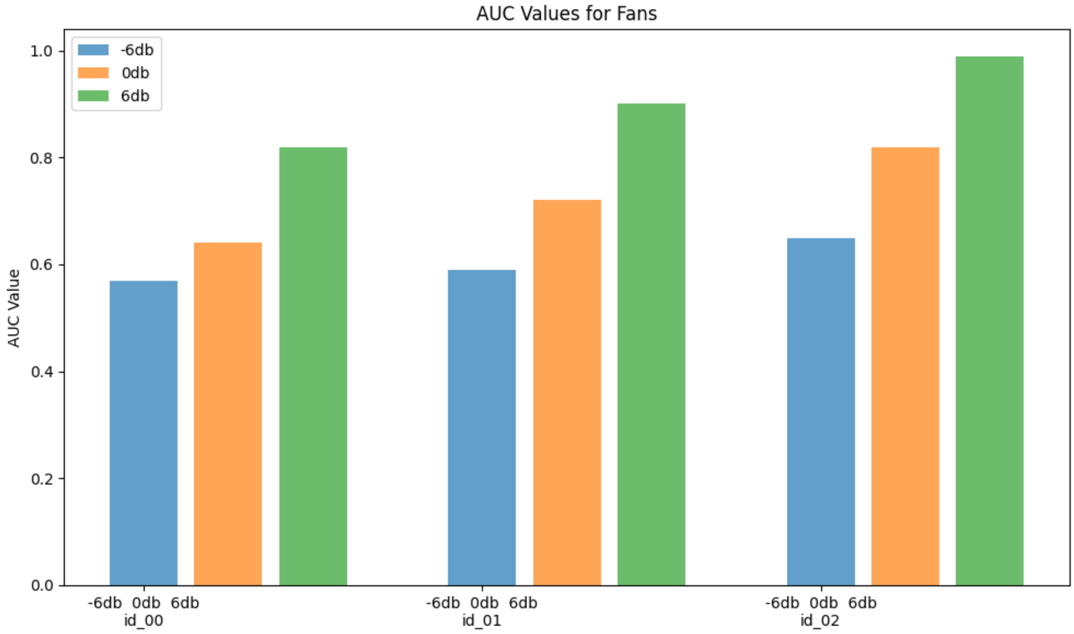

Figure 7: AUCs for Fan

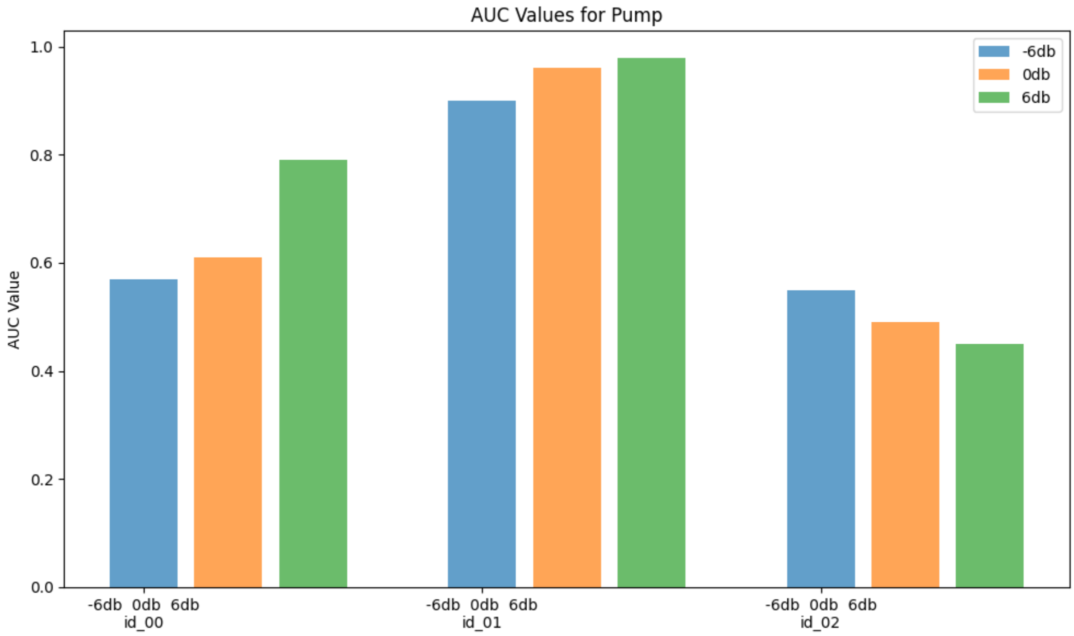

Figure 8: AUCs for Pump

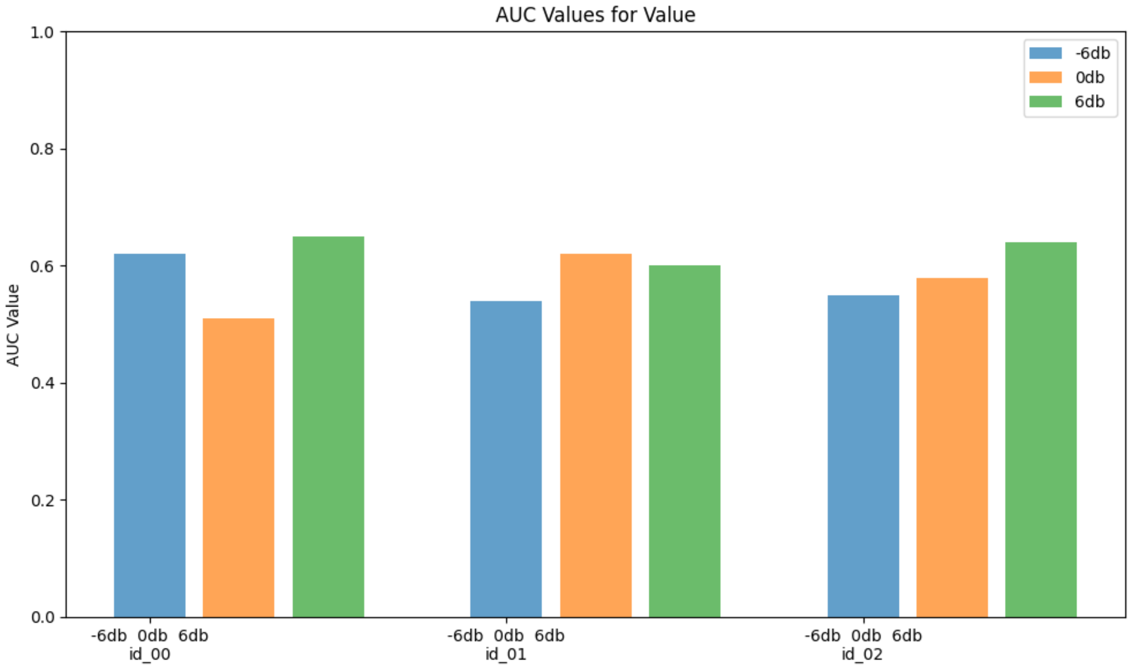

Figure 9: AUCs for Valve

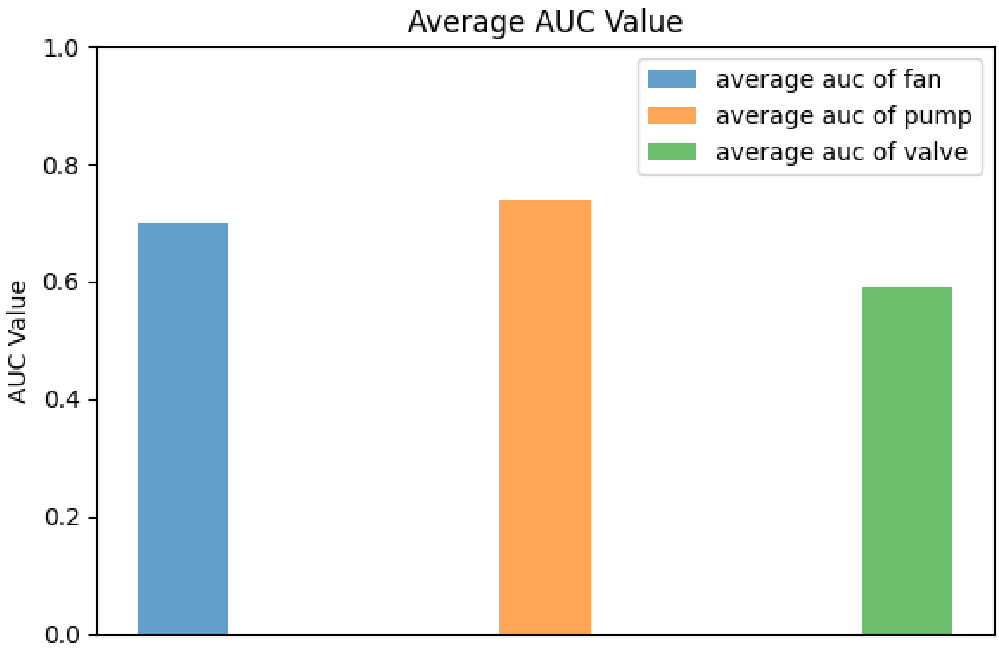

Figure 10: Average AUCs for Fan, Pump and Valve

Chapter 5

Conclusion and Future Work

In conclusion, our evaluation of an autoencoder-based unsupervised anomalous sound detection system revealed two key challenges: non-stationary machine sound signals and noise. The study demonstrated the autoencoder's promise in handling non-stationary signals but emphasized the need for robust preprocessing to tackle noise. Tackling these challenges is essential to improve the efficacy of unsupervised anomaly detection in real-world applications featuring diverse and dynamic machine-generated audio data.

In the future, we plan to improve how we detect problems in machines by combining audio data with other types of information like vibrations. We want to make our fault detection more accurate. We will explore using advanced technologies like recurrent neural networks and attention mechanisms to better understand how things change over time and make our models work even better. Our primary objective is to develop systems capable of swiftly detecting issues in machines as they occur, enabling continuous monitoring. We'll work on making our models not just good at finding faults but also at figuring out exactly where and what kind of issues are happening. Lastly, we'll check if we can use these fault detection models on smaller devices located near the machines (edge devices) to process information on the spot. This could reduce the need to send a lot of audio data to a central server.

Reference

Study of machine fault diagnosis system using neural networks [1].
A Hydraulic Pump Fault Diagnosis Method Based on the Modified Ensemble Empirical Mode Decomposition and Wavelet Kernel Extreme Learning Machine Methods[2].
Deep Transfer Learning for Machine Diagnosis: From Sound and Music Recognition to Bearing Fault Detection[3].