Anomalous sound pattern detection for machine health monitoring

Shivali Dalmia dalm3066@stthomas.edu
Manjeet Rege rege@stthomas.edu
University of St. Thomas, St. Paul MN 55105, USA

Abstract. Anomaly detection using audio signals from industrial machines in the manufacturing industry has gained broad interest over the last few years. For example, predictive maintenance solutions utilize raw analog signals to identify trends and patterns. In a few scenarios, an engineer working in a factory setting can tell when a machine is behaving abnormally just by hearing unexpected sounds (e.g., the loudness of sound) that are well within the human perceivable frequency range (20Hz - lowest pitch to 20 kHz - highest pitch) which are typically concentrated in a narrow range of frequencies and amplitudes. The human perception of the amplitude of a sound is its loudness. However, the audio signal in its raw form is not always the best representation of the important features (e.g., frequencies, amplitude, peaks). Additionally, the machine learning applications which rely on using traditional digital signal processing techniques (e.g., digital signal processors, chips) have a lot of dependency on subject matter experts to tune the system for a better performance. Thus, we investigate how the digital transformation of waveform signals from microphone sensors (e.g., Audio recordings of industrial pumps, valves, slide rails) into Spectrograms[1] can help to monitor machine health (e.g., anomaly classification). In the pre-processing phase, raw audio signals (.WAV format) from each machine are converted to Mel Spectrogram images using short-term Fourier transformation. Then, comparative study of image classification techniques using deep convolutional neural networks (CNN) with and without data augmentation, is conducted to classify images as normal or abnormal. The approach is evaluated using Malfunctioning industrial machine investigation and inspection dataset [1,2,3] (MIMII dataset). Results show that the neural network based models trained on the dataset with Mel Spectrogram transformation perform better than models trained on the raw dataset (i.e., sound samples without spectrogram conversion).
Keywords: Anomaly detection, Machine health monitoring, Preventive maintenance

1 Introduction

Industrial artificial intelligence or industrial AI [12] is the application of AI to industrial use cases. AI technologies can process and interpret large volumes of data coming from the production floor to identify novel, interesting, and relevant patterns to help detect anomalies in production processes in real-time, and more. For example, intelligent, self-optimizing machines automate production processes and predictive maintenance (PdM) [11] using sensors. PdM solutions heavily utilize sensors for data collection, giving us the ability to measure movement, waves, sound, heat, light, and much more. The sensors enable us to catalog analog data from the physical world, and identify trends and patterns. The data collected from different sensors such as, microphone, accelerometer, magnetometer, gyroscope, and temp-humidity which are mounted on machines can play a vital role in building a predictive maintenance and failure monitoring system for industrial machines.

1.1 Why Industrial AI?

Every manufacturer aims to save and make money, reduce risks and improve overall production efficiency. AI tools can process and interpret large volumes of data from the production floor to identify novel, interesting, and relevant patterns to help detect anomalies in production processes in real-time. Some noteworthy examples are as follows:

- Intelligent, self-optimizing machines that automate production processes. For example, Siemens and Google have partnered to develop a computer vision application that can visually inspect products to ensure quality and is further equipped with algorithms that can predict the wear-and-tear on assembly line machines [13].
- Predictive maintenance using sensors. For example, Bosch Rexroth's [14] predictive maintenance solution creates a machine health index which can provide an accurate picture of a drive system's normal state.

Outline: The rest of the paper is organized as follows: In section 2 we provide a brief overview of the related work, our contribution and the scope of this paper. Section 3 describes the exploratory data analysis and data preprocessing. Section 4 describes the experiments conducted for anomaly detection for different industrial machines and section 5 concludes the paper and summarizes the future work.

[1] A spectrogram is a photograph/voiceprint of a signal which expresses an audio signal as an image using different colors to indicate the amplitude or strength of each frequency.

2 Related Work and Our Contributions

There are three main classes of anomaly detection techniques: unsupervised, semi-supervised, and supervised. The majority of supervised algorithms are able to detect pre-defined anomalies, for example certain error types on industrial manufacturing products. In the case of severe class imbalance, where only a small fraction of training data represents anomaly, there is not enough data available for the model to learn unseen anomaly patterns. This limits the scope of improvement in model performance and prediction. The remedy is to use unsupervised or semi-supervised anomaly detection techniques. For example, the approach of using data-driven lower dimensional representation of Spectrograms for training [5] is a semi-supervised approach. Another example of an unsupervised approach is AnoGAN [6] based on image mapping.

M.Abdel et al. [5] proposed a technique for anomaly detection in civil aircraft engines where spectrograms provide a visual representation of the vibrations of engines. As the spectrogram representation is noisy and high dimensional they propose to use two types of low dimensional representations of the spectrograms: a data driven dictionary learning (Non-negative matrix factorization similar to PCA) and a non adaptive dictionary representations which are generated by Fourier basis and other wavelets base or frames.

In another work, Thomas et al. [6] proposed AnoGAN, a deep convolutional generative adversarial network to learn a manifold of normal anatomical variability, accompanying a novel anomaly scoring scheme based on the mapping from image space to a latent space. On test data, the model labels anomalies, and scores image patches indicating their fit into the learned distribution. Hendrycks et al. [7] introduced an abnormality module which had a normal classifier and an auxiliary decoder. The normal classifier was trained as a baseline for detecting misclassified and out of distribution examples in neural networks which utilized probabilities from softmax distributions. The performance of normal binary classifier was evaluated using MNIST, CIFAR-10, and CIFAR-100 datasets. The appended auxiliary decoder reconstructed the input from the normal classifier stage. The results demonstrated that the abnormality module was better than the baseline normal classifier.

Akcay et al. [8] introduced a novel anomaly detection model which used a conditional generative adversarial network which jointly learnt the generation of high-dimensional image space and the inference of latent space. The model architecture employed encoder-decoder-encoder sub-networks. The generative network enabled the model to map the input image to a lower dimension vector, which was then used to reconstruct the generated output image. The use of the additional encoder network maps the generated image to its latent representation. Minimizing the distance between these images and the latent vectors during training helped in learning the data distribution for the normal samples. As a result, a larger distance metric from this learned data distribution at inference time was indicative of an outlier from that distribution i.e., an anomaly. Shahid Khan et al. [9] proposed a SDCNN framework based on deep convolutional neural networks which utilized network spectrogram images generated using the short-time Fourier transform. The experimental results improved the accuracy by 2.5% - 4% in detecting intrusions (for e.g. malicious attacks by third party networks which are either novel or the mutations of older attacks) as compared to other deep learning algorithms such as CNN-1D, RNN, LSTM, and GRU.

Table 1. Summary of related work

Year	Author	Proposed approach	Evaluation dataset
2023	M.Abdel et al.	Data driven lower dimensional representation of Spectrograms	Civil aircraft engine data
2017	Thomas Schlegl et al.	AnoGAN	Spectral-domain OCT scans
2016	Dan Hendrycks et al.	Probabilistic softmax distributions	CIFAR-10, CIFAR-100, THCHS-30, IMDb Movie Reviews
2018	Samet Akcay et al.	GANomaly	CIFAR-10
2021	A. S. Khan et al.	SDCNN	CIC-IDS2017

Limitations of related work: Table 1 summarizes the deep learning based related work proposed in recent years (2016 - 2021) for anomaly detection which were briefly described earlier. We note that the majority of the works used images for training their models. However, these approaches have not been tested for anomaly detection in the manufacturing industry. For example, the linear spectrograms [9] which are a direct output of short-term Fourier Transform may or may not be able to highlight the striking differences between normal and anomaly conditions of an industrial machine. Thus the need arises for identifying techniques for generating fine-tuned spectrograms for a better performance of deep learning based anomaly detection algorithms.

2.1 Our contributions

Our main contributions are as follows:

1. We propose a two-step data preprocessing strategy for converting raw audio signals to spectrograms. In Step-1 we use short-term Fourier Transform (STFT) [15] to see the change in different frequency

components over time within the audio signal and increase the granularity of the visual representation. In Step-2 we optimize the STFT output using mel and db scale. *"Mel scale is a perpetual scale of pitches judged by listeners to be equidistant from one another. The frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal spectrum"* [16, 17]. This results in a better representation of the sound in human perceivable range. *Db scale* represents the third dimension indicating the amplitude of a particular frequency at a particular time. Audio is best represented with a logarithmic amplitude axis in decibels [18].

2. We build a neural network based model to evaluate the effectiveness of the optimized Mel-spectrograms for normal for the task of anomaly prediction. We use Malfunctioning industrial machine investigation and inspection dataset [1, 2, 3] (MIMII dataset) to build the models.
3. We compare the anomaly prediction model trained on Mel Spectrogram data with the models trained on the raw dataset (i.e., normal sound samples).

Scope: To simplify the data transformation, we only use the first channel of the recordings. Further we also acknowledge that due to time constraints the comparison with the related work is limited.

3 Proposed Methodology

In this section we first briefly describe the basic concepts. Then, we describe the dataset used for the investigation which is followed by the details of exploratory data analysis. Then we describe the data preprocessing steps with illustrative examples. Finally, we describe the convolutional neural network based approach to train the models.

3.1 Basic Concepts

A **sound signal** [19] is produced by variations in air pressure and it is measured by plotting the intensity of pressure variations over time. The audio signals from industrial machines when they are operating in normal condition follow a repetitive pattern occurring at a regular interval so that each wave has the same shape. The height shows the intensity of the sound and is known as the amplitude and the time taken to complete one full wave pattern is called the period. Figure 1 shows a representation of amplitude vs. time period in a signal. Frequency represents the number of waves made by a signal over a period of one second and is measured in Hertz (Hz).

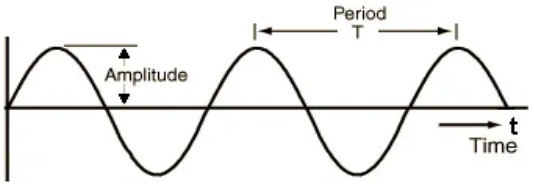

Figure 1. Representation of amplitude vs. time in a signal [21]

The **spectrum** is a set of frequency bands which if combined together represents one signal. It is a representation of amplitude against frequency where the x-axis represents the range of frequency values, at a moment in time, it represents the signal in the frequency domain rather than the time domain.
A **Spectrogram** [16] is a concise snapshot of an audio wave over time. It can be referred to as a photographic representation of the signal where the x-axis represents time and the y-axis represents frequency. Different colors indicate the amplitude or strength of each frequency band. The brightness of color is directly proportional with the energy of the signal. They are generated using fast fourier transformation (FFT) [20]. FFT decomposes the signal into its constituent frequencies and displays the amplitude of each frequency present in the signal. FFT chops up the duration of the signal into smaller time segments and then applies the Fourier transformation to each segment which determines the frequencies in that segment.

3.2 Dataset description

In this work, we use Malfunctioning industrial machine investigation and inspection (MIMII) dataset [1, 2, 3]. The dataset has been used as a benchmark for sound based fault diagnosis in industrial machines. The dataset consists of normal and anomalous operating sounds of four machine types: valves, pumps, fan, and slide rails. Each recording contains a multi-channel audio file with an approximate length of 10 seconds. The recording contains a target machine's operating sound and environmental noise collected using eight microphones with

16kHz sampling rate and 16 bit per sample. To simplify the data transformation, we only use the first channel of the recordings. All recordings are regarded as single-channel recordings of a fixed microphone.

The data is categorized by machine type and machine ID. For each machine ID around 1000 samples of normal sounds are available for training and 200 - 400 samples for the test. Further, the test data contains a similar number (100 - 200) of the normal and anomalous sounds. In summary, each training and test sample has the metadata of machine type, machine ID, and condition (i.e., normal or anomalous). Figure 2 shows an example of the dataset file structure.

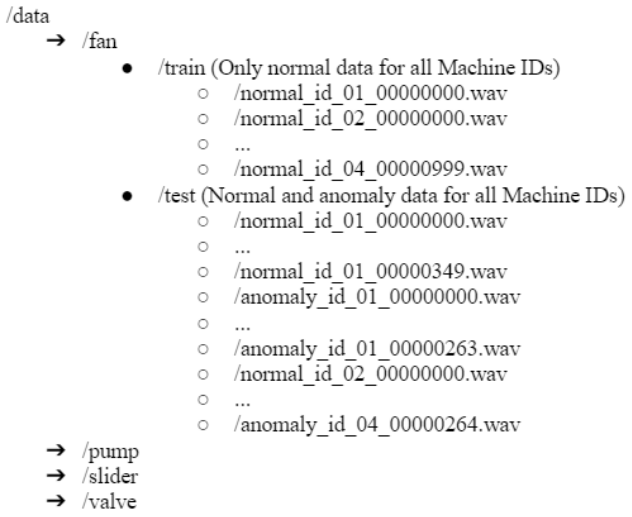

Figure 2. Example of dataset structure for fan machine.

3.3 **Exploratory Data Analysis**

Exploratory data analysis (EDA) is an essential step to understand data characteristics. The analysis helps in identifying the useful patterns, determining important variables, and identifying the necessary data transformation or preprocessing steps required to best represent the data before using it for model training.

As observed in Figure 3 for the human eye it's difficult to distinguish normal and anomalous samples by just observing waveform representations of normal and anomaly samples for all four machines i.e., pump, slider, valve, slider etc. The noise change overs are not evident. This may be because in industrial scenarios the sound pattern from the machines may not follow a consistent regular periodic pattern, instead it may be composed of different frequencies and represent a more complex composite signal.

Thus, we performed additional data transformation steps to generate a distinguished visual representation between the normal and anomalous machine conditions. Mel-Spectrograms with additional preprocessing are leveraged here to better represent the noise change overs for different machines. For example, for a pump machine the noise changeovers are slighlty more evident in anomaly spectrograms as compared to normal spectrograms (Figure 3). With additional pre-processing (of spectrograms) the noise can be observed more clearly as illustrated in Figure 5 and 6 (pg. 7) for the pump machine. Similar observations are made for other three types of machines i..e., slider, valve, and fan.

We further note that with deep learning, the traditional audio processing techniques such as filtering, equalization, noise, suppression, compression which require mandatory inputs of subject matter experts are not required, because the data in its raw form is not used for model training. Instead the audio samples when converted to images are expected to be more noticeable and clearly perceived by electrical and manufacturing engineers working in factory plants. The conversion of audio signals into images is done by generating spectrograms using audio signals.

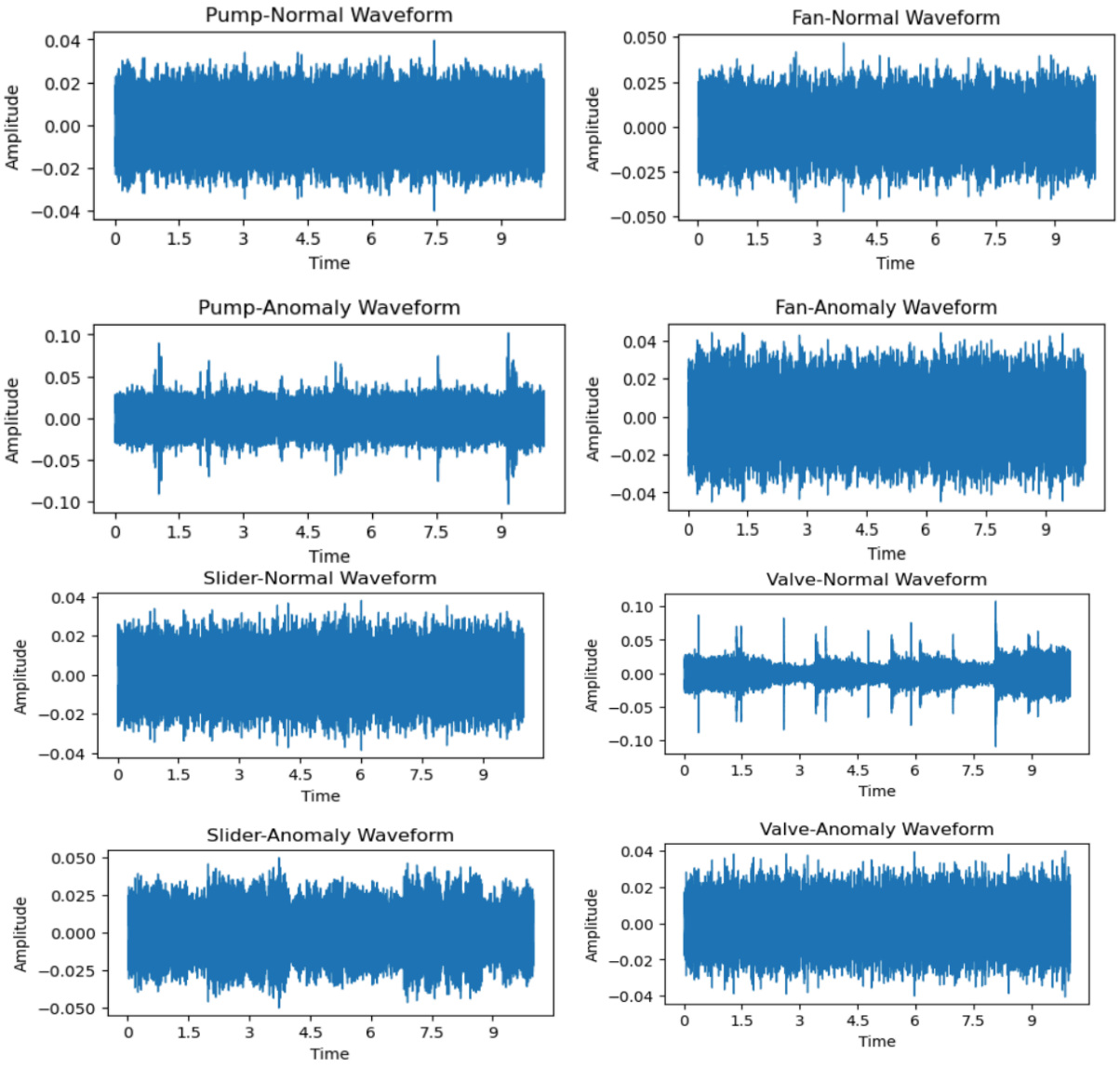

Figure 3. Waveform representation of Normal and Anomalous samples for machines.

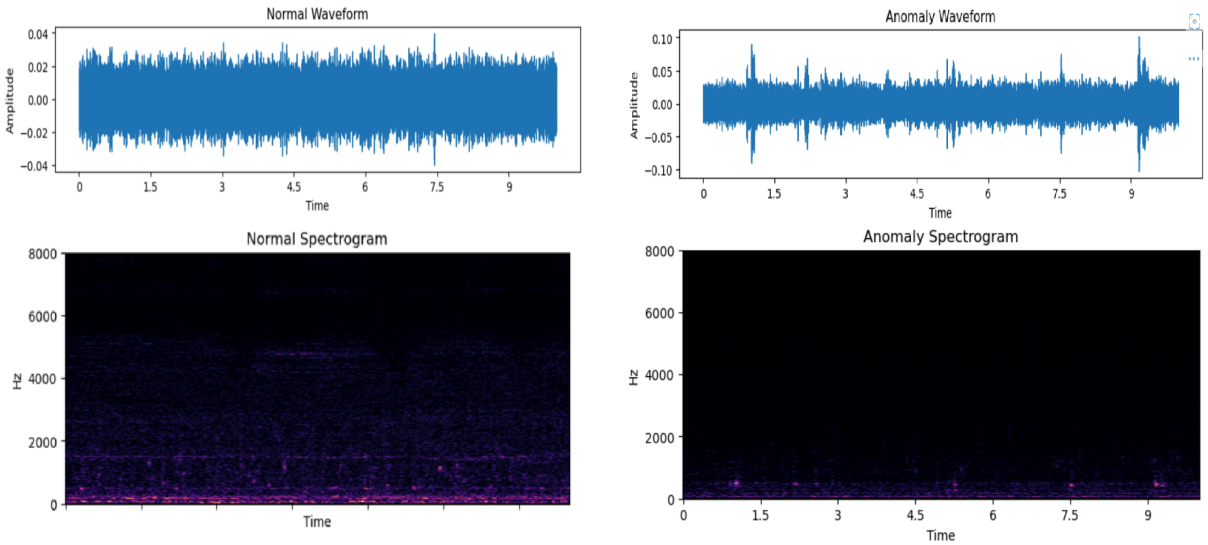

Figure 4. Waveform vs. Linear Spectrogram representation for Normal and Anomalous samples (pump).

3.4 Data Pre-processing

Below, we describe the pre-processing steps to convert an audio file to an optimized mel-spectrogram.

Mel-Spectrograms: Humans are more receptive to differences within lower frequencies than within higher frequencies and hear them on a logarithmic scale rather than linear. As observed in Figure 4, a linear spectrogram doesn't give much insight into different noise patterns in signals for machines such as pumps. Thus to overcome this issue Mel-Spectrograms [17] are generated. Mel-spectrograms use mel scale which is the perceptual scale of pitches judged by listeners to be equal in distance from one another. The reference point between Mel scale and normal frequency measurement is defined by assigning a perceptual pitch of 1000 mels to a 1000 Hz tone, 40 dB above the listeners threshold. Additionally, the human perception of sound is by its loudness which is heard logarithmically rather than linearly. The decibel scale is used to represent the loudness of a signal in spectrograms. Overall, linear spectrograms are converted to mel-spectrograms by using mel-scale on y-axis instead of frequency and decibel scale instead of amplitude to indicate colors.

Optimized Mel-Spectrogram: Mel-Spectrograms are optimized to increase the granularity of the distinction between different frequency bands in a spectrogram image. Short Time Fourier Transformation (STFT) is applied to determine the change in frequency components over time i.e., which part of the signal has low frequencies and which has high frequencies. STFT breaks up the audio signal into smaller sections by using a sliding time window. This splits the signal into sections along the time axis. It takes FFT on each section and then combines them. It is thus able to capture the variations of the frequency with time.

Further, STFT also splits the signal into sections along the frequency axis. It takes a full range of frequencies and divides it up into equally spaced bands (Mel scale). Then, for each time section, it calculates the amplitude or energy for each frequency band.

Mel Hyper-parameters: Mel-spectrograms are generated using the following parameters.

1. Minimum frequency (fmin)
2. Maximum frequency (fmax)
3. Number of frequency bands (n_mels – height of the spectrogram) are frequency hyperparameters
4. Sliding window length (n_fft)
5. Number of samples by which to slide the window at each step (hop_length)

The width of the spectrogram can be derived using the above mentioned hyperparameters as follows:

$$\text{Spectrogram width} = \frac{Total\ number\ of\ samples}{Hop\ length}$$

Steps: The following data pre-processing steps are performed for data from all the four machines (i.e., slider, valve, pump, rail) to convert raw audio files into corresponding mel-spectrogram images.

Step 1. Linear power spectrogram is generated

Step 2. Spectrogram with Mel scale as y-axis is generated

Step 3. Spectrogram with Mel and Db scale is generated.

We evaluated different hyperparameters values for generating the optimized mel-spectrogram. Table 2 describes the parameters evaluated and the values highlighted in bold are selected for data pre-processing.

Table 2. Mel hyper parameters

f_min (Hz)	f_max (Hz)	n_mels	n_fft	hop_length
0	5000	32	256	64
0	5000	64	512	128
0	**5000**	**128**	**1024**	**256**

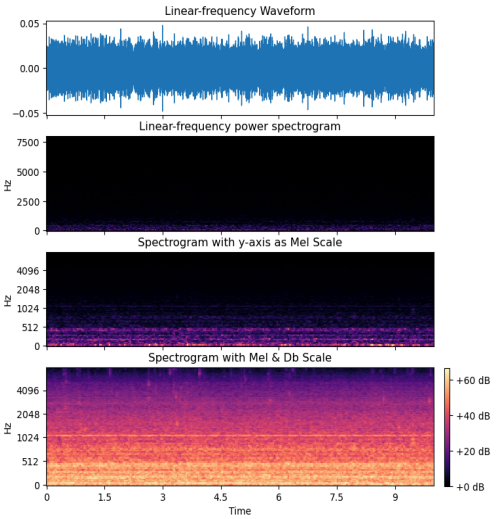

Figure 5. Mel-spectrogram conversion of a normal sample.

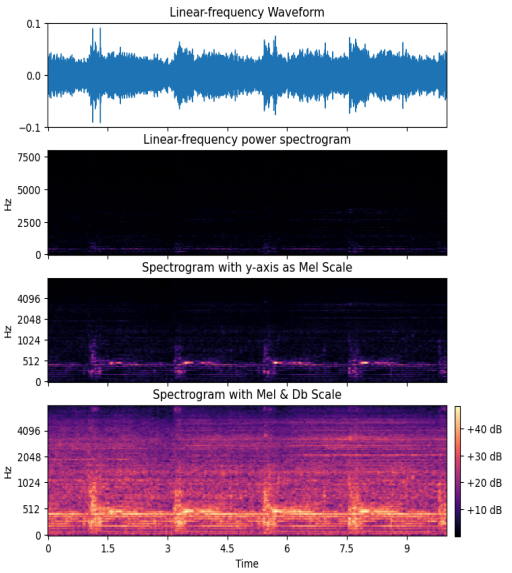

Figure 6. Mel-spectrogram conversion of an anomalous sample.

Figure 5 and 6 shows the conversion of a signal from its raw waveform to an optimized mel-spectrogram for both normal and anomalous machine (pump) conditions.

3.5 Convolutional Neural Network

Convolutional neural networks (CNN) are typically used for analyzing image datasets. Feed forward CNNs are designed to process pixel data and are increasingly being used in image recognition and processing. A basic CNN has an input layer defined by input array shape and size. Next, it has a stack of multiple convolutional layers (CL) and pooling layers. The convolution layer has different input hyper parameters such as number of feature maps, filter size, strides defining the sliding window for filter, and activation function. The stack is followed by a fully connected layer, and finally the output layer (with activation function) for classification.

Convolutional layers (CL) [10] are the major building block of CNN. A convolution is the application of a filter function to an input which results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input image. For convolution, the filter (window) slides across the height and width of the image and the dot product between every element of the filter and the input is calculated at every spatial position. An example of filter operation is shown in Figure 7 below.

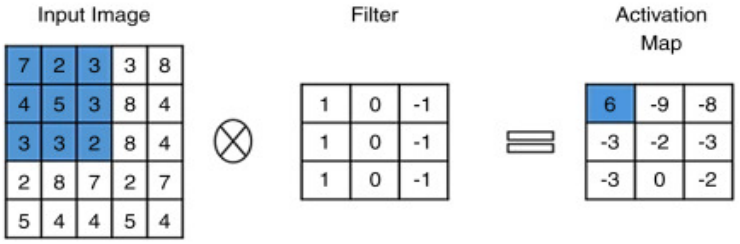

Figure 7. Filter operation in convolution.

Activation Functions are the mathematical functions used in the neural networks to control the output. In this study, ReLU, sigmoid, and linear activation functions were investigated. The mathematical formulas for the activation functions are follows:

ReLU = max (0,x)	Sigmoid (x) = $1/1+e^{-x}$	Softmax (x_k) = $e^{x}_{k} / \Sigma^{n}_{i=1} e^{x}_{i}$

where x represents the input.

Fully Connected layer (FCL) follows the stack of convolutional and pooling layers. These layers connect every neuron in one layer to every neuron in another layer. The flattened matrix of the output from the stack of CL and PL goes through a fully connected layer to classify the images.

Output layer. The final layer is used for the classification and use an activation function. In our experiments, we use linear activation for predicting linear probability distribution of normal data.

4. Experiments

4.1 **Image data conversion**

The spectrogram images were converted to *NumPy* arrays using Python Image Library (PIL) [26] to make them compatible as input to the convolutional neural network (CNN) layers. Following steps were performed in a loop for each input image to create training and test dataset for each machine.

1. Input image was loaded using the "open" function of the "Image" module. It is a lazy operation which identifies the file, but the file remains open and the actual image data is not read from the file until data processing is initiaited.
2. Image object was converted to a numpy array using "scikit-image" which uses standard numpy arrays to allow maximum interoperability with other python libraries in the neural network ecosystem.
3. Image object was sliced from 480 x 640 x 4 (RGBA) to 480 x 640 x 3 (RGB) to limit the level of opacity in the spectrogram images.
4. Image object was converted to float and then normalized by dividing the numpy array with 255. This reduces the computational issues arising from large numeric values when an image is passed through a neural network.
5. Image array was then appended to their class labels, i.e., "normal" and "anomaly".

4.2 **Model configuration**

The detailed configuration of CNN Model of proposed framework is described below in terms of the input and the output of each layer to predict the linear probability score of each image. The first stage contains the first layer as an input layer with the input size of 480 X 640 X 3 followed by a stack of five convolutional layers (Conv2D) with the following number of dimension sizes: 32, 64, 128, 64, 32. Each layer had a filter size (3,3), strides (2), and activation function as 'relu'. The output of the first stage is flattened and then fed into the second stage which consists of a stack of dropout layer (0.5), then a dense fully connected layer with the dimension size of 32, having 'relu' activation function. The last output layer is a fully connected layer with one output feature map and 'linear' activation function.

In the model compilation stage we use the 'Adam' optimizer [22,23] with a learning rate of 0.001, loss function of mean squared errors (MSE), and the evaluation metrics as the mean absolute error (MAE). Finally, we fit the model on a training dataset consisting of only normal samples with a different combination of batch size and epochs for all the four machines i.e., pump, slider, valve, and fan.

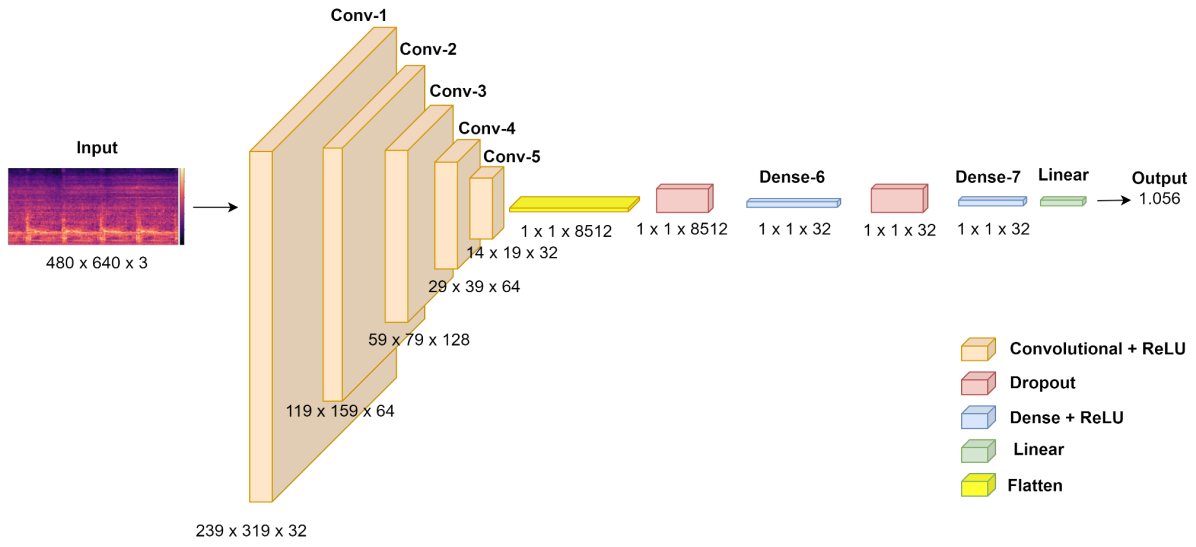

Figure 8. Convolutional Neural Network Architecture

4.3 **Evaluation Metrics**

The overall performance is evaluated using Accuracy [25], Misclassification Rate [25], Precision [25], Recall [25], Specificity or True Negative Rate (TNR) [25], and F1 Score [25]. All the metrics are determined using different attributes of a confusion matrix [24]. In the confusion matrix, true positive (TP) and true negative (TN) are the correctly predicted Normal and Anomaly samples respectively; whereas false positive (FP) and false negative (FN) represent the samples of Normal and Anomaly classes which are incorrectly predicted respectively.

Confusion Matrix [24]: A confusion matrix represents the summary of predicted results (Figure 9).

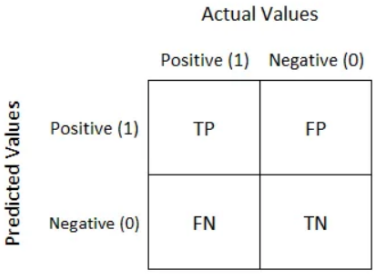

Figure 9. Confusion Matrix [25]

Accuracy is the ratio of the total number of correctly identified samples to the total number of samples [25].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN)}$$

Misclassification Rate is determined as the ratio of the total number of incorrectly predicted samples to the total number of samples [25].

$$Miss.\ Rate = \frac{FP + FN}{TP + TN + FP + FN +}$$

Precision is the ratio of correctly predicted normal samples to the total number of positive samples [25].

$$Precision = \frac{TP}{TP + FP}$$

Recall or True Positive Rate (TPR) is the ratio of correctly classified normal samples to the total number of positive samples [25].

$$TPR = \frac{TP}{TP + FN}$$

F1-score: It is represented as the harmonic mean of Precision and Recall and is a statistical metric to determine the overall correctness of the system [25].

$$F1\ score = \frac{2 + Precision * Recall}{Precision + Recall}$$

4.4 **Experimental Setup**

The experiments are performed on an HP ZBook with 32.0 GB RAM, x64-based processor, and 11th Gen Intel(R) Core(™) i7-11850H processor, and a 64-bit Windows 11 Pro operating system. Anaconda framework (Version 23.1.0) and Python (Version 3.9.10) tools were used to implement the proposed approach. Librosa (Version 0.9.2) library was used to generate mel-spectrograms from audio files for all the machines in MIMII dataset. The convolutional neural networks were implemented using Tensorflow (Version 2.11.0). All the implementation and experimentation was done in the JupyterLab (Version 3.5.3) environment.

Table 3. Number of Samples

	Training Samples	Test Samples	
MachineID	**Normal**	**Normal**	**Anomaly**
pump_id_00	906	100	143
pump_id_02	905	100	111
pump_id_04	602	100	100
pump_id_06	936	100	102
fan_id_00	911	100	407
fan_id_02	916	100	359

fan_id_04	932	100	348
fan_id_06	915	100	361
valve_id_00	891	100	119
valve_id_02	608	100	120
valve_id_04	900	100	120
valve_id_06	892	100	120
slider_id_00	968	100	356
slider_id_02	968	100	267
slider_id_04	434	100	178
slider_id_06	434	100	89

4.5 **Results**

Table 4 shows the numbers for baseline F1 score (baseline_F1_score) and the F1 score, accuracy, misclassification rate, precision, and recall for the CNN based approach. The F1-score evaluation metric is preferred over accuracy to compare the performance of suggested CNN architecture (with mel-spectrogram data augmentation) with the baseline deep neural network architecture trained on raw dataset due to class imbalance in training dataset (i.e., we only have training samples classified as normal). As shown, the CNN based approach gives a higher average F1 score of 0.6921 for pump machines as compared to average F1 score of 0.5185 with a baseline dense neural network. Similarly, for the other three machines fan, valve, and slider the CNN based approach has a higher F1 score of 0.8802, 0.7040, 0.7850 as compared to 0.5858, 0.4923, 0.5499 for the baseline respectively. In summary, we find that there is an average improvement in F1 score by 0.1736, 0.2944, 0.2117, 0.2351 for pump, fan, valve, and slider respectively.

Table 4. Evaluation Metric results

Machine ID	baseline_F1_Score	F1_score	Accuracy	Miss Rate	Precision	Recall
pump_id_00	0.5159	0.7409	0.4074	0.5926	0.0629	0.4737
pump_id_02	0.5125	0.6894	0.5261	0.4739	1.0000	0.5261
pump_id_04	0.456	0.6666	0.5650	0.4350	0.2700	0.6585
pump_id_06	0.5899	0.6711	0.5050	0.4950	1.0000	0.5050
fan_id_00	0.5925	0.8904	0.8028	0.1972	0.9975	0.8040
fan_id_02	0.5056	0.8778	0.7821	0.2179	1.0000	0.7821
fan_id_04	0.7324	0.8744	0.7768	0.2232	1.0000	0.7768
fan_id_06	0.5128	0.8783	0.7831	0.2169	1.0000	0.7831
valve_id_00	0.4999	0.7041	0.5434	0.4566	1.0000	0.5434
valve_id_02	0.4712	0.7059	0.5455	0.4545	1.0000	0.5455
valve_id_04	0.5059	0.7059	0.5455	0.4545	1.0000	0.5455
valve_id_06	0.4923	0.7003	0.5409	0.4591	0.9833	0.5438
slider_id_00	0.4891	0.8768	0.7807	0.2193	1.0000	0.7807
slider_id_02	0.5675	0.8423	0.7275	0.2725	1.0000	0.7275
slider_id_04	0.6754	0.7807	0.6403	0.3597	1.0000	0.6403
slider_id_06	0.4678	0.6403	0.4709	0.5291	1.0000	0.4709

5. Conclusion and Future work

Conclusion: In this study we explore the idea of converting audio samples from different industrial machines such as pump, slider, fan, and valve to mel-spectrograms images using Short Time Fourier Transform (STFT). The converted audio samples were then used to train convolutional neural network (CNN) based models using "normal" mel-spectrogram images. The model performance was evaluated on a test data set having unseen normal and anomaly samples. The performance of CNN architecture was compared with the baseline deep neural network trained using raw audio samples. The results show that CNN based models perform better than models trained on the raw dataset (i.e., sound samples without spectrogram conversion).

Future Work: In the future we plan to extend the work in the following manner: 1. Evaluate and optimize the CNN architecture with additional normal instances to the training dataset. 2. Explore train accuracy optimization using different hyper parameters such as batch size, epochs, and learning rate. 3. Compare the performance of suggested CNN architecture with other related work discussed in Section 2 above. 4. Evaluate performance of suggested CNN approach with pre-trained networks (e.g., VGG-16).

References

1. Yuma Koizumi, Shoichiro Saito, Noboru Harada, Hisashi Uematsu, and Keisuke Imoto, "ToyADMOS: A Dataset of Miniature-Machine Operating Sounds for Anomalous Sound Detection," in Proc of Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2019. [pdf]
2. Harsh Purohit, Ryo Tanabe, Kenji Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi, "MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection," in Proc. 4th Workshop on Detection can you and Classification of Acoustic Scenes and Events (DCASE), 2019. [pdf]
3. Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido, Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, and Noboru Harada, "Description and Discussion on DCASE2020 Challenge Task2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring," in Proc. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), 2020. [pdf]
4. Delattre, Pierre. "The physiological interpretation of sound spectrograms." Pmla 66.5 (1951): 864-875.
5. M.Abdel-Sayed, D.Duclos, G.Faÿ, J.Lacaille, and M.Mougeot, "Anomaly detection on spectrograms using data driven and fixed dictionary representations", 17 Jan 2023
6. Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, Georg Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery", 17 Mar 2017
7. Dan Hendrycks, Kevin Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks", 7 Oct 2016.
8. Samet Akcay, Amir Atapour-Abarghouei, Toby P. Breckon, "GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training", 17 May 2018.
9. A. S. Khan, Z. Ahmad, J. Abdullah and F. Ahmad, "A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network," in IEEE Access, vol. 9, pp. 87079-87093, 2021, doi: 10.1109/ACCESS.2021.3088149.
10. Sakib Mostafa, Fang-Xiang Wu, Chapter 3 - Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images, Editor(s): Ayman S. El-Baz, Jasjit S. Suri, Neural Engineering Techniques for Autism Spectrum Disorder, Academic Press, 2021, Pages 23-38, ISBN 9780128228227, https://doi.org/10.1016/B978-0-12-822822-7.00003-X
11. Mobley, R. Keith. An introduction to predictive maintenance. Elsevier, 2002.
12. Fox, Mark S. "Industrial applications of artificial intelligence." Robotics 2.4 (1986): 301-311.
13. Siemens and Google Cloud to cooperate on AI-based solutions in manufacturing, ://press.siemens.com/global/en/pressrelease/siemens-and-google-cloud-cooperate-ai-based-solutions-manufacturing
14. From condition monitoring to predictive maintenance: turning date to real customer value, https://en.industryarena.com/boschrexroth/news/from-condition-monitoring-to-predictive-maintenance-turning-date-to-real-customer-value--10294.html
15. Sejdić, Ervin, Igor Djurović, and Jin Jiang. "Time–frequency feature representation using energy concentration: An overview of recent advances." Digital signal processing 19.1 (2009): 153-183.
16. Xu, Min, et al. "HMM-based audio keyword generation." Advances in Multimedia Information Processing-PCM 2004: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30-December 3, 2004. Proceedings, Part III 5. Springer Berlin Heidelberg, 2005.
17. S. S. Stevens and J. Volkmann, Harvard University, Cambridge, Massachusetts, E. B. Newman, Swarthmore College, Swarthmore, Pennsylvania, "A Scale for the Measurement of the Psychological Magnitude Pitch", The Journal of the Acoustical Society of America 8, 185-190 (1937) https://doi.org/10.1121/1.1915893
18. E. Sejdic, I. Djurovic and L. Stankovic, "Quantitative Performance Analysis of Scalogram as Instantaneous Frequency Estimator," in IEEE Transactions on Signal Processing, vol. 56, no. 8, pp. 3837-3845, Aug. 2008, doi: 10.1109/TSP.2008.924856.
19. Wikipedia contributors. "Audio signal." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 27 Oct. 2022. Web. 16 Apr. 2023.
20. Heideman, Michael T., Don H. Johnson, and C. Sidney Burrus. "Gauss and the history of the fast Fourier transform." Archive for history of exact sciences (1985): 265-277.

21. Analog vs. Digital Signals: Uses, Advantages and Disadvantages | Article | MPS (monolithicpower.com)
22. A. Agnes Lydia and , F. Sagayaraj Francis, Adagrad - An Optimizer for Stochastic Gradient Descent, Department of Computer Science and Engineering, Pondicherry Engineering College, May 2019
23. Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: neural networks for machine learning, 4(2):26–31, 2012
24. Stephen V. Stehman, Selecting and interpreting measures of thematic classification accuracy, Remote Sensing of Environment, Volume 62, Issue 1, 1997, Pages 77-89, ISSN 0034-4257, https://doi.org/10.1016/S0034-4257(97)00083-7.(https://www.sciencedirect.com/science/article/pii/S0034425797000837)
25. Powers, David M W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" (PDF). Journal of Machine Learning Technologies.
26. Python Imaging Library (PIL), https://web.archive.org/web/20201121102218/http://www.pythonware.com/products/pil/