AML challege: Anomaly Detection

Giuseppe Antonio Orlando
Politecnico di Torino
EURECOM
orlando@eurecom.fr

Simone Varriale
Politecnico di Torino
EURECOM
varriale@eurecom.fr

Gabriele Sanmartino
Politecnico di Torino
EURECOM
sanmarti@eurecom.fr

I. Introduction

The task for this challenge is to detect unknown anomalous sounds under the condition that only normal sound samples have been provided as training data. Anomalous sound detection in industrial environments is essential for preventing malfunctions and ensuring operational efficiency. This task focuses on detecting such anomalies using data from the ToyADMOS and MIMII datasets, specifically targeting slide rail machines. Different models will be proposed and their capability of generalizing to new data will be studied.

II. Dataset

The data used for this task is part of ToyADMOS and the MIMII Dataset consisting of the normal/anomalous operating sounds of six types of toy/real machines. Each recording is a single-channel (approximately) 10-sec length audio that depicts a target machine's operating sound. Only one type of machine will be considered: Slide rail (MIMII Dataset). The dataset provided can be divided into 2 big subsets:

- Development dataset of 2370 (i) normal samples for training and 1101 (ii) normal and anomalous samples for the test. All the samples present in this portion of dataset come from 3 different machines, with id 00, 02, 04.
- Evaluation dataset of 2370 (iii) normal samples for training and 834 (iv) not annotated samples for the test. All the samples present in this portion of dataset come from 3 different machines, with id 01, 03, 05. It is worth noticing how the machines from which samples have been collected for this portion of dataset are not the same machines selected for the development dataset.

The development portion will be used during the train phase, due to the availability of labeled test data, useful to understand how the picked models will perform in the anomaly detection task. The evaluation dataset will instead be used in Section VII.

III. Data pre processing

Various data preprocessing techniques are analysed in this section in order to extract the maximum amount of information from the provided audio samples. Techniques III-A, III-B, III-C will be used for an autoencoder model; technique III-D will instead be used for a Gaussian Mixture Model. Both these models will be presented in Section IV.

For the rest of this section, examples will be shown for both normal and anomalous samples. Those examples are taken from the test portion of the development dataset, although they will not be included in the training data of the proposed models. In Figure 1 the comparison between the wave of one normal sample and one anomalus sample is shown.

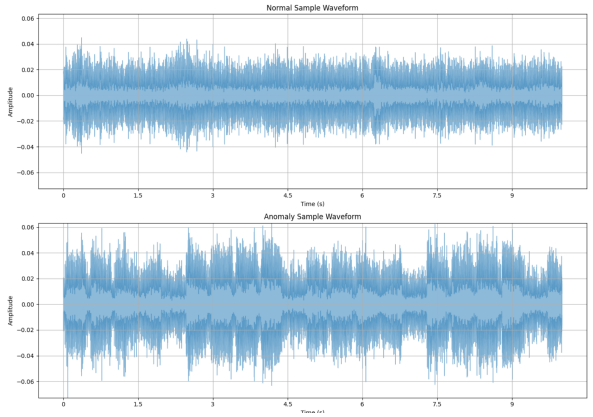


Fig. 1: Wave plot of two samples

A. Fourier Transform

To study the different frequencies present in the snippets, Fourier transform was initially applied. This transformation decomposes a signal into its constituent frequencies, providing insight into the spectral characteristics of the audio data. It is then possible to analyze the distribution of energy across different frequencies. Figure 2 shows some examples of Fourier transformed audio samples.

B. Log Mel Spectogram

To bind together informations about time and frequencies the Log Mel spectogram has been employed.

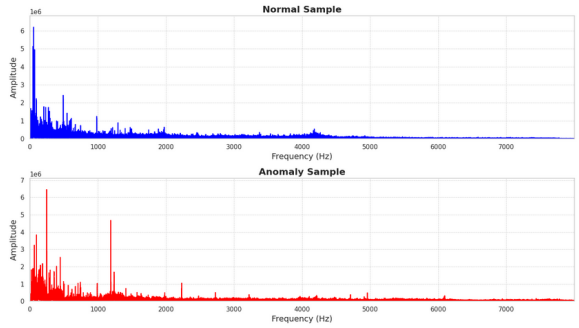


Fig. 2: Fourier transform of two samples

It is derived from the Mel spectrogram, which splits the audio spectrum into frequency bands called Mel bins. The log-mel spectrogram then takes the logarithm of the power values which makes it more effective in the extraction of meaningful features. An example for a normal and an anomalous sample is reported in Figure 3.

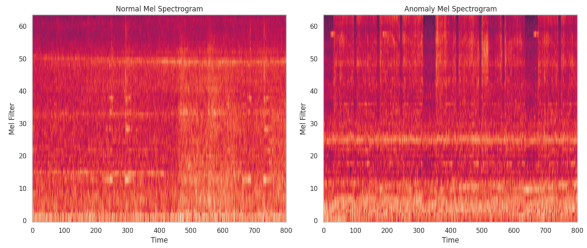


Fig. 3: Log mel spectogram of two samples

C. Normalization

Normalization of mel spectrograms is a key preprocessing step in audio analysis. It adjusts the data to have a mean of zero and a standard deviation of one. This process enhances model training by improving convergence speed, stabilizing the learning process, and ensuring each feature contributes equally. By normalizing, the model handles different scales of input data better, leading to more efficient and effective learning.

D. Mel-frequency cepstral coefficients

Mel-Frequency Cepstral Coefficients is a technique often used for timbral description and timbral comparison. It compresses the spectrum into a smaller number of coefficients that, when taken together, describe the general contour of the spectrum [4]. The MFCC values are derived by first obtaining a mel-frequency spectrum, just as in MelBands. The coefficients are then computed using that mel-frequency spectrum as input to the discrete cosine transform. Each MFCC value represents how similar the mel-frequency spectrum is to one of these cosine shapes

IV. Model Selection

Two models will be presented in this section: an autoencoder model and a Gaussian Mixture model. The autoencoder learns, in the training phase, to reproduce the training samples. It can then be employed, looking at the reconstruction error, to understand if a given input sample is an anomaly. This is based on the assumption that anomalies, that are not present at training time and are distant from the normal samples, won't be reconstructed well by the model. Both the encoder and the decoder are constructed starting from repeated blocks, made of a dense linear layer, batch-norm 1D, and ReLU activation function. The size of the bottleneck will be the main object of study. Alongside this deep learning model, a Gaussian Mixture Model (GMM) is also used due to its promising performance in anomaly detection [5]. A GMM is able to identify anomalies by modeling normal data distributions and pointing out samples with low probability. Performances of the two models will be compared using the AUC score obtained on part (ii) of the development data, being (ii) the only portion of the dataset where anomaly samples are present.

A. Autoencoder

This subsection will focus on the training of an autoencoder model. As proposed in [1], [2], [3], the strategy to train the autoencoder will be the following:

First, convert the input audio sequence $X = \{X_t\}_{t=1}^{T}$, where $X_t \in \mathbb{R}^F$, into a log-mel-spectrogram. Then, extract the acoustic feature at each time frame t by concatenating P consecutive frames of the log-mel-spectrogram. Form the feature vector $\psi_t = (X_t, \ldots, X_{t+P-1}) \in \mathbb{R}^D$, where $D = P \times F$.

The autoencoder is trained using only normal (non-anomalous) sounds from all machines ids to minimize the reconstruction error, defined as the ℓ_2 norm between the input feature vector ψ_t and its reconstruction $r_\theta(\psi_t)$. In order to create a general model, able to work with any number of different machines, the machine id has not been leaked to it. It's noteworthy noticing that only non-overlapping windows were used in training, meaning each window starts after the first one terminates to speed up the training process.

The anomaly score $A_\theta(X)$ is instead computed as the average reconstruction error over all time frames:

$$A_\theta(X) = \frac{1}{DT} \sum_{t=1}^{T} \|\psi_t - r_\theta(\psi_t)\|_2^2$$

This strategy ensures that the autoencoder is optimized to accurately reconstruct normal sounds. Anomalous samples should instead be badly reconstructed by the model, since it hasn't learned to reconstruct them.

The first analysis conducted was aimed at identifying the best size of the autoencoder bottleneck. Table I shows the obtained result.

Bottleneck size	*AUC*
4	0.8574
8	0.8781
16	**0.8832**
32	0.8803

TABLE I: Performances of the baseline model

The number of frames per window P has also been studied to assess whether varying the amount of information in the context window would be beneficial. This analysis has been conducted exclusively for the best bottleneck size found above and its results are reported in Table II.

P	*AUC*
1	0.8847
3	**0.8966**
5	0.8942
10	0.8832
20	0.8575
30	0.8409

TABLE II: Performances based on P

The last study conducted was aimed at understanding if some overlap in the windows used for training could help model performances. The overlap has only been studied for the best parameters obtained above. The results are shown in Table III.

Hop size	*AUC*
1	**0.8997**
2	0.8962
3	0.8966

TABLE III: Performances based on hop size

B. GMM

This subsection will focus on the training of a Gaussian Mixture Model. This type of model can be used in anomaly detection tasks thanks to its ability in estimating samples distributions. If a data point does not lie in the fitted distribution, then it can be considered an anomaly. The log likelihood is used to score samples: if a sample has a low log likelihood, it might be outside of our normal data distribution, hence it will be flagged as an anomaly.

To reduce the dimentionality of data, in order to fit a GMM model, mel-frequency cepstral coefficients have been obtained for each audio samples. Figure 4 shows different settings for the model. Here, it is possible to see that the best *AUC* scores are obtained with the value of n_{mel} set to 64, while the performances are stable when the *sample rate* is scaled by a factor ranging from $\frac{1}{32}$ to $\frac{1}{8}$. For the next steps, $\frac{1}{16}$ will be used as scaling factor, being in the middle of the range.

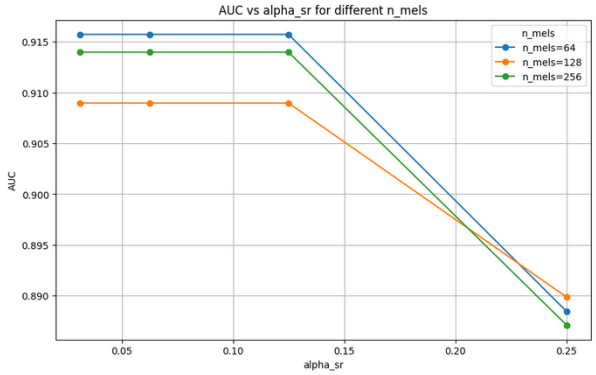


Fig. 4: AUC scores with GMM

After identifying the optimal parameters for feature extraction, a second validation was conducted to determine the best number of components for the GMM. As illustrated in Figure 5, there is a slight improvement in the AUC when the number of components is set to 5. Eventually, the PCA method is applied to the training input to reduce its dimensionality, and after validation on the number of PCA parameters, the best performances are obtained with 21 components, as shown in Figure 6. Table IV shows the performance obtained with the best number of components for GMM and PCA.

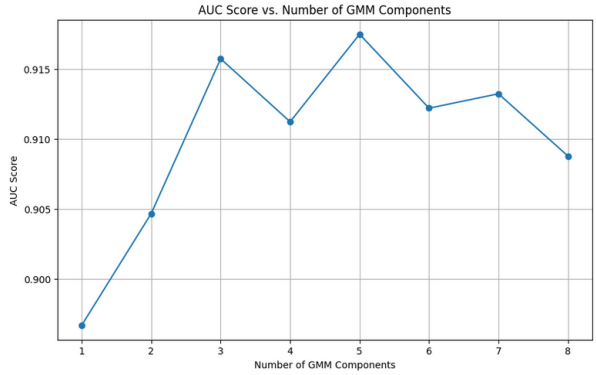


Fig. 5: AUC changing the number of components

GMM Components	**PCA Components**	**AUC**
5	21	0.9234

TABLE IV: Best GMM score

V. Model Performance

Based on the analysis conducted in Section IV, the final autoencoder configuration is:

- 16 as size for the bottleneck

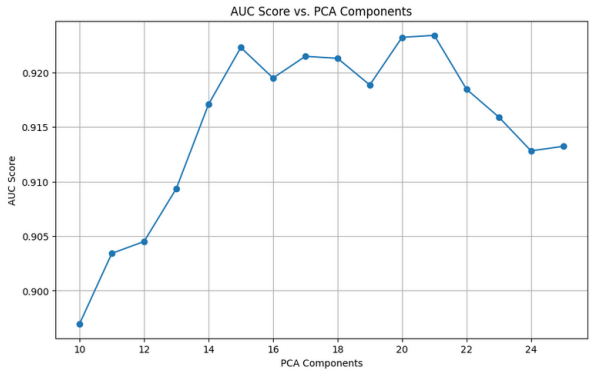


Fig. 6: AUC changing the number of PCA components

- 3 number of frames per window
- 1 hop size for the overlap of the window

While, the final GMM configuration is:

- 5 components
- 21 features for the data
- 64 number of mels
- $\frac{1}{16}$ of target sample rate

The best performances obtained for each model are summarized in table V.

Models	AUC
Autoencoder	0.8997
GMM	0.9234

TABLE V: Final scores

The slightly worse performance of the autoencoder will be explained in Section VI. The histograms in Figure 7 and Figure 8 depict the distributions of the anomaly scores on the test set. The plot shows that there is an overlapping of the two distributions, mostly due to the dataset containing samples taken from different machines.

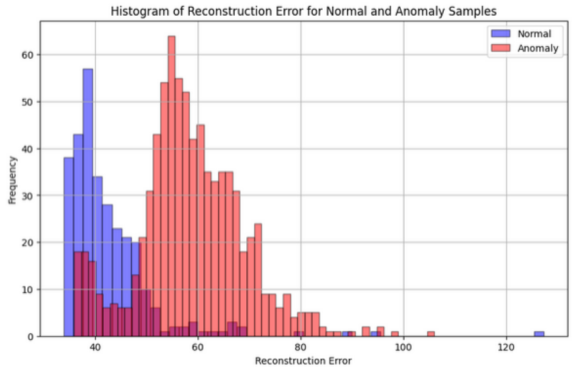


Fig. 7: Distribution of reconstruction error

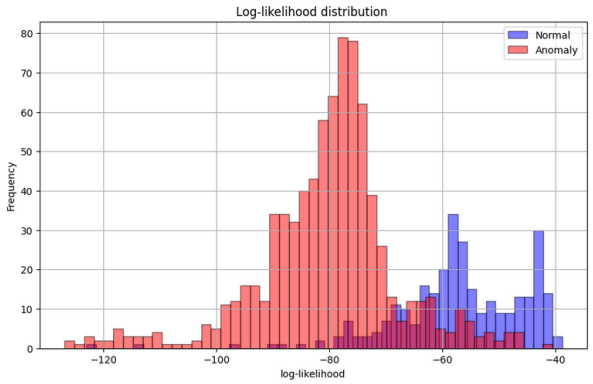


Fig. 8: Distribution of log-likelihood score

VI. Multiple Autoencoders

This section provides an in-depth analysis of the results obtained from the autoencoder model. The model was trained on samples from three different machines. The following analysis will cover the performance of the autoencoders on each machine and the overall effectiveness of using multiple autoencoders for this type of data.

A. Training and Validation

The autoencoder model was trained separately on data from each of the three machines. Performances were evaluated on the portion of test dataset coming from the selected machine.

B. Results

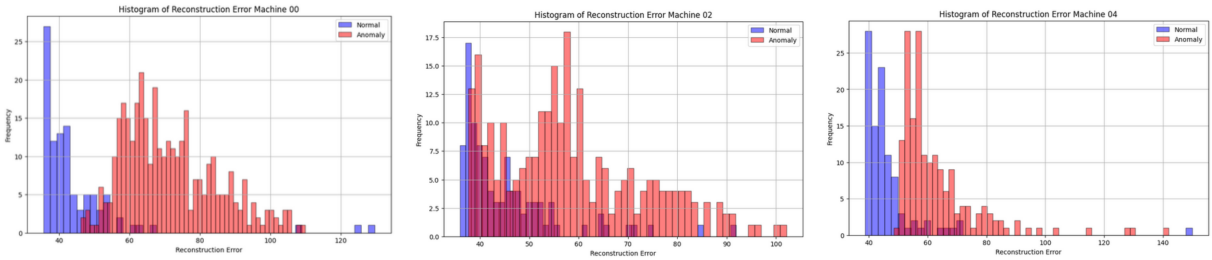
1) Machine 00: the autoencoder achieved an AUC score on the test of 0.9543.

2) Machine 02: the autoencoder achieved an AUC score on the test of 0.7963.

3) Machine 04: the autoencoder achieved an AUC score on the test of 0.9283.

C. Discussion

The varying AUC scores across the different machines highlight the differences in data characteristics and complexities from each machine. Machine 00 showed the best performance, while Machine 02 was the most challenging for the autoencoder. These results suggest that while a single autoencoder can be effective, the variability in performance indicates potential benefits in customizing models or employing different techniques for different machines. The histograms in Figure 9 reports the distribution of reconstruction errors for both normal and anomalus samples for each machine. In the figure it is possible to notice how, the overlap of the distributions for machine 02 is higher than the ones seen for the other machines. This explains why



((a)) Machine 0 ((b)) Machine 2 ((c)) Machine 4

Fig. 9: Reconstruction error distribution

the performances of the best model studied in Section V cannot be improved.

VII. Machine Generalization

In this last section the behaviour of the autoencoder model, in presence of data from more machines will be studied. The model will be provided with data from both portions (i) and (iii) during training. The performances obtained on set (ii) will then be discussed. All the hyper-parameters found in Section IV will be used for simplicity. The AUC results obtained on the test set are the following:

- All machines: 0.8712
- Machine 00: 0.9525
- Machine 02: 0.7817
- Machine 04: 0.9423

From the reported results it is possible to notice how adding new data from different machines did not excessively worsen the performances of the model. As expected from previous results, the model did not perform as well on Machine 02. In contrast, it performed excellently on Machines 00 and 04. This suggests the model is robust but may struggle with specific Machine 02 related anomalies.

VIII. Conclusion

The task of this challenge was detecting unknown anomalous sounds using only normal sound samples for training. Two methodologies were employed: an autoencoder model and a Gaussian Mixture Model (GMM). The latter outperformed the autoencoder, achieving an AUC score of 0.9234 compared to the autoencoder's 0.8997. The autoencoder's performance varied significantly across different machines, with Machine 02 presenting the greatest challenge. Training separate autoencoder models for each machine yielded higher AUC scores, indicating the potential benefits of a more customized approach. Incorporating data from multiple machines into the training process maintained robust performance for some machines but still faced challenges with Machine 02.

Future work could enhance model robustness through techniques such as domain adaptation. Samples from difficult domains could be mapped to more "easy to classify" domains. More sophisticated preprocessing and feature selection techniques could also be applied to reduce the domain gap between the different machines.

References

[1] Kota Dohi, Keisuke Imoto, Noboru Harada, Daisuke Niizumi, Yuma Koizumi, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo, Masaaki Yamamoto, and Yohei Kawaguchi. Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques. In *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, pages 1–5, Nancy, France, November 2022.

[2] Kota Dohi, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo, Masaaki Yamamoto, Yuki Nikaido, and Yohei Kawaguchi. MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task. In *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, pages 1–5, Nancy, France, November 2022.

[3] Noboru Harada, Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Masahiro Yasuda, and Shoichiro Saito. ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions. In *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, pages 1–5, Barcelona, Spain, November 2021.

[4] K.V. Krishna Kishore and P. Krishna Satish. Emotion recognition in speech using mfcc and wavelet features. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 842–847, 2013.

[5] Kazuki Morita, Tomohiko Yano, and Khai Q Tran. Anomalous sound detection by using local outlier factor and gaussian mixture model. In *Proceedings of the 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), Tokyo, Japan*, pages 2–4, 2020.