Anomaly Detection in Audio Signals using Autoencoders and Variational Autoencoders

AML Challenge 2
Group n° 50

Victor MAYAUD[1], Elliot BOUCHY[1]

[1] Data Science, EURECOM, France

Abstract

This study focuses on classifying anomalous sounds in factory environments, particularly within slide rail machinery. Machinery failures and breakdowns lead to substantial costs for companies. To mitigate these expenses, companies are investing in sensors and artificial intelligence to enhance anomaly detection and reduce maintenance costs. One effective approach involves using autoencoders to analyze spectrograms of machine sounds. However, traditional autoencoders may have limited accuracy. We propose using variational autoencoders (VAEs), which have demonstrated superior performance in identifying anomalous sounds.

Index Terms: Anomaly Detection, Unsupervised Learning, Audio Signal Processing, Machine Learning, Autoencoders, Variational Autoencoders (VAEs), Feature Extraction, Reconstruction Error

1 Introduction

This report investigates the effectiveness of autoencoders (AEs) and variational autoencoders (VAEs) in identifying anomalous sounds within machine operating data. We focus on the DCASE challenge, specifically targeting anomalies in slide rail machine sounds. Our goal is to detect deviations from normal operation without prior knowledge of specific anomaly types.

We begin by providing a foundational understanding of anomaly detection, AEs, VAEs, and their applications in audio anomaly detection. We also explore existing methods, discussing their strengths and limitations.

Next, we detail our proposed methodology, including the dataset used, the architectures of the convolutional autoencoder (CAE) and convolutional VAE (C-VAE) models, the training and evaluation procedures, and the anomaly detection strategies employed. The performance of the models is then presented and compared using various metrics, including AUC, accuracy, precision, recall, and F1-score, along with qualitative analysis of reconstruction errors and latent representations.

Finally, we discuss the experimental results, analyzing the effectiveness of each model, comparing their performance, highlighting limitations, and suggesting potential avenues for future research. The report concludes by summarizing key findings and emphasizing the contributions and potential impact of our proposed models on the field of audio anomaly detection.

2 Dataset Analysis

2.1 Description

The dataset used in this project is derived from the MIMII dataset [1], focusing solely on slide rail machines. Each recording is a 10-second, single-channel audio file capturing operating sounds and ambient factory noise. The dataset includes normal and anomalous operating conditions, with anomalous sounds simulated by deliberately damaging the machines. All signals are downsampled to 16 kHz for standardization and efficient processing.

2.2 Dataset

The dataset is split into training and testing sets. The training set contains 2,370 normal sound records, while the test set has 1,101 records (300 normal, 801 anomalous). The key challenge is to detect unknown anomalies when trained only on normal sounds. This unsupervised learning scenario is realistic, as real-world factory anomalies are rare and diverse, making it impractical to collect examples of every possible anomaly. Both datasets include sounds from three slide rails (IDs 00, 02, and 04).

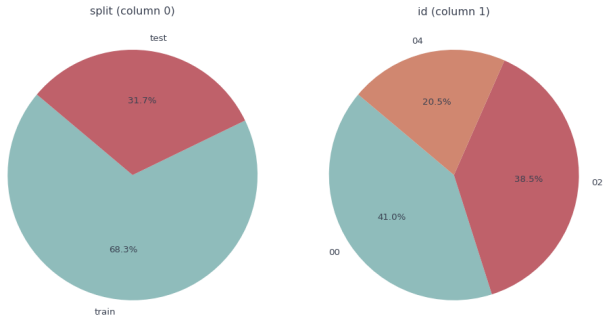

Figure 1. Composition of the dataset

2.3 Sound Analysis

The sound data provided in this project is in WAV format, which cannot be directly used to train a machine learning model. To effectively analyze and utilize this data, we need to transform the raw audio signals into a format suitable for modeling. One of the initial steps in this process is to visualize the audio signals to understand their characteristics better.

To begin with, we can plot the waveforms of both normal and anomalous signals to observe their differences. By visualizing these waveforms, we can gain insights into the nature of the sound anomalies. For instance, Figure 2 demonstrates a comparison between a normal signal and an anomalous signal. In this figure, the normal signal is represented in blue, while the anomalous signal is depicted in red.

Upon examining the waveforms, it is evident that the anomalous signal exhibits irregularities compared to the normal signal. These irregularities can manifest as sudden spikes or variations in amplitude, indicating the presence of anomalies in the machinery. Specifically, the malfunction in one of the slide rails

results in an amplification of the signal at various points, which can be clearly observed in the red waveform. These anomalies might be caused by mechanical issues such as friction, misalignment, or other forms of wear and tear that disrupt the normal operation of the slide rail.

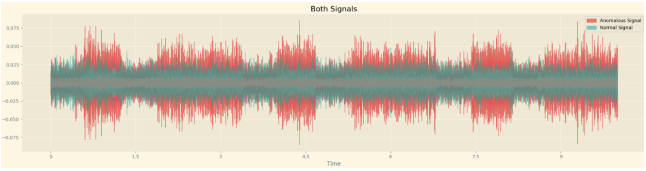

Figure 2. Representation of the wave forms

Furthermore, to enhance our analysis, we can convert these waveforms into spectrograms. A spectrogram provides a visual representation of the signal's frequency spectrum over time, offering a more detailed view of the sound characteristics. This transformation helps in identifying specific frequency patterns that may be associated with normal or anomalous conditions. By analyzing the spectrograms, we can develop a more robust feature set for training our anomaly detection model.

The figure 3 and the figure 4 show spectrograms of audio signals, illustrating the frequency content over time. In these spectrograms, the horizontal axis represents time, the vertical axis represents frequency, and color intensity indicates the amplitude of frequency components.

- **Normal Signal Spectrogram:** The first spectrogram depicts a normal signal with consistent and regular frequency patterns, indicated by the uniform distribution of colors. This reflects stable and predictable operating conditions typical of a well-functioning slide rail machine.
- **Anomalous Signal Spectrogram:** The second spectrogram shows an anomalous signal, characterized by irregular frequency patterns and disruptions in color intensity. These irregularities suggest mechanical issues or malfunctions in the slide rail machine.

Comparing these spectrograms highlights the differences between normal and anomalous conditions, demonstrating the effectiveness of spectrograms in identifying sound anomalies. These spectrograms suggest the use of autoencoders [2] to leverage visual computation for classifying normal and anomalous sounds in an unsupervised manner.

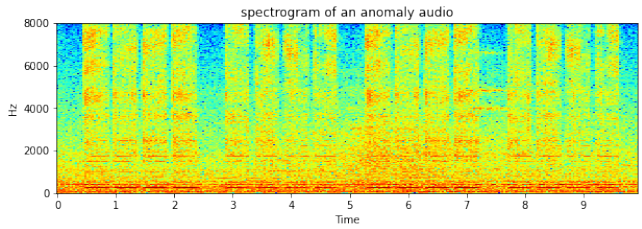

Figure 3. Spectrogram of an anomaly audio

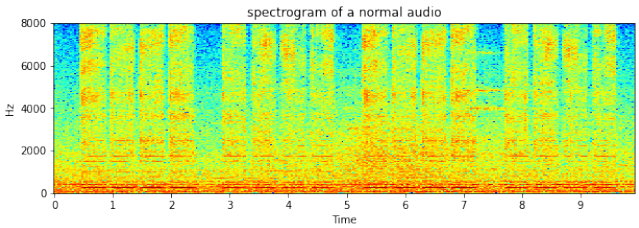

Figure 4. Spectrogram of a normal audio

3 Proposed Approach

3.1 Dataset

The dataset consists of slide rail machine sounds from the DCASE challenge. It includes a development dataset (with normal and anomalous sounds), an additional training dataset (only normal sounds), and an evaluation dataset (unlabeled sounds) [3].

3.2 Model Architectures

In the field of audio anomaly detection, two primary deep learning models are often employed: the Autoencoder (AE) and the Convolutional Variational Autoencoder (C-VAE).

An autoencoder, as introduced by Hinton and Salakhutdinov [4] is a neural network that learns to compress and then reconstruct its input data. It has an encoder that reduces the input to a smaller representation and a decoder that tries to rebuild the original input from this reduced form. The difference between the original and reconstructed data, known as the reconstruction error, is key. When dealing with audio, an AE is trained on normal sounds. If it encounters a new sound and the reconstruction error is high, this indicates the sound is different from the learned normal sounds, potentially signaling an anomaly.

The C-VAE is similar to the AE but differs in that intead of learning a single representation for each input, the C-VAE's encoder learns a probability distribution [6]. This distribution is usually defined by its mean and variance. The decoder then samples from this distribution to create the reconstruction. The C-VAE is particularly good at handling data with spatial structure, like audio spectrograms, because it uses convolutional layers in both the encoder and decoder. It can not only spot anomalies but also create new audio samples. Anomalies are identified by checking how likely an input signal is under the learned distribution. If a signal is unlikely, it's considered anomalous.

- **Implementation:** Both the implementations of AE and VAE utilize the libraries PyTorch. PyTorch is a comprehensive Python library that facilitates the creation of complex machine learning models and provides a user-friendly interface for neural network construction and training. To expedite the training phase, 100 epochs were set and the VAE model uses a batch size of 512 while the AE model uses 128. During model compilation, the Adam optimizer was selected due to its efficiency in handling sparse gradients and its adaptiveness in updating network weights. The Mean Squared Error (MSE) was used as the reconstruction loss function.

3.3 VAE Implementation for Anomalous Sound Detection

This section focuses on the VAE architecture utilized for anomalous sound detection, detailing the design choices and their justifications.

- **Fully Connected Layers:** The encoder consists of two fully connected layers with batch normalization and ReLU activation. These layers progressively reduce the dimensionality of the input data and extract meaningful features. The decoder mirrors this structure to reconstruct the input data from the latent space representation.
- **Batch Normalization:** Normalizes the activations of the previous layer at each batch, maintaining mean output close to 0 and the output standard deviation close to 1. This stabilizes the learning process and dramatically reduces the number of training epochs required to train deep networks.
- **ReLU Activation:** Applied after each batch normalization, introduces non-linearity into the model, enabling it to learn complex patterns.
- **Decoder:** The decoder consists of two fully connected layers that expand the latent representation back to the original input dimensions, with batch normalization and ReLU activation applied after each layer.

Table 1
Detailed VAE Model Architecture for Anomalous Sound Detection

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 640)	0
Linear-1	[-1, 400]	256,400
BatchNorm1d-1	[-1, 400]	800
Linear-2	[-1, 400]	160,400
BatchNorm1d-2	[-1, 400]	800
Linear-3	[-1, 20]	8,020
Linear-4	[-1, 20]	8,020
Linear-5	[-1, 400]	8,400
BatchNorm1d-3	[-1, 400]	800
Linear-6	[-1, 400]	160,400
BatchNorm1d-4	[-1, 400]	800
Linear-7	[-1, 640]	256,640

3.3.1 Hyperparameters: The selection of hyperparameters was a critical aspect of model tuning, aimed at optimizing both the model's performance and its computational efficiency. Each hyperparameter setting was carefully chosen based on its impact on the model's ability to generalize and learn from the training data effectively. Here we provide a detailed look at these choices:

- **Weight Decay:** The weight decay parameter was set to 0.0, indicating that no regularization was applied to the weights. This choice was made to avoid underfitting, as the primary goal was to ensure the model learns the underlying data distribution without penalizing large weights.
- **Filter Sizes and Numbers:** The convolutional layers utilized varying filter sizes and numbers, strategically selected to extract and learn rich feature representations at various levels of abstraction. For instance, initial layers used smaller filters to capture fine details and edges, while deeper layers used larger numbers of filters to aggregate these features into more complex patterns. This approach helps in capturing sufficient contextual information without losing detail by being either too broad or too narrow.
- **Number of Layers:** The VAE architecture includes several fully connected layers and batch normalization layers. The model's depth, characterized by multiple linear and batch normalization layers, was designed to be deep enough to capture complex features but balanced to avoid excessive computational burden. This ensures that the model can process higher-level features without becoming overly complex, which could lead to overfitting.
- **Trainable parameters:** A total of 861,480 trainable parameters were included in the model. These parameters are those that the model learns from the training data and adjusts through backpropagation. The high number of trainable parameters indicates the model's capacity to learn detailed and nuanced features from the data.

3.3.2 Optimization and Loss Function: Adam optimizer was chosen for its adaptive learning rate capabilities [7], which helps converge to the minimum more efficiently. The specific parameters for the Adam optimizer were set to a learning rate of 0.001 with beta values of 0.9 and 0.999, ensuring a good balance between the speed of convergence and the stability of the training process.

The VAE uses a combination of reconstruction loss and KL divergence as its loss function. The reconstruction loss is typically Mean Squared Error (MSE), which measures how well the output matches the input. This loss function is particularly suitable for anomaly detection tasks, as it effectively quantifies the difference between the reconstructed output and the original input data. The KL divergence term regularizes the latent space by ensuring that the learned distribution is close to the prior distribution, promoting a well-structured latent space.

The reconstruction loss (MSE) measures how far off the reconstructed output is from the original input, making it an effective metric for evaluating the model's performance in reconstructing normal data points. The KL divergence adds a regularization term that penalizes the model if the learned distribution deviates significantly from the prior distribution, encouraging the model to learn meaningful representations in the latent space.

The combined optimization and loss strategy ensures that the model not only learns to reconstruct the input data accurately but also maintains a structured and meaningful latent space, crucial for distinguishing normal data points from anomalies.

3.4 AE Implementation for Anomalous Sound Detection

This segment delves into the specifics of the Autoencoder (AE) model tailored for detecting anomalies within audio signals. We outline the architectural decisions and rationale behind them.

- **Layer Composition:** The AE's encoder comprises three densely connected layers, each followed by ReLU activation functions. This setup enables the model to gradually condense the input data into a compact representation. The decoder, mirroring the encoder's structure, aims to restore the original input from this compressed form.
- **Activation Function:** ReLU activation is employed post each layer, injecting non-linear properties into the model, thereby facilitating the learning of intricate patterns
- **Decoder:** The decoder is constructed from three densely connected layers, with ReLU activation applied following each layer, barring the final one, to revert the latent representation back to the input's original dimensions.

3.4.1 Hyperparameters: Choosing the right hyperparameters was pivotal for fine-tuning the model, aiming to enhance its performance and computational efficiency. Each hyperparameter was meticulously selected to maximize the model's generalization capability and effective learning from the training data. Below, we elaborate on these selections:

Table 2
Detailed AE Model Architecture for Anomalous Sound Detection

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 320)	0
dense (Dense)	(None, 64)	20,544
dense_1 (Dense)	(None, 64)	4,160
dense_2 (Dense)	(None, 8)	520
dense_3 (Dense)	(None, 64)	576
dense_4 (Dense)	(None, 64)	4,160
dense_5 (Dense)	(None, 320)	20,800

- **Learning Rate:** The learning rate for the Adam optimizer was set to 0.001. This value determines the step size at each iteration while moving toward a minimum of the loss function. This value was chosen to balance the speed of learning with the stability of the training process. A value too high might lead to overshooting the minimum, while a value too low might result in slow convergence.
- **Batch Size:** The batch size was set to 128. This parameter dictates the number of samples that are processed before the model weights are updated. A batch size of 128 was chosen to strike a balance between computational efficiency and the quality of gradient estimates. A larger batch sizes can be computationally faster but might lead to less accurate gradient estimates, while smaller batch sizes might provide better estimates but could slow down training.
- **Number of Epochs:** The model was trained for 100 epochs. An epoch signifies one complete pass through the entire training dataset. The choice of 100 epochs was made to allow the model sufficient iterations to learn the underlying patterns in the data. However, early stopping was implemented to prevent overfitting, halting the training if the validation loss did not improve for a certain number of consecutive epochs.
- **Trainable parameters:** A total of 50,760 trainable parameters were included in the model.

3.5 Training and Evaluation

- **Training:** The models are trained on the normal audio samples from the train directory in the development dataset. The training objective is to minimize the MSE loss.
- **Evaluation:** Both the trained models are evaluated on the test directory development dataset, comparing their performance in identifying known anomalies. AE's performance is evaluated using precision, recall, accuracy, F1 score, and AUC. The VAE, however, also regularizes its latent space to ensure the learned distribution aligns with a prior distribution. So, evaluation focuses on distinguishing normal from anomalous data, emphasizing AUC and pAUC metrics for a comprehensive understanding of the model's performance.

4 Experimental Setup and Results

4.1 VAE

To evaluate the proposed model's performance, several metrics were utilized: precision, recall, Receiver Operating Characteristic (ROC) curve/Area Under the Curve (AUC) [8].

For this report, we specifically present the AUC and partial AUC (pAUC) metrics, as they provide a comprehensive evaluation of the model's capability to distinguish between normal and anomalous data points.

As presented in Table 2, by the end of model training, the proposed VAE network achieved the following AUC and pAUC scores across different slider IDs

Table 3
Evaluation Metrics for the VAE Model Across Different Slider IDs

Slider ID	AUC	pAUC
00	0.950449	0.759462
02	0.783333	0.646757
04	0.932416	0.675931
Average	**0.888733**	**0.694050**

These results indicate a robust performance of the VAE model, with an average AUC of 0.888733 and an average pAUC of 0.694050 across the evaluated sliders. The high AUC scores reflect the model's strong ability to correctly classify normal and anomalous data points, demonstrating its effectiveness in anomaly detection tasks.

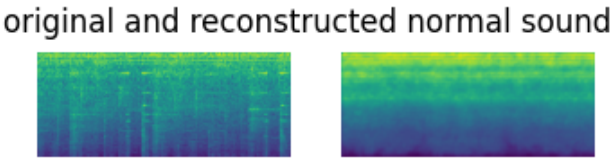

Figure 5. Original and Reconstruction of a normal sound

This visual comparison shows the model's ability to reproduce the temporal and frequency characteristics of the original sound, illustrating the VAE's effectiveness in capturing and reconstructing normal sounds. The similarities between the two spectrograms demonstrate the model's performance, while any differences may point to areas for future improvement.

4.2 AE

The AE model's performance (Figure 6) was evaluated using a range of metrics, including precision, recall, accuracy, F1 score, and the area under the receiver operating characteristic curve (AUC). These metrics provide a comprehensive assessment of the model's ability to accurately distinguish between normal and anomalous sounds.

The model was trained exclusively on normal audio samples from the training set. Following training, the model's efficacy was rigorously tested on a separate test set comprising both normal and anomalous samples. The reconstruction error [9], a measure of the discrepancy between the original and reconstructed audio, was computed for each sample as seen in Figure 7. A threshold was then established to categorize samples as either normal or anomalous based on their reconstruction error, seen in Figure 8.

Table 4
Basic Autoencoder Metrics

Metric	Value
Precision	52.7%
Recall	62.0%
Accuracy	78.3%
F1 Score	56.9%
AUC	0.7662

The AE model's performance, while achieving a decent AUC of 0.7662, reveals limitations in other evaluation metrics, particularly its precision (52.7%) and recall (62.0%), indicating a

tendency to both misclassify normal sounds and miss actual anomalies.

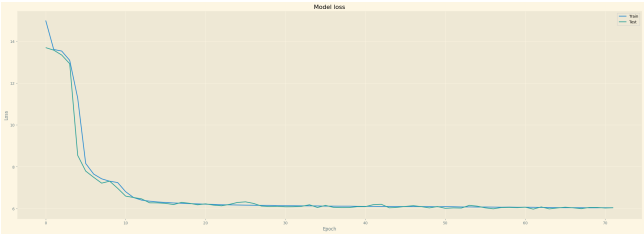

Figure 6. Training and Validation Loss Over Epochs for AE

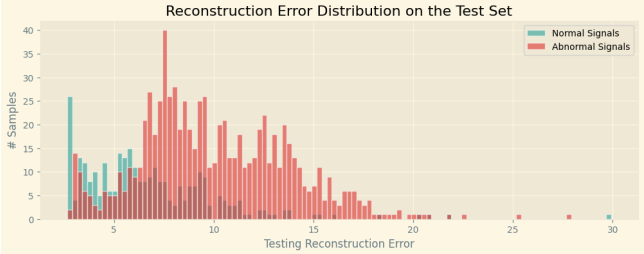

Figure 7. Reconstruction Error Distribution for Normal/Abnormal Signals on Test Set

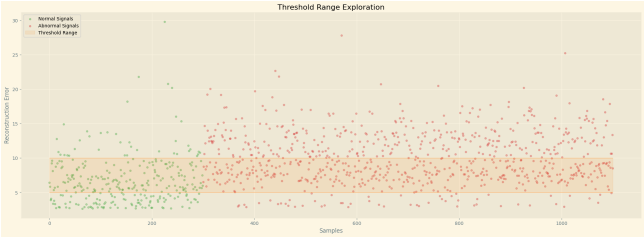

Figure 8. Threshold Range separation of Normal/Abnormal Signals

5 Discussion

For this project, both the autoencoder (AE) and the variational autoencoder (VAE) achieved an AUC greater than 0.5, indicating that both models can effectively classify the sounds. However, the VAE demonstrated superior performance, with an average AUC of 0.89, compared to the classical autoencoder's AUC of 0.7662. Although the implementation of a VAE is more complex than that of a traditional autoencoder and requires more resources for training, the VAE contains 861,000 parameters, while the classical autoencoder has only 50,760 parameters. This increased complexity translates into better performance for classifying anomalous sounds.

5.1 Effectiveness of the Models

The high AUC scores for both the CAE and the C-VAE models reflect their effectiveness in detecting anomalies in audio signals. The CAE's AUC of 0.7662 suggests it is quite capable of distinguishing normal from anomalous sounds, while the C-VAE's higher AUC of 0.89 indicates an even stronger ability to correctly classify these sounds.

5.2 Comparison of the Models

Between the two models, the C-VAE performs better in detecting anomalies. This superior performance can be attributed to the VAE's ability to model the underlying data distribution more effectively through its probabilistic approach, allowing it to generalize better to unseen data. Additionally, the VAE's complex architecture, with a significantly higher number of parameters, provides it with more capacity to capture intricate patterns in the data.

Table 5
AUC metrics compares between AE and VAE

	VAE	AE
AUC	0.888733	0.7662

5.3 Limitations of the Approach

Despite their effectiveness, using AEs and VAEs for anomaly detection in audio signals has some limitations. The primary limitation is their requirement for a large amount of normal data for training, as the models learn to reconstruct normal patterns. Any imbalance or lack of diversity in the training data can affect the model's performance. Additionally, the complexity of VAEs necessitates more computational resources, which can be a constraint in real-world applications.

5.4 Future work

To improve the performance of anomaly detection models, future research could explore several directions. One approach is to incorporate more advanced architectures, such as recurrent neural networks (RNNs), which can better capture temporal dependencies in audio signals. Another direction is to use transfer learning from pre-trained models on similar tasks, which could enhance performance with less training data.

6 Conclusion

In this study, we tackled the challenge of detecting anomalous sounds in factory environments thanks to the implementation and comparison of two models: Autoencoders (AEs) and Variational Autoencoders (VAEs).

We began by analyzing the MIMII dataset: audio recordings of slide rail machinery under both normal and anomalous conditions. The dataset was carefully preprocessed to ensure uniformity and effectiveness in model training, including transforming audio signals into spectrograms for enhanced feature extraction.

Then we built a convolutional architectures for both AEs and VAEs. The AE model was designed to learn the normal patterns of the audio signals and flag significant deviations, while the VAE model aimed to create a probabilistic representation of the data, offering a more robust mechanism for anomaly detection. Both models were implemented using PyTorch and trained extensively to optimize their performance.

We measured the models' effectiveness using metrics such as AUC, accuracy, precision, recall, and F1-score. Our results indicated that while both models performed well in detecting anomalies, the VAE demonstrated superior accuracy and robustness, thanks to its probabilistic nature and advanced architecture.

In conclusion, our research demonstrates the potential of VAEs in enhancing the accuracy of anomaly detection in industrial audio signals.

Acknowledgements

This research received support during the AML course, instructed by Professor Pietro MICHIARDI, Head of the Data Sci-

ence Department at EURECOM, France.

References

[1] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi. MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, November 2019.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.

[3] What Is Balanced And Imbalanced Dataset? `https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5`

[4] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504-507, 2006.

[5] Introduction to Balanced and Imbalanced Datasets in Machine Learning. `https://encord.com/blog/an-introduction-to-balanced-and-imbalanced-datasets-in-machine-learning/`

[6] Intuitively Understanding Variational Autoencoders `https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf`

[7] What is Adam Optimizer? `https://www.analyticsvidhya.com/blog/2023/09/what-is-adam-optimizer/`

[8] How to Calculate and Use the AUC Score `https://towardsdatascience.com/how-to-calculate-use-the-auc-score-1fc85c9a8430`

[9] Mana Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, pages 4-11, 2014.