JOHANNES KEPLER
UNIVERSITY LINZ

Author
Lukas Selch
11941656

Submission
Institute of Computational Perception

Thesis Supervisor
Paul Primus

December 2023

Semi-Supervised Anomaly Detection in Respiratory Sounds: A Comparative Study of Reconstruction and Density Estimation Methods

Bachelor's Thesis

to confer the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

JOHANNES KEPLER UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Abstract

Space for your abstract.

Contents

Chapter 1

Introduction

Respiratory diseases are a leading cause of premature mortality worldwide. With over four million annual deaths attributed to these diseases, early identification and treatment efforts are imperative [12]. The use of chest auscultation, a technique in which respiratory sounds are analyzed with instruments like stethoscopes, is a simple and effective way for diagnosing respiratory diseases.
Automated systems for detecting sound anomalies have become of increasing relevance in the medical field and are driving machine learning research [5]. They have the potential to improve diagnostic accuracy for healthcare professionals and provide initial assessments for patients, ultimately leading to more efficient allocation of healthcare resources.

1.1 Motivation

1.2 Objectives and Approach

1.3 Outline

Chapter 2

Theoretical Background

2.1 Respiratory Sounds

The respiratory system, which includes the airways and lungs, plays a crucial role in gas exchange, a vital function in the human body. Respiratory sounds, created by airflow during breathing, can reveal a lot about respiratory health [10]. These sounds, observed through a process called 'auscultation'—listening to the chest with a stethoscope—are critical for detecting respiratory diseases. This method is cost-effective, non-invasive, and a standard part of physical examinations [5].
We can categorize respiratory sounds into normal and anomalous based on their characteristics during auscultation. Normal sounds are typically heard during inhalation and at the start of exhalation, within a frequency range of 100 to 1000 Hz [5]. In contrast, abnormal sounds come in various forms. Here, we will discuss two common types: Wheezes and Crackles.

Wheezes are long and musical sounds that last over 100 milliseconds. They occur during inhalation and exhalation, often caused by narrowed or restricted airways. Their frequency usually falls between 100 to 1000 Hz, but higher harmonics are also possible [5].

Crackles, conversely, are brief, non-musical sounds that signal sporadic airway openings, often due to secretions. We can further divide them into Fine and Coarse Crackles. Fine Crackles are short, with a frequency of around 650 Hz and a duration of about five milliseconds. Coarse Crackles last longer, over 15 milliseconds, and occur at lower frequencies, below 350 Hz [5].

Understanding these distinctions in pulmonary sounds is vital for developing automated systems for their detection. Electronic stethoscopes can convert lung sounds into digital signals, enabling the use of advanced anomaly detection algorithms in computer-aided medical diagnosis.

2.1.1 Digital Representation of Sound

Sound signals, variations in air pressure known as sound waves, can be digitally represented in several ways. The following will give an overview of the used methods in this thesis.

Waveforms are the most straightforward representation, where the sound signal is a sequence of numbers $\mathbf{x}_n$ representing air pressure at time step $n \in \mathbb{N}$. The key parameter here is the sampling rate, which dictates how often the audio signal is sampled per second [3].

Spectrogram representations, unlike waveforms, allow for a visual examination of the sound signal. This involves transforming the signal from a real-valued time-domain to a complex-valued frequency-domain representation using the Discrete Fourier Transform (DFT), mathematically defined as

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k \times n}{N}}$$

. The result of this transformation provides a good overview of the frequencies that make up the sound signal. However, we are more interested in local events for non-stationary signals like respiratory sounds.
The Short Time Fourier Transform (STFT) helps obtain a representation that summarizes the sound signals' constituting frequencies while showing local changes in their distribution. It first divides the signal into short slices, also called windows. It then applies a windowing function to each slice, gradually reducing the signal amplitude towards the edges. This continuity between the windows is crucial for minimizing spectral leakage, a phenomenon where sudden changes in the signal between the windows get misinterpreted as sudden changes in the original signal. Finally, STFT involves applying a DFT on every window. We obtain a time-frequency representation typically visualized by converting X_k to the log-spectrum $20\log_{10}\|X_k\|$ [3].
Mel-Spectrograms refine this representation by aligning frequencies to the mel scale, approximating human hearing perception. This transformation is mathematically expressed as $m = 2595\log_{10}(1 + \frac{f}{700})$ as formulated by O'Shaughnessy (1987) [21].

Mel-Frequency Cepstral Coefficients (MFCCs) are used for dimensionality reduction in spectrograms to preserve essential information while reducing the number of coefficients. The process involves performing DFT on the waveform, computing the log amplitude spectrum, transforming the spectrum to the mel scale, and finally applying the Discrete Cosine Transform, a simplified version of the DFT resulting in a real-valued representation [3]. While not easily interpretable by humans, the resulting cepstrum is highly relevant for input into machine learning algorithms.

2.2 Fundamentals of Anomaly Detection

Anomaly detection is a process that identifies data points deviating from expected patterns, known as anomalies or outliers [7]. These deviations often signal critical changes in a system, requiring intervention. In healthcare, as discussed in section 2.1, anomalies in respiratory sound patterns can indicate various conditions.
Anomaly detection algorithms characterize normal behavior, flagging deviations as anomalies. However, the rarity of anomalies and the uncertainty in their distribution poses significant challenges. Because anomalies are much more infrequent by nature, datasets typically are heavily unbalanced and contain a much larger number of normal samples than anomalies. Furthermore, the anomalous data points can exhibit various forms of non-normality, meaning there can be a substantial variation within the set of outliers [23]. It is also essential to balance false positives and negatives based on the specific application domain.
Outlier detection can be categorized based on the data available:

1. **Supervised Anomaly Detection**: This method uses a fully labeled dataset to differentiate between normal and abnormal data points. However, the lack

of thorough datasets representing all the variance in anomalies limits these approaches.

2. **Unsupervised Anomaly Detection**: More common in real-world scenarios, this approach uses only normal data points, requiring the system to learn their defining characteristics autonomously.
3. **Weakly Supervised Anomaly Detection**: This approach, which we focus on in our research, uses primarily normal data points with a significantly smaller subset of anomalies. It is advantageous as it requires fewer anomalous data points than fully supervised methods while allowing for generalization abilities.

Anomaly detection algorithms may provide a direct classification of data points as normal or anomalous or output an anomaly score indicating deviation from normality. This score helps identify anomalous samples by finding a threshold above which all samples can be considered anomalous. Traditional methods like K-nearest neighbor (KNN) and Support Vector Machines (SVMs) rely on distance metrics or decision boundaries to identify outliers. KNN works by identifying data points significantly distant from the closest set, while SVMs create a boundary between classes, effective in scenarios with clear separation [7].
However, these traditional methods can be limited in handling complex, high-dimensional, and noisy data or when anomalies closely resemble normal data. Deep learning methods can extract features from raw data and have shown remarkable effectiveness in learning complex patterns, such as those in audio signals. We will explore two distinct deep-learning architecture families used in this research, highlighting their suitability for anomaly detection in respiratory sounds.

2.3 Reconstruction-Based Methods

Reconstruction-based methods in anomaly detection use reconstruction errors as anomaly scores to identify anomalies. These methods involve two primary steps: dimensionality reduction and data reconstruction. Initially, the data is transformed into a latent, more compact representation in a latent space. This space aims to retain essential data features while discarding noise and irrelevant details. The subsequent step involves reconstructing the original data from this compact representation. The core challenge lies in achieving a balance where the reduced representation is compact yet retains sufficient information for accurate reconstruction without overfitting.
These models are trained unsupervised, using only normal data points. This approach ensures that the model learns typical patterns of such. The reconstruction error is calculated during training, reflecting the accuracy with which the model can recreate the input data. The underlying assumption is that a model trained on normal data will yield minimal error in reconstructing similar data. However, when encountering anomalous data that deviates from learned patterns during evaluation or inference, the model faces difficulties in reconstruction, resulting in a higher reconstruction error. This error then serves as an anomaly score, with higher errors indicating a greater likelihood of anomaly and vice versa.

2.3.1 Essentials of Autoencoders

Autoencoders are a prevalent neural network architecture in reconstruction-based methods, known for their ability to learn compact data representations unsupervised. An autoencoder comprises two main components: the Encoder and the Decoder.

- **Encoder** ($A : \mathbb{R}^n \to \mathbb{R}^l$): This component maps high-dimensional input data (dimensionality n) into a lower-dimensional latent space (dimensionality $l < n$). This process compresses the data into an efficient, compact form.
- **Decoder** ($B : \mathbb{R}^l \to \mathbb{R}^n$): The Decoder performs the inverse function, reconstructing data from the latent space back to its original dimensionality. Its goal is to replicate the original input as closely as possible.

The optimization challenge in autoencoders can be formulated as follows:

$$\arg\min_{A,B} E[\Delta(\mathbf{x}, B \circ A(\mathbf{x}))],$$

where E represents the expected value of the reconstruction loss Δ for input $\mathbf{x}$ [4].

2.4 Density Estimation Methods

Density estimation methods in anomaly detection revolve around the concept of probability distributions. Probability distributions are mathematical functions that describe the probability associated with each possible value of a random variable. Suppose the random variable can take any value within a specific range. In that case, the probability distribution is continuous and can be described by a Probability Density Function (PDF), providing a probability of a random variable falling within a specific range.
For a continuous real-valued random variable X, the PDF is defined as

$$P(a \leq X \leq b) = \int_a^b f(x)\,dx$$

for all $a, b \in \mathbb{R}$ [16]. Here, $f(x)$ represents the probability density function of X. It is important to note that the PDF does not give probabilities directly. Instead, the area under the PDF curve between these points gives the probability of X falling within the interval from a to b.
Density estimation involves estimating the PDF from observed data by estimating a joint distribution $p(\mathbf{x})$ from a set of examples $\{\mathbf{x}^{(t)}\}_{t=1}^{T}$ [14]. The estimation can be parametric, where the data is assumed to follow a known distribution like Gaussian with the mean and standard deviation as parameters, or nonparametric, which does not presume a specific distribution and directly estimates the PDF from the data. Nonparametric methods can handle more complex distributions but usually require more data to produce an accurate estimate.
In anomaly detection, estimating the PDF of a dataset helps identify regions of low probability. Data points in these regions are potential anomalies, making density estimation a powerful tool for detecting outliers.

2.4.1 Introduction to Masked Autoencoders

Masked Autoencoders for Distribution Estimation (MADE) extend the concept of autoencoders so that they can understand the data distribution. Unlike tradi-

tional autoencoders, MADE enforces the autoregressive property and considers the input data order so that each output part is influenced only by preceding input parts.
The autoregressive property is accomplished by masking the weights in each autoencoder layer, controlling the information flow. Each neuron is labeled with a number from 1 to $D-1$ (where D is the input dimensionality) and the following rule is applied to determine allowed connections: a neuron in layer l (called k') can only be connected to a neuron in the previous layer $l-1$ (called k) if its label is greater than or equal to the label of k. Mathematically, Germain et al. (2015) states this concept as

$$M^{\mathbf{W}^l}_{k',k} = \begin{cases} 1, & \text{if } m^l(k') \geq m^{l-1}(k) \\ 0, & \text{otherwise.} \end{cases}.$$

Here, $M^{\mathbf{W}^l}_{k',k}$ is the weight matrix mask that determines whether a connection is allowed or not. For the output layer, the rule needs the slight modification of making the condition strict ($m^l(k') > m^{l-1}(k)$) to maintain the autoregressive property.
MADE's architecture allows for calculating the probability of observing the input $\mathbf{x}$ as

$$p(\mathbf{x}) = \sum_{d=1}^{D} p(x_d|\mathbf{x}_{<d}).$$

This probabilistic model is beneficial in anomaly detection as it can identify data points with low probability, indicating potential anomalies.

2.5 Evaluation Metrics for Model Comparison

Evaluating and comparing the performance of different anomaly detection models requires carefully chosen metrics relevant to real-world applications and offers clear, intuitive interpretations of the models' effectiveness. Before introducing the five core metrics considered in this thesis, we need to understand the concept of a confusion matrix. In anomaly detection, we can categorize a model's prediction outcome into four types: true negatives (correct normal prediction), false negatives (anomalies incorrectly labeled as normal), true positives (correct anomaly prediction), and false positives (normal data points mistakenly identified as anomalies). A confusion matrix summarizes these four outcomes.
Having introduced the needed terminology, we focus on defining the evaluation metrics.

Sensitivity (True Positive Rate - TPR)

$$\text{TPR} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}}.$$

Sensitivity measures the proportion of actual anomalies correctly identified. It ranges from 0 (no anomaly detected) to 1 (perfect anomaly detection), with a higher value preferable, especially in critical applications like healthcare, where leaving an anomaly undetected can have serious implications.

False Positive Rate (FPR)

$$\text{FPR} = \frac{\text{FalsePositives}}{\text{TrueNegatives} + \text{FalsePositives}}.$$

FPR assesses the proportion at which normal data points are incorrectly classified as anomalies. It ranges from 0 (no false alarms) to 1 (all normal data classified as anomalies). A lower FPR is desired, as it indicates fewer false alarms.

Specificity (True Negative Rate - TNR)

$$\text{TNR} = \frac{\text{TrueNegatives}}{\text{TrueNegatives} + \text{FalsePositives}}.$$

Specificity quantifies how well the model identifies normal data points. It also ranges from 0 (poor recognition of normal conditions) to 1 (excellent recognition of normal conditions). Higher specificity is essential to minimize false alarms in sensitive domains like medical settings.

Area Under the Curve (AUC)

The Area Under the Curve (AUC) is a metric calculated using the Receiver Operating Characteristic (ROC) curve, which portrays the True Positive Rate (TPR) against the False Positive Rate (FPR) within a unit square, with TPR on the Y-axis and FPR on the X-axis. This visualization emphasizes the trade-off between a model's capability to correctly identify anomalies (TPR) and its tendency to misclassify normal data as anomalous (FPR).
In this graph, a point at (0,0) signifies a model that categorizes all data as normal, while (1,1) indicates a model labeling everything anomalous. The theoretical perfect model achieves flawless classification, represented by (0,1). The ROC curve emerges by systematically adjusting the model's classification threshold and plotting corresponding TPR and FPR values, forming a curve that illustrates performance across various thresholds [11].
A random model aligns with the diagonal line from (0,0) to (1,1) in binary classification tasks like anomaly detection. Performance surpasses randomness when the ROC curve lies above this diagonal and diminishes when below. The AUC is a quantitative measure of this performance, quantifying the area enclosed by the ROC curve and the x-axis. It ranges from 0.5 (no discrimination ability) to 1 (perfect classification). Notably, models with an AUC below 0.5 can be adjusted to achieve better-than-random classification.
A high AUC is crucial in medical diagnostics to detect pathologies while avoiding false alarms and overtreatment of healthy patients.

Balanced Accuracy (BALACC)

Balanced Accuracy (BALACC) assigns equal importance to correctly identifying anomalies and normal data points. It is calculated as the average of Sensitivity (True Positive Rate) and Specificity (True Negative Rate):

$$\text{BALACC} = 0.5 \times (\text{Sensitivity} + \text{Specificity})$$

This metric is especially useful in scenarios where the dataset is unbalanced, a common occurrence in anomaly detection where anomalies are much rarer

than normal samples. In such cases, traditional accuracy metrics can be misleading. For example, consider a dataset where 95% of the data points are normal and only 5% are anomalies. A model that naively classifies every data point as normal would achieve a 95% accuracy rate, which is misleadingly high. Balanced accuracy corrects this distortion by considering both the ability to detect anomalies (sensitivity) and recognize normal instances (specificity), providing a more truthful measure of a model's performance. In our example, it would only result in a 50% balanced accuracy.
Balanced accuracy has a goal similar to the AUC's: correctly identifying diseased and healthy individuals.

Chapter 3

Literature Review

3.1 Current State of Respiratory Sound Analysis

Lung auscultation is the standard method of diagnosing respiratory disease by listening to the patient's lungs through the chest. However, this approach, which relies on manual assessment by healthcare professionals, has several limitations. Its effectiveness depends on the physician's skill, experience, and auditory sensitivity, leading to potential inaccuracies in diagnosis [22]. In addition, manual auscultation is typically limited to clinical settings, missing critical auditory cues that may occur outside of these settings, such as nocturnal breath sounds common in conditions such as asthma [27].
These limitations, combined with technological advances, have led to the development of computerized respiratory sound analysis. In this approach, lung sounds are digitally recorded and then analyzed. Early techniques focused on the graphical representation of sound waves, allowing medical professionals to identify abnormalities visually. However, this method did not fully mitigate the risk of human error. Subsequently, statistical approaches were developed to assess the frequency of specific respiratory events based on historical data patterns. According to systematic reviews [22], machine learning-based approaches provide the most promising results but were limited by the need for sufficiently large data sets.

3.1.1 The ICBHI Challenge 2017

During the 2017 Annual International Conference on Biomedical and Health Informatics, a central challenge was launched in response to the scarcity of comprehensive lung sound data. This challenge aimed to foster the development and evaluation of advanced algorithms for automated lung sound classification using a novel dataset curated specifically for this purpose. Known as the Respiratory Sound Database [28], this collection stands out as one of the field's earliest and most comprehensive publicly available datasets, comprising 6898 respiratory cycles from 126 patients. These recordings, collected by professional teams in Greece and Portugal, represent diverse audio samples, capturing sounds from healthy individuals and patients suffering from lung diseases such as COPD, asthma, and bronchiectasis. Each breathing cycle in the database is annotated by domain experts and categorized as normal, with wheezes, crackles, or both wheezes and crackles. The challenge encouraged many submissions, showcasing a range of innovative machine-learning approaches. Below, we compare a selection of these methods.

3.1.2 Existing Approaches

Starting with traditional artificial intelligence methods, Jakovljević and Lončar-Turukalo (2018) published their work based on hidden Markov models (HMM) alongside the paper introducing the Respiratory Sound Database. [18]. Using MFCCs as features, they employed a four-class classifier with the official 60/40 split at the recording level, using 60% of the data for training and the remaining 40% for evaluation. The four classes were healthy, crackles, wheezes, both crackles and wheezes. A balanced accuracy score of 0.39 was achieved, with a sensitivity of 0.38 and a specificity of 0.41.
Chambres et al. (2018) used boosted decision trees (BDT) to address the four-class classification task [6]. They used the same 60/40 split and MFCCs as features. The model architecture significantly improved the balance accuracy to 0.49, with a sensitivity of 0.78 and a specificity of 0.21.
Shortly after, the use of neural networks gained traction. Ma et al. (2019) proposed the use of a bi-ResNet (LungBRN) architecture [30] consisting of multiple concatenated convolutional neural network layers [20]. Using the same split as the other approaches but short-time Fourier transform (STFT) and wavelet analysis to extract features, they achieved an official balanced accuracy of 0.5, specificity of 0.69, and sensitivity of 0.31 for the four-class problem.
The Microsoft Research India team around Gairola et al. (2021) published their RepireNet [13] network and benchmarked it in various data splits and binary and four-class classification problems. The backbone is blocks of ResNet34 [17] deep convolutional neural networks (CNN). Using MelSpectrograms as features, their baseline CNN achieved 0.55 balanced accuracy on the official 60/40 split, 0.66 balanced accuracy for the four-class problem on a self-defined random 80/20 split at the breathing cycle level, and 0.72 on the same 80/20 split but treating the problem as a binary classification, which allows for an easier comparison to a anomaly detection setting.

Table 3.1: Performance comparisons of the showcased models

Model	Split	Features	Se	Sp	BALACC
HMM	60/40 4 class	MFCC	0.38	0.41	0.39
BDT	60/40 4 class	MFCC	0.78	0.21	0.49
LungBRN	60/40 4 class	STFT + Wavelet	0.69	0.31	0.5
RespireNet CNN	60/40 4 class	MelSpectrogram	0.39	0.71	0.55
RespireNet CNN	80/20 4 class	MelSpectrogram	0.54	0.79	0.66
RespireNet CNN	80/20 2 class	MelSpectrogram	0.61	0.83	0.72

It is important to note that all mentioned respiratory sound analysis approaches rely on treating it as a supervised classification task. While effective in their context, these methods assume the availability of extensive labeled data representing various specific respiratory conditions. However, such comprehensive data sets are rarely readily available in real-world scenarios. Furthermore, the strict categorization of respiratory sounds into predefined classes may overlook respiratory anomalies' nuanced and unpredictable nature. Therefore, the remainder of this thesis will explore respiratory sound analysis through the lens of anomaly detection.

3.2 Respiratory Sound Analysis from an Anomaly Detection Perspective

Using anomaly detection methods to solve sound analysis problems is not new. In particular, these methods have proven their effectiveness in industrial sound analysis, as demonstrated in Task 2 of the annual DCASE Challenge [9], where machine condition monitoring is performed by observing the sound produced by these machines. The sound emitted can be either normal or anomalous, and machine learning algorithms learn to understand the characteristics of healthy machine sounds to accurately predict machine failure in the case of anomalous sounds such as rattling or whirring.
A similar approach can be used in breathing sound analysis. The different anomalous respiratory sounds can all be grouped into a single anomaly class, and anomaly detection models can learn the constitution of healthy respiratory cycles. If a sample deviates significantly from the learned representation of a healthy sound, the system can flag it as anomalous.

3.3 Variational Autoencoders

Cozzatti et al. (2022) [8] explored the first anomaly detection approach to the respiratory sound database. Their work used MFCCs as features, and the breathing cycles containing wheezes, crackles, or both were all summarized in an anomaly class. A Variational Autoencoder (VAE) was trained using only known healthy breathing cycles.
Variational Autoencoders are similar to Autoencoders, consisting of an encoder and a decoder part. By comparison, the encoder of a VAE uses variational inference to output the parameters of a continuous and easily sampled distribution, usually the mean and standard deviation of a Gaussian [8]. As a result, the input to the decoder is a single sample from that predicted distribution. This allows the model to provide a measure of certainty of the reconstructed data using the variability of the latent space [2].

Training the VAE with normal sounds only teaches it to reconstruct physiological respiratory cycles accurately. The reconstruction error reported by the model is small in this case. When the model attempts to reconstruct a respiratory sound with pathologies, the parameters of the Gaussian will most likely not match the parameters of the learned distribution of healthy sounds. Thus, the reconstruction will have a higher error. The paper then used a small subset of the original dataset containing healthy and unhealthy lung sounds to determine a threshold in the reconstruction error above which all higher errors should be marked as anomalous, making the process weakly supervised. The proposed model achieved competitive results in the binary class problem, with a balanced accuracy of 0.57 for the official 60/40 split and 0.6 for a random 80/20 split.

Table 3.2: Performance of the proposed method

Split	Se	Sp	BALACC
60/40	0.33	0.80	0.57
80/20	0.58	0.61	0.60

3.4 Group Masked Autoencoders

In section 2.4.1, we have discussed how Masked Autoencoders are an alternative anomaly detection approach to generative models by evaluating probability densities. The basic concept focused on modifying an existing autoencoder structure to satisfy the autoregressive property by masking the weights of the neural network layers so that each output dimension depends only on the preceding input dimensions.
When working with sound data representations such as MelSpectrograms or MFCCS, the focus shifts from the autoregressive ordering of individual input dimensions to the ordering of sound frames. Here, Group Masked Autoencoders (GMADE) [15] provide a more tailored approach for audio anomaly detection tasks where temporal context is important. GMADE differs from traditional MADE in that it does not split the joint distribution into individual dimensional conditions but rather into conditionals on the grouped frames. This approach is particularly relevant when dealing with sound data, where each time frame in a MelSpectrogram is considered a separate group.
In GMADE, the input space has the dimensionality $T \times M$, where T is the number of frames concatenated in the input and M is the number of Mel frequency bands. If an input sample can be thought of as $\mathbf{t} = [\mathbf{t_{i+1}}, \mathbf{t_{i+2}}, ..., \mathbf{t_{i+T}}]$ with $\mathbf{t_i} \in \mathbb{R}^{M \times 1}$, the joint density can be decomposed as

$$p(\mathbf{t}) = \sum_{i=1}^{T} p(\mathbf{t_i} | \mathbf{t}_{<i})$$

where the probability of each frame depends on all previous frames and their mel bins and no other frames [15]. To maintain the autoregressive property, the generation of the weight matrices must be slightly adapted from the MADE approach to assign labels to the neurons only from 1 to $T - 1$ to correctly zero connections between groups instead of units.
The paper also explored orderings other than causal, where a frame can only depend on its predecessors. Backward ordering predicts the probability density of frames given only their succeeding frames, while middle frame ordering attempts to predict the middle frame given only the frames surrounding it. Ensembles of all three approaches were also evaluated. GMADE achieved state-of-the-art results in Task 2 of the DCASE Challenge 2020 in the machine condition monitoring task, especially for non-stationary sounds. While this is promising for respiratory sound analysis due to the non-stationary nature of lung sounds, the efficacy of GMADE in detecting anomalies in respiratory sounds is yet to be tested.

We highlighted the significant progress in respiratory sound analysis, moving from traditional auscultation to advanced computational methods. Developing the Respiratory Sound Database has been an important catalyst, enabling the application of machine learning methods in this domain. Notably, the ICBHI Challenge 2017 provided a platform for showcasing diverse approaches, ranging from hidden Markov models and boosted decision trees to more advanced neural network architectures like LungBRN and RespireNet.
The shift towards anomaly detection methods marks a transition in the approach towards the dataset. Cozzatti et al. (2022) introduced a novel approach using Variational Autoencoders (VAE), demonstrating the potential of reconstruction-based models in identifying respiratory anomalies. Focusing solely on normal respiratory cycles for training, this method distinguishes abnormal sounds based on the deviation in reconstruction error, achieving notable results in binary classification tasks.

Building upon these insights, we will next explore and compare two approaches to respiratory sound analysis: reconstruction-based methods, represented by VAE, and density-estimation-based approaches, represented by Group Masked Autoencoders (G-MADE). We will reevaluate the VAE for its effectiveness in distinguishing between normal and pathological respiratory sounds and assess G-MADE's ability to accurately model the probability densities of respiratory sounds.

Chapter 4

Methodology

4.1 Dataset

4.1.1 Definition

The Respiratory Sound Database was established for the 2017 International Conference on Biomedical Health Informatics [28]. It comprises an open-access collection of audio recordings from 126 patients captured via electronic stethoscopes. These recordings encompass a diverse patient demographic, varying in sex and age, and were obtained using different recording devices amid typical clinical environmental noise.
The dataset includes individuals both with and without respiratory diseases. It features explicitly patients diagnosed with one of three conditions: Chronic Obstructive Pulmonary Disease (COPD), Lower Respiratory Tract Infection (LRTI), and Upper Respiratory Tract Infection (URTI). Nine hundred twenty recordings were gathered, amounting to 5.5 hours of audio, with individual recordings ranging from 10 to 90 seconds. Each recording contains multiple breathing cycles, encompassing both inhalation and exhalation, and these cycles may be normal or exhibit signs of anomalies such as wheezes or crackles.

Crackles are brief, sharp, non-melodic sounds typically heard during inhalation but can also occur during exhalation. They are categorized into two subtypes:

- Fine crackles are high-pitched, short sounds lasting about five milliseconds and associated with fluid in small airways.
- Coarse crackles are lower-pitched, longer sounds lasting about 15 milliseconds, indicating disruptions in larger airways.

Wheezes are longer, distinctive sounds, often exceeding 100 milliseconds. Their musical tone is apparent, presenting as sinusoidal waves in sound analyses. These waves predominantly fall within the 100 to 1000 Hz frequency range, sometimes producing harmonics above [5].

In the entire dataset, domain experts annotated 6.898 respiratory cycles. Among these, 3.642 cycles show no anomalies, 1.864 contain crackles, 886 feature wheezes, and 506 simultaneously exhibit both crackles and wheezes.

4.1.2 Data Splitting

We divided the dataset into training, validation, and test sets for a semi-supervised learning approach and to assess model generalization. Following the common practice in the 2017 ICBHI Challenge literature, we adopted an

80/20 split between training and test data, focusing on individual breathing cycles. This split ensures comparability with previous studies. Initially, the dataset underwent a random split, stratified to maintain equal proportions of normal and anomalous samples, resulting in separate training and test sets. Subsequently, the training set was further divided by performing another stratified random split, with 20% forming the validation set and the remaining 80% left in the training set cleared of anomalous data to consist solely of normal breathing cycles. We used `train_test_split` from the scikit-learn library [25] to conduct those splits.
We acknowledge the limitations of this approach. Notably, excluding 2.084 anomalous samples from the training set might affect the model's generalization ability. While reallocating these to the validation or test sets could improve generalization assessment, it raises concerns about data imbalance. Additionally, the initial splitting at the breathing cycle level poses a risk of data leakage, as cycles from the same recording could be distributed across training and test sets, potentially overestimating model performance metrics.
To mitigate this, the dataset creators suggested a 60/40 recording-level split where a patient's recording can only appear in either data split. We adopted a similar recording-level split, but we allocated 80% for training and 20% for testing to keep enough information available in the unsupervised mode. While ensuring a big enough training dataset even after removing anomalous data, this approach also minimizes data leakage.
By employing both splitting strategies, we aim to align our methodology with existing literature and robustly evaluate our model's performance, particularly regarding data leakage prevention.

4.2 Preprocessing

Effective preprocessing is crucial for transforming raw audio data into a format suitable for our machine-learning models. This section outlines the steps taken to achieve this. Initially, we loaded audio files, each comprising multiple respiratory cycles, along with their corresponding annotation files. These annotations, indicating each cycle's start and end times and the presence of crackles or wheezes, enabled us to extract individual cycles and assign binary labels (1 for abnormal cycles with crackles or wheezes and 0 for normal cycles).
Given the dataset's variety in sampling rates across different electronic stethoscopes, a uniform sampling rate was necessary. After reviewing literature [8, 29] and considering the Nyquist Theorem [26], we standardized all audio samples to a 4000 Hz sampling rate. This rate effectively captures wheezing sounds (with significant components below 2000 Hz), ensuring that both wheezes and crackles are accurately represented without aliasing.
To accommodate the fixed-size input requirement of our models, we normalized the length of respiratory cycles, which ranged from 0.2s to 16.2s, to a consistent 5s. The normalization was achieved by truncating longer sequences and zero-padding shorter ones.
Subsequently, we computed 13 MFCCs for each audio sample using PyTorch's [24] `torchaudio.transforms.MFCC` implementation. Adjustments to the underlying spectrogram calculations included setting the number of mels to 64, the size of the fast Fourier transform to 265 with a hop length of 128 and a maximum frequency of 2000 Hz. These parameters align with the implementation of Gairola et al. (2021).
We then applied the `torchaudio.transforms.AmplitudeToDB` transformation to scale the amplitude valued logarithmically. The final preprocessing step involved standardizing the MFCCs to have zero mean and unit standard deviation,

preparing them for efficient model processing.

4.3 Detailed Overview of the Models

In our research, we explored two distinct models to evaluate the efficacy of reconstruction-based and density estimation-based approaches in detecting anomalies in respiratory sounds. Our goal was to maintain consistency in training procedures and preprocessing across both models to ensure fair comparability. However, it is important to note that the architectures of these models are inherently different, and we will delve into the specifics of each.

4.3.1 Implementation of the VAE

The Variational Autoencoder (VAE) serves as our reconstruction-based model. It inputs Mel Frequency Cepstral Coefficients (MFCCs) and outputs reconstructed MFCCs of the same dimension. We began with the well-established DCGAN architecture and adapted it for one-dimensional convolution along the time axis.

Encoder: The Encoder is composed of five consecutive convolutional layers. Each layer includes a one-dimensional convolution (kernel size 4, stride 2, padding 1), followed by batch normalization and a LeakyReLU activation function with $p = 0.2$. After the convolution, there are two linear layers: one outputs the mean (μ), and the other outputs the log-variance ($\log(\mathrm{var})$). This setup is designed to model a Gaussian distribution in the latent dimension z, with $\log(\mathrm{var})$ enhancing numerical stability.

Sampling and Reparametrization Trick: We encounter a challenge in sampling from this Gaussian distribution, as it is not a differentiable process, which is essential for gradient-based optimization. To address this, we use the reparametrization trick. This method introduces a deterministic noise variable (ϵ) and computes the latent sample as $z = \mu + \sigma \cdot \epsilon$, enabling differentiability and backpropagation.

Decoder: The Decoder mirrors the Encoder, comprising five convolutional layers. The activation function is replaced with standard ReLU and the last convolutional layer, as it is the output layer of the Decoder, does not use batch normalization and has linear output activation. A linear layer initially processes the latent space sample, reshaping it for compatibility with the convolutional layers.

Initialization and Loss Function: We initialize all layer weights using the Xavier method. We combine mean squared error (MSE) loss for the reconstruction error and the Kullback-Leibler divergence (KLD) for a measurement of how well the output distribution of the Encoder aligns with the standard Gaussian distribution. The final loss is the weighted sum of both losses $\mathrm{Loss} = \alpha \cdot \mathrm{MSE} + (\alpha - 1) \cdot \mathrm{KLD}$, with α as the weight.

4.3.2 Implementation of the G-MADE

The Group Makes Autoencoder for Density Estimation (G-MADE) represents our alternative model for respiratory sound anomaly detection, operating on

the principles of autoregressive density estimation.

Model Architecture: At its core, G-MADE uses a modified linear autoencoder structure, where we replace the traditional linear layers with MaskedLinear layers that add the functionality for a configurable weight mask to enforce the autoregressive property. The model is constructed as a feed-forward network with several MaskedLinear layers followed by ReLU activation functions. The final output layer omits the ReLU activation for linear output.

```
class MaskedLinear(nn.Linear):
    def __init__(self, in_features, out_features, bias=True):
        super().__init__(in_features, out_features, bias)
        self.register_buffer('mask', torch.ones(out_features, in_features))

    def set_mask(self, mask):
        self.mask.data.copy_(torch.from_numpy(mask.astype(np.uint8).T))

    def forward(self, input):
        return F.linear(input, self.mask * self.weight, self.bias)
```

Listing 4.1: MaskedLinear PyTorch implementation as described by Karpathy, Andrej (2018) [19]

Mask Generation and Update: A significant aspect of the implementation is the generation and updating of the weight masks as they determine the connections between neurons in subsequent layers. The masks are dynamically generated based on the input ordering and connectivity of neurons and allow for varying ordering strategies, such as causal, backward, and middle frame orderings. The model can generate multiple masks to allow for the ensemble of different neuron connectivities. To illustrate the implementation, we consider the case of just one mask used.

```
def update_masks(self):
    # define the number of linear layers of the model
    L = len(self.hidden_sizes)

    # create a numpy random number generator instance
    rng = np.random.RandomState(self.seed)

    # define input order
    # here: forward ordering for 5 frames and 13 MFCCs
    expanded_order = np.tile([0, 1, 2, 3, 4], 13)

    # sample the order of the inputs and the connectivity of all neurons
    # store each layer's weights in self.m
    self.m[-1] = expanded_order # input ordering for first layer
    for l in range(L):
        # for each layer, assign label ranging from smallest label
        # of previous layer to the number of groups - 1 at random
        self.m[l] = rng.randint(self.m[l-1].min(), self.num_frames-1, size=
self.hidden_sizes[l])

    # construct the mask matrices according to the MADE definition
    masks = [self.m[l-1][:,None] <= self.m[l][None,:] for l in range(L)]
    # strictly larger in the case of output layer
    masks.append(self.m[L-1][:,None] < self.m[-1][None,:])

    # set the masks in all MaskedLinear layers
    layers = [l for l in self.net.modules() if isinstance(l, MaskedLinear)]
```

```
27         for l,m in zip(layers, masks):
28             l.set_mask(m)
```

Listing 4.2: PyTorch implementation of the mask generation, adapted from Karpathy, Andrej (2018) [19]

Loss Function: We used the Gaussian Negative Log Likelihood (GaussianNLL) for training G-MADE. This choice aligns with the model's focus on density estimation, as it effectively measures how well the model predicts the probability distribution of the output given the input.

Chapter 5

Experiments and Results

5.1 Experimental Setup

We conducted our experiments on a machine equipped with an AMD Ryzen 5 1600 CPU and a NVIDIA GeForce GTX 1070 GPU. The initial step involved establishing a baseline performance to later asses both models under various conditions. We utilized the PyTorch framework [24] for training, taking advantage of its pre-built functions for neural network training. The Adam optimizer was employed throughout the training processes, along with the `ReduceLROnPlateau` learning rate scheduler set to a patience of 5 epochs. Additionally, we implemented early stopping, evaluating the model's performance on the validation set after each training epoch, with a patience threshold of 10 epochs.
For optimal baseline determination, we used the Optuna framework [1] for hyperparameter optimization. The objectives included maximizing the AUC-Score and Balanced Accuracy on the validation set while minimizing the epochs required to achieve respectable results. This process yielded the following hyperparameters for our models:

VAE: The optimal batch size was determined to be 32, with a learning rate of 1e-4. The *nf* hyperparameter in the DCGAN architecture, which affects the depth of feature maps in both the generator and discriminator, was found to be most effective at 128. The size of the latent space vector was set to 128, and the alpha weighting factor balancing the Mean Squared Error loss and the Kullback-Leibler Divergence was adjusted to 0.5.

GMADE: A batch size of 128 and a learning rate of 1e-3 were optimal for this model. Using a single mask configuration without resampling the mask during training, we achieved the best results with a hidden layer structure of [256, 156, 512].

5.2 Generalization Capabilities

The ability of a machine learning model to generalize to unseen data is as crucial as its performance during training. We evaluated the generalization of our models using the test set, as described in section 4.1.2, with an 80/20 split. We assessed all three input orderings and their mean ensemble for the G-MADE model.
We established a binary classification threshold based on anomaly scores to gauge the Balanced Accuracy, True Positive Rate (TPR), and True Negative Rate (TNR). This threshold was determined using the validation set, which includes some anomalous data. We tested each anomaly score as a potential threshold,

selecting the one that maximized balanced accuracy on the validation set. This approach underpins the weakly-supervised aspect of our methodology.

Table 5.1: Experiment Results for VAE and G-MADE Models with Different Orderings

Model	Metric	Validation Set	Test Set
VAE	ROC-AUC	**0.64**	**0.62**
	BALACC	**0.61**	**0.61**
	TPR	0.84	0.84
	TNR	0.39	0.39
G-MADE (Forward)	ROC-AUC	0.59	0.57
	BALACC	0.57	0.57
	TPR	**0.90**	**0.90**
	TNR	0.24	0.24
G-MADE (Backward)	ROC-AUC	0.58	0.57
	BALACC	0.57	0.57
	TPR	0.78	0.78
	TNR	0.36	0.37
G-MADE (Mid-Frame)	ROC-AUC	0.56	0.55
	BALACC	0.56	0.57
	TPR	0.86	0.86
	TNR	0.27	0.29
G-MADE (Ensemble)	ROC-AUC	0.59	0.58
	BALACC	0.57	0.56
	TPR	0.71	0.70
	TNR	**0.42**	**0.43**

Our performance results indicate that with its reconstruction-based approach, the Variational Autoencoder (VAE) slightly outperforms the G-MADE models in accuracy and AUC-Score. However, all models show a notable number of false positives, possibly due to the high environmental noise in the recordings the models assume to be anomalies. The ensemble G-MADE model demonstrated the highest TNR, indicating a better balance in minimizing false positives while identifying healthy patients. Interestingly, while the ensemble improved the TNR and the AUC-Score marginally, it did not enhance the balanced accuracy compared to individual orderings due to the lower TPR.
All models are competitive to the weakly-supervised approach introduced by [8] and allow for a more nuanced inspection of the capabilities of the models.

5.3 Age and Gender Differences in Model Accuracy

Machine learning algorithms often suffer from imbalances in gender representation, mainly due to insufficient data from females, youths, and older individuals. This imbalance is critical in medical applications where fairness across different genders and age groups must be assured. To address this, we evaluated our models' performance on various subgroups within our dataset, using thresholds established as previously described.

Gender: We analyzed the models' performance by filtering the test set for male and female samples separately.

Table 5.2: Experiment Results by Gender for VAE and G-MADE Models

Model	ROC-AUC		BALACC		TPR		TNR	
	F	M	F	M	F	M	F	M
VAE	**0.59**	**0.64**	**0.59**	**0.61**	0.81	0.84	0.37	0.39
G-MADE (Forward)	0.55	0.58	0.58	0.56	**0.92**	**0.89**	0.25	0.23
G-MADE (Backward)	0.54	0.58	0.54	0.6	0.75	0.80	0.33	0.39
G-MADE (Mid-Frame)	0.52	0.57	0.56	0.58	0.84	0.87	0.27	0.29
G-MADE (Ensemble)	0.56	0.59	0.56	0.56	0.68	0.7	**0.44**	**0.42**

All models, except the ensemble G-MADE, display reduced accuracy for female recordings. Interestingly, the ensemble G-MADE maintains equal balanced accuracy across genders, although it exhibits a slightly lower AUC-Score for females. Among all models, the VAE consistently performs better in both male and female subsets. However, the difference in its accuracy between genders is more pronounced than in the ensemble G-MADE. Despite achieving a high True Positive Rate, the forward-ordered G-MADE tends to generate many false positives. In contrast, the Ensemble achieves the highest True Negative Rate (TNR).
This observed discrepancy in gender-based accuracy is likely due to an imbalance in the dataset composition. Of the 6.898 recorded breathing cycles, 4.485 originate from male patients, while only 2.413 are from female patients, suggesting a potential bias in data representation.

Age: We categorized recordings into three age groups: (0-7], (7-70], and (70-85]. These groups were chosen for their comparative size and representativeness, with the youngest patient being less than a year old and the oldest at 85.

Table 5.3: Performance by Age Group for VAE and G-MADE Models

Age Group	Model	ROC-AUC	BALACC	TPR	TNR
(0-7]	VAE	0.49	0.52	0.25	(0.8)
	G-MADE (Forward)	0.50	0.47	(0.48)	0.47
	G-MADE (Backward)	0.52	0.53	0.33	0.73
	G-MADE (Mid-Frame)	0.52	0.51	0.46	0.56
	G-MADE (Ensemble)	(0.62)	(0.56)	0.36	0.76
(7-70]	VAE	(0.6)	(0.58)	0.85	0.30
	G-MADE (Forward)	0.56	0.54	(0.90)	0.18
	G-MADE (Backward)	0.55	0.56	0.83	0.30
	G-MADE (Mid-Frame)	0.54	0.55	0.83	0.30
	G-MADE (Ensemble)	0.56	0.54	0.69	(0.39)
(70-85]	VAE	(0.6)	(0.59)	0.83	0.34
	G-MADE (Forward)	0.56	0.54	(0.9)	0.18
	G-MADE (Backward)	0.55	0.56	0.83	0.30
	G-MADE (Mid-Frame)	0.54	0.55	0.87	0.22
	G-MADE (Ensemble)	0.52	0.54	0.73	(0.35)

A distinct pattern emerges: almost all models struggle with recordings from the (0-7] age group. Unlike older age groups, where anomalies are detected more reliably but with higher false positives, younger patients' anomalies are often missed, while normal conditions are identified more accurately. The exception is the ensemble G-MADE, which is still able to achieve a respectable accuracy in the youngest group, outperforming all individual orderings and VAE. In the older groups, VAE is performing better than G-MADE.
The difficulty in analyzing younger patients' data could stem from their faster breathing cycles compared to older subjects. The average cycle length for patients older than seven is 2.37 seconds, while for those seven or younger, it is 1.64 seconds, potentially making anomalous sounds too short for effective detection.

5.4 Model Sensitivity to Noise

A critical measure of a model's practical applicability is its tolerance to noise. We assessed the resilience of our models against varying noise levels. Noise was generated by sampling from a normal distribution and added to the test set's signal at different Signal to Noise Ratios (SNRs) using PyTorch's `torchaudio.functional.add_noise`. SNR, measured in decibels, compares the background noise level to the original signal strength. A lower SNR indicates stronger noise: positive SNR values imply the signal is stronger than the noise, negative values indicate the noise is stronger, and 0 dB SNR signifies equal loudness for both noise and signal.
Our investigation began with an SNR of 30 dB, where the original signal was dominant, and progressively increased the noise level to 20 dB, 10 dB, and finally 0 dB. We then plotted the Balanced Accuracy of our models against these SNR values.

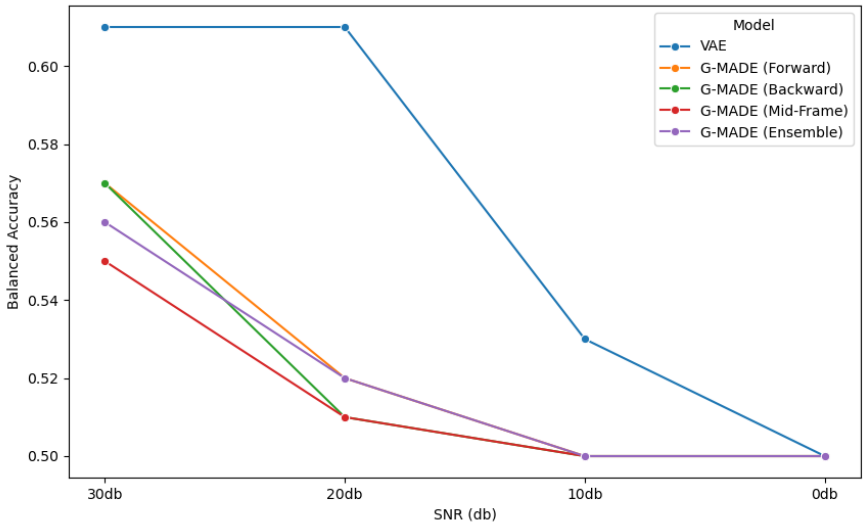

Figure 5.1: Balanced Accuracy vs SNR Noise Level for Different Models

As anticipated, the Balanced Accuracy of all models decreases with increasing noise. While we observed near-original performance at 30 dB SNR, a significant

drop is noticeable at 20dB for all G-MADE models. Interestingly, the VAE could keep the same accuracy at 20dB SNR compared to 30dB SNR and performs best at these levels of noise. However, all models fail to have meaningful separation power from 10dB SNR.
It is noteworthy that the original signal already contained substantial environmental noise. Nevertheless, the VAE seems to be more robust against noise in the sound signal.

5.5 Model Performance on Crackles and Wheezes

The ability to distinguish between crackles and wheezes is vital in diagnosing different respiratory conditions, given their distinct acoustic characteristics. To assess this, we modified the test set to evaluate performance with crackles and wheezes separately, alongside healthy sounds.

Table 5.4: Model Performance for Crackles and Wheezes

	Crackles				Wheezes			
Model	**ROC-AUC**	**BALACC**	**TPR**	**TNR**	**ROC-AUC**	**BALACC**	**TPR**	**TNR**
VAE	**0.60**	**0.60**	0.81	0.38	**0.64**	**0.61**	0.84	0.37
G-MADE (Forward)	0.55	0.57	**0.91**	0.24	0.57	0.55	0.87	0.23
G-MADE (Backward)	0.55	0.56	0.75	0.36	0.59	0.59	0.82	0.37
G-MADE (Mid-Frame)	0.53	0.56	0.84	0.29	0.58	0.57	**0.86**	0.29
G-MADE (Ensemble)	0.57	0.56	0.7	**0.43**	0.57	0.55	0.67	**0.43**

The results do not indicate a clear trend between the performance for wheezes and crackles. While the AUC-Scores for wheezes is higher than those for crackles, the balanced accuracy does not confirm this tendency. Among the G-MADE models, backward ordering performs most effectively on wheezes, whereas forward ordering is most effective on crackles. Notably, the VAE model consistently achieves the highest accuracy for wheezes and crackles. Interestingly, as seen in previous tests, the ensemble G-MADE exhibits the highest True Negative Rate (TNR) for both crackles and wheezes.
Considering the dataset distribution of 1.864 crackles to 886 wheezes, it is intriguing that the models perform similar across crackles and wheezes despite their substantially lower representation. It shows that the models are able to detect both respiratory phenomena.

5.6 Impact of Hyperparameter Variations

Optimizing hyperparameters is a crucial yet challenging task in machine learning model training. We utilized the Optuna optimization framework for our models, conducting 30 studies for each model to analyze the relative importance of various hyperparameters in achieving high balanced accuracy. A higher value indicates greater importance, suggesting that a modification in that hyperparameter would significantly impact the results.

In the case of the Variational Autoencoder (VAE), the most influential factor is the `alpha` weighting term, which balances the Mean Square Error and Kullback-Leibler Divergence in the overall loss function. The `beta1` parameter of the Adam optimizer and the dimension of the latent vector (`nz`) also play significant roles.

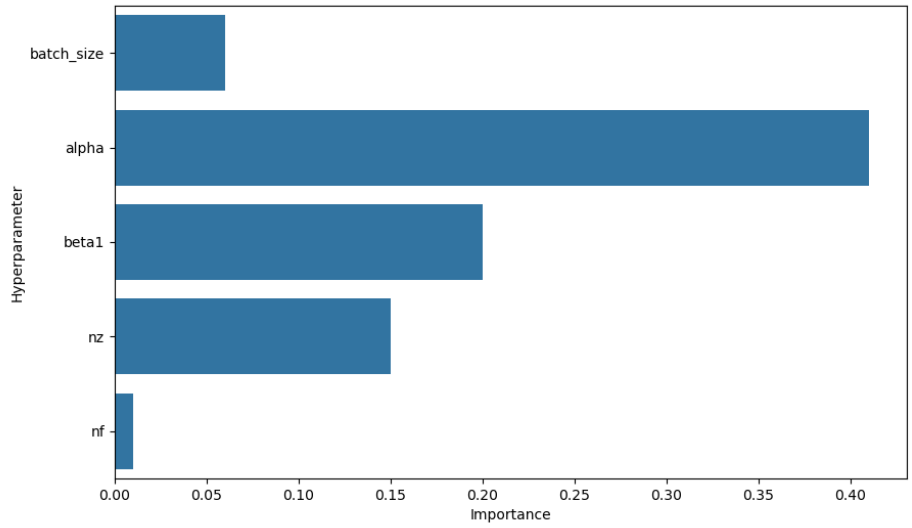

Figure 5.2: Hyperparameter performance in VAE training

Interestingly, the `nf` parameter, which affects the feature depth in the DCGAN Generator and Discriminator, has a minimal impact on overall model performance.

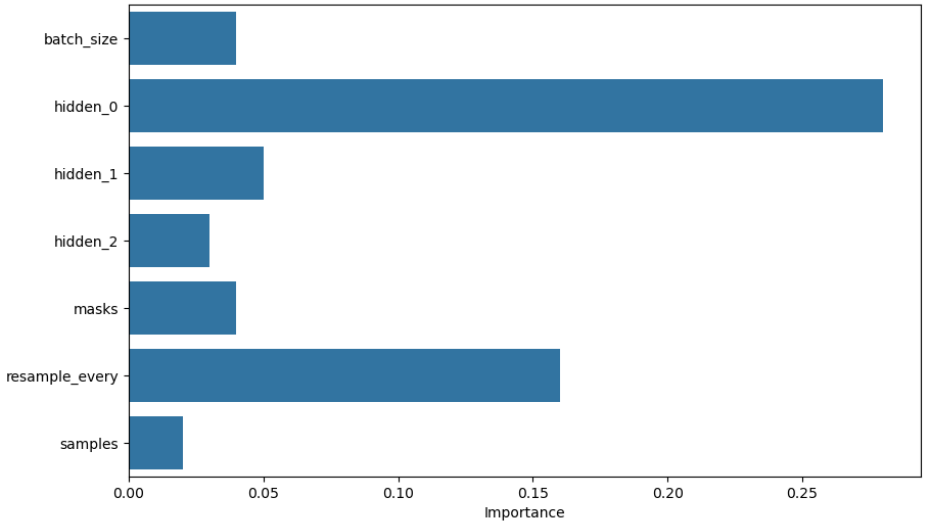

Figure 5.3: Hyperparameter performance in G-MADE (forward) training

The G-MADE model, featuring hyperparameters for its hidden layers, inherently requires more tuning than the VAE. The size of the first hidden layer is paramount in training G-MADE, followed by the frequency of mask resampling during forward passes. The subsequent hidden layers, the number of different weight masks, and the frequency of mask resampling per forward pass all show similar and relatively low importance.

5.7 Assessment of Data Splitting at Recording Level

In line with concerns mentioned in section 4.1.2, splitting data at the breathing cycle level risks overestimating the model generalization capabilities, as cycles from the same patient might appear in training and test sets. To address this, we examined the impact of our custom-designed recording level split on model performance.

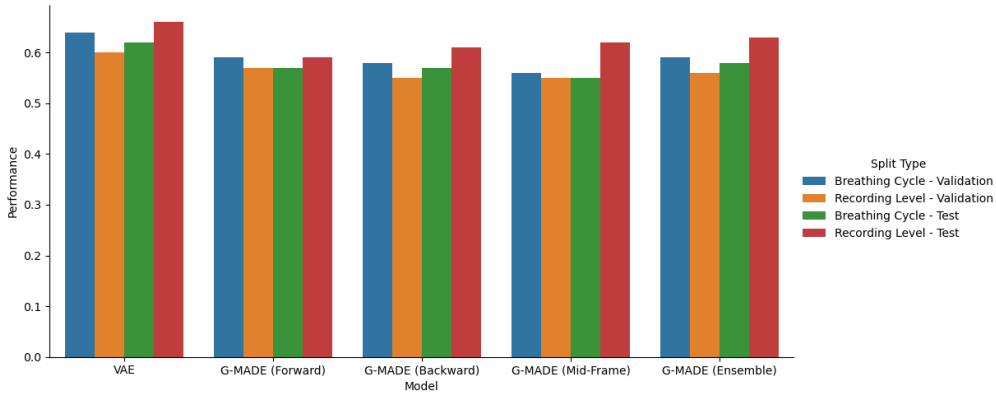

Figure 5.4: Model performance on different data split

The results demonstrate effective generalization across models for both data splits. However, an interesting trend can be observed: while the test set performance was marginally lower than the validation set for the breathing cycle level split, it improved for the recording level split. This improvement in the test set performance for the recording level split might be attributed to a fortunate data division, indicating that the split could be unintentionally biased or more representative of the model's capabilities.

5.8 Comparison of Feature Extraction Methods: MFCC vs. MelSpectrogram

MFCCs and MelSpectrograms are widespread for representing sound data in machine learning but offer different advantages. MelSpectrograms are known for better human interpretability and retaining more temporal information compared to MFCCs. To evaluate the impact of these differences, we adapted our preprocessing as described in section 4.2, replacing `torchaudio.transforms.MFCC` with `torchaudio.transforms.MelSpectrogram` and retrained all models with the new feature representation.

Our findings reveal a significant variance in model performance when using MelSpectrograms compared to MFCCs. The Variational Autoencoder (VAE) could not learn any representation of normal data points, as indicated by the accuracy close to random behavior. The forward G-MADE model maintains performance close to its MFCC counterpart but shows reduced generalization to the test set. The backward, mid-frame, and ensemble G-MADE models suffer notable performance declines, especially in test set generalization.
Incorporating dropout (with $p = 0.3$) in the Masked Autoencoder models improved their generalization capabilities. Notably, the forward, ensemble and mid-frame G-MADE models approached performances similar to those

Table 5.5: Experiment Results for VAE and G-MADE Models using MelSpectrograms

Model	Metric	Validation Set	Test Set	Test Set (Dropout)
VAE	ROC-AUC	0.46	0.44	-
	BALACC	0.51	0.5	-
	TPR	0.98	0.96	-
	TNR	0.04	0.04	-
G-MADE (Forward)	ROC-AUC	**0.57**	**0.54**	**0.55**
	BALACC	**0.56**	**0.53**	0.56
	TPR	0.36	0.34	0.88
	TNR	0.77	0.71	0.24
G-MADE (Backward)	ROC-AUC	0.39	0.39	0.4
	BALACC	0.50	0.50	0.5
	TPR	1.0	1.0	1.0
	TNR	0.0	0.0	0.0
G-MADE (Mid-Frame)	ROC-AUC	0.49	0.44	0.54
	BALACC	0.54	0.51	0.56
	TPR	0.28	0.28	0.92
	TNR	0.80	0.75	0.20
G-MADE (Ensemble)	ROC-AUC	0.5	0.46	0.53
	BALACC	0.54	0.50	**0.57**
	TPR	0.41	0.37	0.91
	TNR	0.68	0.63	0.22

achieved with MFCCs. However, the backward ordering model failed to benefit from dropout and could not learn any meaningful pattern.
These results suggest that G-MADE models possess a degree of flexibility regarding input features, effectively accommodating both MelSpectrograms and MFCCs. However, our implementation of the Variational Autoencoder did not perform as well with MelSpectrograms. This could imply that the VAE architecture is struggling with the level of detail in the MelSpectrogram representation and can not retrieve the characteristics that make up a normal breathing cycle and may be more suited to the feature representation provided by MFCCs due to their compact and efficient encapsulation of relevant sound characteristics. Further refinements in the VAE's architecture or its training process might be needed to handle better the richer temporal information presented by MelSpectrograms.

5.9 Comparative Analysis and Discussion

We now want to provide a comprehensive analysis and discussion of the experiments' findings to determine how well reconstruction-based and density-estimation-based methods are suited for anomaly detection in respiratory sounds. The Variational Autoencoder (VAE), our reconstruction-based model, showed significant effectiveness in our experiments. Its primary strength lies in its ability to learn a compact representation of normal breathing cycles, facilitating effective anomaly detection through reconstruction error

measurement. Across various datasets, including different noise levels, age groups, and genders, the VAE consistently outperformed with higher AUC-Score and Balanced Accuracy.
Despite these strengths, the VAE's performance was less effective with Mel-Spectrograms, indicating a possible limitation in processing rich temporal information. Additionally, the VAE demonstrated greater variance in accuracy between genders and age groups, suggesting a potential sensitivity to data imbalances.
G-MADE, employing density estimation, showcased its adaptability with different feature representations, maintaining relatively stable performance with both MFCCs and MelSpectrograms. Its ensemble approach notably enhanced the True Negative Rate, underlining its efficacy in minimizing false positives. However, G-MADE generally scored lower in AUC-Score and Balanced Accuracy than VAE, except in the analysis of infant and child respiratory sounds, where it performed comparably well. This observation and its compatibility with MelSpectrogram features suggest G-MADE's potential to handle nuanced temporal information. The ensemble strategy significantly contributes to the model's performance and stability.

Table 5.6: Comparison of Proposed Models with Existing Literature (80/20 Split, Test Set)

Model	**ROC-AUC**	**BALACC**	**TPR**	**TNR**
VAE (Ours)	**0.62**	**0.61**	**0.84**	0.39
VAE (Cozzatti et al.) [8]	0.61	0.60	0.58	**0.61**
G-MADE (Ensemble)	0.58	0.56	0.7	0.43

Reconstruction-based (VAE) and density-estimation-based (G-MADE) show promising respiratory sound anomaly detection capabilities and enable weakly-supervised learning in this domain. However, VAE's superior accuracy, noise robustness, and overall generalization capabilities make it a more suitable choice for this specific application. However, the decision on the model selection should also account for task-specific requirements, such as dataset characteristics and the importance of reducing false positives.

Chapter 6

Conclusion and Outlook

6.1 Summary of Findings

6.2 Potential Applications and Practical Implications

6.3 Limitations of the Proposed Approaches

6.4 Directions for Future Research

Bibliography

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631.

[2] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2, 1, 1–18.

[3] Tom Bäckström, Okko Räsänen, Abraham Zewoudie, Pablo Perez Zarazaga, Sneha Das, et al. 2020. Introduction to speech processing. (2020).

[4] Dor Bank, Noam Koenigstein, and Raja Giryes. 2023. Autoencoders. *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, 353–374.

[5] Abraham Bohadana, Gabriel Izbicki, and Steve S Kraman. 2014. Fundamentals of lung auscultation. *New England Journal of Medicine*, 370, 8, 744–751.

[6] Gaëtan Chambres, Pierre Hanna, and Myriam Desainte-Catherine. 2018. Automatic detection of patient with respiratory diseases using lung sound analysis. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE, pp. 1–6.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41, 3, 1–58.

[8] Michele Cozzatti, Federico Simonetta, and Stavros Ntalampiras. 2022. Variational autoencoders for anomaly detection in respiratory sounds. In *International Conference on Artificial Neural Networks*. Springer, pp. 333–345.

[9] DCASE. 2023. DCASE2023 Challenge - DCASE — dcase.community. https://dcase.community/challenge2023/index. [Accessed 30-11-2023]. (2023).

[10] J Earis. 1992. Lung sounds. *Thorax*, 47, 9, 671.

[11] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters*, 27, 8, 861–874.

[12] Thomas Ferkol and Dean Schraufnagel. 2014. The global burden of respiratory disease. *Annals of the American Thoracic Society*, 11, 3, 404–406.

[13] Siddhartha Gairola, Francis Tom, Nipun Kwatra, and Mohit Jain. 2021. Respirenet: A deep neural network for accurately detecting abnormal lung sounds in limited data setting. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, pp. 527–530.

[14] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. 2015. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*. PMLR, pp. 881–889.

[15] Ritwik Giri, Fangzhou Cheng, Karim Helwani, Srikanth V. Tenneti, Umut Isik, and Arvindh Krishnaswamy. 2020. Group masked autoencoder based density estimator for audio anomaly detection. In *Detection and Classification of Acoustic Scenes and Events Workshop 2020*. https://www.amazon.science/publications/group-masked-autoencoder-based-density-estimator-for-audio-anomaly-detection.

[16] Charles Miller Grinstead and James Laurie Snell. 2006. *Grinstead and Snell's introduction to probability*. Chance Project.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

[18] Nikša Jakovljević and Tatjana Lončar-Turukalo. 2018. Hidden markov model based respiratory sound classification. In *Precision Medicine Powered by pHealth and Connected Health: ICBHI 2017, Thessaloniki, Greece, 18-21 November 2017*. Springer, pp. 39–43.

[19] Andrej Karpathy. 2018. GitHub - karpathy/pytorch-made: MADE (Masked Autoencoder Density Estimation) implementation in PyTorch — github.com. https://github.com/karpathy/pytorch-made. [Accessed 05-12-2023]. (2018).

[20] Yi Ma, Xinzi Xu, Qing Yu, Yuhang Zhang, Yongfu Li, Jian Zhao, and Guoxing Wang. 2019. Lungbrn: A smart digital stethoscope for detecting respiratory disease using bi-resnet deep learning algorithm. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, pp. 1–4.

[21] Douglas O'shaughnessy. 1987. *Speech communications: Human and machine (IEEE)*. Universities press.

[22] Rajkumar Palaniappan, Kenneth Sundaraj, Nizam Uddin Ahamed, Agilan Arjunan, and Sebastian Sundaraj. 2013. Computer-based respiratory sound analysis: a systematic review. *IETE Technical Review*, 30, 3, 248–256.

[23] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54, 2, 1–38.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825–2830.

[26] Emiel Por, Maaike van Kooten, and Vanja Sarkovic. 2019. Nyquist–Shannon sampling theorem. *Leiden University*, 1, 1.

[27] Renard Xaviero Adhi Pramono, Stuart Bowyer, and Esther Rodriguez-Villegas. 2017. Automatic adventitious respiratory sound analysis: A systematic review. *PloS one*, 12, 5, e0177926.

[28] BM Rocha, Dimitris Filos, L Mendes, Ioannis Vogiatzis, Eleni Perantoni, Evangelos Kaimakamis, P Natsiavas, Ana Oliveira, C Jácome, A Marques, et al. 2018. A respiratory sound database for the development of automated classification. In *Precision Medicine Powered by pHealth and Connected Health: ICBHI 2017, Thessaloniki, Greece, 18-21 November 2017*. Springer, pp. 33–37.

[29] Gorkem Serbes, Sezer Ulukaya, and Yasemin P Kahya. 2018. An automated lung sound preprocessing and classification system based on-spectral analysis methods. In *Precision Medicine Powered by pHealth and Connected Health: ICBHI 2017, Thessaloniki, Greece, 18-21 November 2017*. Springer, pp. 45–49.

[30] Runze Wang, Yanan Guo, Wendao Wang, and Yide Ma. 2019. Bi-ResNet: fully automated classification of unregistered contralateral mammograms. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings 28*. Springer, pp. 273–283.