JOHANNES KEPLER
UNIVERSITY LINZ

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

Author
Lukas Selch
11941656

Submission
**Institute of
Computational Perception**

Thesis Supervisor
Paul Primus

November 2023

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 [5]. 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 [2]. 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, comprising the airways and lungs, is responsible for the vital function of gas exchange in the human body. Respiratory sounds are generated by the airflow within this system during the inhalation and exhalation [4]. Because these sounds are known to be of great importance in the detection of respiratory pathology, listening to the sounds of breathing through the chest using a stethoscope, called *auscultation*, is a cost-effective, non intrusive, and common part of the physical examination [2].

The characteristics of the sounds observed during chest auscultation can be precisely defined, allowing for a clear distinction between normal and pathological sounds. Normal breathing sounds are heard throughout the inhalation phase, but only at the very beginning of the exhalation phase, and have a relatively narrow frequency band from 100 Hz to 1000 Hz. While there are a variety of different abnormal breathing sounds, we will focus on two of the most prominent and easily recognizable features.

First, *Wheezes* are musical sounds of long duration over 100 milliseconds and of sinusoidal oscillations that can occur during expiration and inspiration caused by airway narrowing or restriction. They range from 100 Hz to 1000 Hz, with higher harmonics possible above that.

The second sound of relevance are *Crackles*. These non-musical sounds are brief and indicate occasional airway opening, possibly caused by secretions. They can be further subdivided into Fine Crackles and Coarse Crackles, which differ in frequency and duration. While Fine Crackles have a characteristic frequency of about 650 Hz and last about 5 milliseconds, Coarse Crackles are longer sounds of more than 15 milliseconds and occur at lower frequencies below 350 Hz [2].

Knowing what constitutes physiological and pathological pulmonary sounds, it is possible to pave the way for automated systems to detect them. Electronic stethoscopes can convert the sound signals from the lung to digital signals, allowing the utilization of advanced anomaly detection algorithms for computer-aided medical diagnosis.

2.2 Fundamentals of Anomaly Detection

Anomaly detection addresses the task of identifying deviations in data from anticipated patterns, commonly termed as anomalies or outliers [3]. These anomalies often signal deviations of a system from the norm that are potentially critical and require intervention by the system user. In healthcare, as discussed in section 2.1, outliers in patient respiratory sound patterns can indicate certain conditions.

To find these outliers, anomaly detection algorithms typically define a certain concept of what is considered normal, and any data point that deviates is flagged as an anomaly. The major challenge here is that anomalies are rare and usually not available on a large scale, resulting in a huge data imbalance. In addition, the distribution of anomalies remains uncertain, and even within the set of outliers there may be substantial variation, as data can exhibit various forms of non-normality [8]. In addition, it is important to strike a balance between false positives and false negatives based on the specific application domain, as the severity of one over the other can vary significantly. Outlier detection systems must be properly calibrated to match the characteristics of the domain.
Anomaly detection includes different modes depending on the available data. The simplest, but less common in practical scenarios, is *supervised anomaly detection*. In this approach, a fully labeled dataset is used and the task of the detection system is to determine the boundary between normal and abnormal data points. However, it's important to note that real-world datasets often lack comprehensive coverage of different outlier types, making *unsupervised anomaly detection* more prevalent. In unsupervised anomaly detection, only data points known to be normal are provided, and the system must autonomously learn their defining characteristics. Between these two extremes, there are intermediate modes. In our research, we will address the challenge of *weakly supervised* data, where primarily normal data points are available, but a subset of anomalies is also included to evaluate the real-world effectiveness of the learned representation within a detection system.
Outlier detection algorithms typically output a label that directly classifies the data point as normal or anomalous, or they output an anomaly score, which is a measure of the degree of deviation from normality. This score can then be used to define a threshold above which samples are considered anomalous. In addition to traditional anomaly detection methods such as k-nearest neighbor (KNN) or support vector machines (SVMs), which rely on distance metrics or decision boundaries in the feature space to identify outliers, the application of deep learning methods to anomaly detection has gained traction recently. Traditional methods such as KNN work by identifying the closest data points in a data set and flagging those that are significantly farther away as anomalies. SVMs, on the other hand, focus more on defining a boundary between classes and are particularly effective in scenarios where the separation is clear and well-defined [3]. However, while these methods have proven to be highly effective in many applications, they can be limited in a setting with complex, high-dimensional and noisy data, or data where the anomalies are very similar to the normal.
Deep learning methods, on the other hand, have the ability to learn and extract features from raw input data and have shown a remarkable ability to effectively learn patterns in complex data structures such as audio signals. In the following, we will delve into two distinct deep learning architecture families that will be employed in this research.

2.3 Reconstruction-Based Methods

In the context of anomaly detection, reconstruction-based methods use reconstruction errors as anomaly scores. The process unfolds in two main steps. First, the method reduces the dimensionality of the original data by transforming it into a latent, more compact representation. Ideally, this *latent space* captures the essential features of the data while dropping noise and unimportant information. Then follows the actual reconstruction where an attempt is made to receive back the original data from the compact representation. The chal-

lenge is to find a dimensionally reduced representation that is as compact as possible, while retaining enough essential information from the original data to allow accurate reconstruction and avoid overfitting.
During the training process, the input to a reconstruction-based model contains only data points that are known to be normal, making the process unsupervised. This is critical for the model to learn the typical patterns of standard data. Throughout training, a *reconstruction error* is calculated, which is a measure of how well the reconstructed data matches the original data. In other words, this error evaluates how accurately the model can recreate the input data. The basic assumption is that a model trained exclusively on normal data will be able to reconstruct the original data with minimal error. Consequently, when the trained model encounters anomalous data that deviates from the normal patterns, it will struggle to reconstruct it, resulting in a higher reconstruction error. The magnitude of the reconstruction error can be used as an anomaly score, where a higher error indicates a higher probability that a data point is anomalous and vice versa, allowing the model to be used as an anomaly detection system.

2.3.1 Essentials of Autoencoders

Autoencoders are one way to address the dimensionality reduction and are a popular choice of neural network architecture for reconstruction-based methods, as they are capable of efficiently learn a compact representation of data in an unsupervised fashion. An autoencoder consists of two key components: the Encoder (denoted as $A : \mathbb{R}^n \to \mathbb{R}^l$) and the Decoder (denoted as $B : \mathbb{R}^l \to \mathbb{R}^n$).
The Encoder is responsible for mapping high-dimensional input data of dimensionality n into a lower-dimensional latent space of dimensionality $l < n$. This represents the compression step into a efficient, compact representation of the original data. The Decoder performs the inverse operation where it takes the data from the latent space and reconstructs it back to its original dimensionality, aiming to reproduce the original input as accurately as possible. The result is an optimization problem defined as

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

where E is the expected value of the reconstruction loss Δ for input $\mathbf{x}$ [1].

2.4 Density Estimation Methods

Probability distributions are mathematical functions that describe the probability associated with each possible value of a random variable. If the random variable can take any value within a certain range, the probability distribution is continuous. It is typically described using a *probability density function* (PDF). The PDF is a fundamental concept in probability and statistics, providing a way to calculate the probability of the random variable falling within a specific interval.
For a continuous real-valued random variable X, the PDF is defined as

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

for all $a, b \in \mathbb{R}$ [7]. Here, $f(x)$ represents the probability density function of X. It's important to note that the PDF itself does not give probabilities directly.

Instead, the probability of X falling within the interval from a to b is given by the area under the curve of the PDF between these two points.

Density estimation is concerned with finding an estimate of the PDF from observed data by estimating a joint distribution $p(\mathbf{x})$ from a set of examples $\{\mathbf{x}^{(t)}\}_{t=1}^{T}$ [6], which helps to understand the distribution of data points within a data set. It can be *parametric* or *nonparametric*. Parametric density estimation assumes that the data follow a certain distribution. The task is then to estimate the parameters of that distribution. For example, in the case of a normal distribution, the mean and variance are to be estimated. Nonparametric density estimation, on the other hand, does not assume a specific distribution for the data and instead attempts to estimate the PDF directly from the data. This allows for more complex distributions, but typically requires more data to produce an accurate estimate.

Density estimation is well suited for anomaly detection tasks. By estimating the PDF of a dataset, it is possible to identify low-probability regions. Data points that fall into these regions can be considered anomalies, making it a straightforward task once the PDF is properly captured.

2.4.1 Introduction to Masked Autoencoders

Masked Autoencoders for Distribution Estimation (MADE) aims to modify autoencoders so that they don't just learn to compress and reconstruct data, but also to understand how the data is distributed. The notion of autoregressive property is essential. While traditional autoencoders do not consider the order of the input data, MADE introduces a mechanism where each part of its output is determined only by the parts of the input that precede it.

To ensure this ordering, MADE masks the weights of each autoencoder layer to control the information flow between neurons of two layers. Each neuron is thereby given a label, a number between 1 and $D-1$, where D is the dimensionality of the input. The masks are then built using this general rule: 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. This can be mathematically expressed 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. To strictly maintain the autoregressive property, the rule for the output layer must be slightly modified. Here, a neuron in the last hidden layer can only influence an output neuron if its label is strictly smaller [6].

In summary, MADE provides a solution for controlling the flow of information in an autoencoder so that each part of the output depends only on the preceding parts of the input, providing a probabilistic model that can be further used for anomaly detection.

2.5 Evaluation Metrics for Model Comparison

To effectively evaluate and compare the performance of different anomaly detection models, the selection of evaluation metrics is critical. These metrics must not only be relevant to real-world scenarios, but must also provide clear and insightful interpretations that allow for an intuitive understanding of

the model's capabilities and effectiveness. In this work, five different metrics are specifically considered, for which the understanding of the concept of the confusion matrix needs to be further explained.
In anomaly detection, a model can attribute a datapoint to be normal or an anomaly. A prediction can have four different outcomes. When the model identifies a data point as normal and it truly is normal, this correct identification is known as a *true negative*. On the other hand, if the model labels a data point as normal but it is actually an anomaly, this incorrect identification is termed a *false negative*. Conversely, when the model predicts a data point as an anomaly and it indeed is an anomaly, we call this a *true positive*. However, if the model mistakenly predicts a normal data point as an anomaly, this error is referred to as a *false positive*. All four outcomes can be summarized in a confusion matrix.

Having introduces this terminology, we focus on defining the used evaluation metrices.

2.5.1 Recall

Recall is the True Positive Rate (TPR), which can be expressed mathematically as

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

It is the proportion of anomalies correctly identified by the model. It's values range from 0 (not a single anomaly correctly detected) to 1 (perfect anomaly detection), with the goal being to maximize this metric. Especially in medical applications, where missing an anomaly can have serious implications for a patient, this value is desired to be high.

2.5.2 False Positive Rate (FPR)

The FPR is a measure of the proportion of incorrect positive predictions relative to the total number of actual negatives. It is expressed mathematically as

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

It focuses on which normal data points are misclassified as anomalies, ranging from 0 (not a single normal sample is labeled as an anomaly) to 1 (every normal data point is labeled as an anomaly). Lower FPR values are preferred because they indicate a lower rate of false alarms.

2.5.3 Area Under the Curve (AUC)

To evaluate the performance of a model, the trade-off between TPR (benefit) and FPR (cost) is of primary interest and can be plotted on a Receiver Operating Characteristic (ROC) graph. This graph, structured as a unit square, uses the X-axis to plot the FPR and the Y-axis to plot the TPR.

On the unit square, point (0,0) represents a model that predicts all data to be normal, while point (1,1) represents a model that predicts all data to be anomalous. The theoretical goal is to achieve perfect classification, represented by the

point (1,0).
The ROC curve is obtained by varying the classification threshold of a model, calculating the TPR and FPR at each threshold, plotting a point on the ROC graph, and then connecting each point with a line. Each point thereby reflects a different trade-off between TPR and FPR.

For a binary classification task such as anomaly detection, a model with random performance represents a diagonal line from (0,0) to (1,1). Any curve above this diagonal is considered better than random, and any curve below is considered worse than random.
To obtain a single value that accurately encodes the insight gained form the ROC, the Area Under the Curve (AUC) is considered. It is the area enclosed by the ROC curve and the x-axis on the unit square and ranges from 0.5 (no discriminative ability of the model) to 1 (perfect classification). It is important to note that the output of models with an AUC below 0.5 can be inverted to obtain better than random classification.
To illustrate the relevance of AUC as a performance measurement, let's consider an example from the field of medicine. In medical diagnostics, a high AUC value is crucial as it indicates the model's ability to accurately distinguish between patients with a specific disease and those who are healthy. This accuracy is vital for ensuring that patients who require treatment are correctly identified and treated, while at the same time avoiding unnecessary medical interventions in healthy individuals. Such a balance is essential in medical practice to provide effective care and minimize harm.

2.5.4 Specificity

Specificity, also known as the True Negative Rate (TNR), is mathematically expressed as

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

This metric quantifies the proportion of normal data points correctly identified by the model. It's value ranges from 0 (every normal data point is falsely flagged as an anomaly) to 1 (perfect recognition of normal samples), with the goal of achieving a high TNR. In medical diagnostics, maintaining high specificity is essential to avoid false alarms. Together with the Recall, it can be used to calculate the Balanced Accuracy.

2.5.5 Balanced Accuracy (BALACC)

Balanced Accuracy is a performance metric that equally weights the importance of correctly identifying anomalies and normal data points:

$$BALACC = 0.5 \times (Recall + Specificity)$$

TPR and TNR are combined in this metric to provide a more holistic view of model performance, especially in cases where datasets are unbalanced. In a highly unbalanced dataset where one class significantly outnumbers the other, which is often the case in anomaly detection since the anomaly typically occurs less frequently than normal behavior, traditional accuracy can be misleading. For example, in a data set with 95% normal data points and only 5% outliers, a model that always predicts the 'normal' class would achieve 95% accuracy. By averaging both specificity and recall, the balanced accuracy in this case would be 50%, indicating random performance. This metric has similar goals to AUC, emphasizing the importance of both reliably detecting anomalies and avoiding false alarms. This metric will be used later for better comparison with existing literature.

Chapter 3

Literature Review

3.1 Anomaly Detection Techniques in Respiratory Sound Analysis

3.1.1 The Respiratory Sound Database

3.1.2 Existing Approaches

3.2 Variational Autoencoders

3.3 Group Masked Autoencoders

Chapter 4

Methodology

4.1 Dataset

4.1.1 Definition

4.1.2 Data Splitting

4.2 Preprocessing

4.3 Detailed Overview of the Models

4.3.1 Implementation of the VAE

4.3.2 Implementation of the G-MADE

Chapter 5

Experiments and Results

5.1 Experimental Setup

5.2 Generalization Capabilities

5.3 Age and Gender Differences in Model Accuracy

5.4 Model Sensitivity to Noise

5.5 Model Performance on Crackles and Wheezes

5.6 Impact of Hyperparameter Variations

5.7 Assessment of Data Splitting at Recording Level

5.8 Comparison of Feature Extraction Methods: MFCC vs. MelSpectrogram

5.9 Interpretability and Explainability of the Proposed Models

5.10 Comparative Analysis and Discussion

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

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

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

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

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

[6] 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.

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

[8] 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.