

ARI2201 - IAPT
Individual Assigned Practical Task
Classification of Lung Respiratory Sounds



**L-Università
ta' Malta**

Name: **Andrea Filiberto Lucas**
ID No: 0279704L
Course: ARI2201 - Artificial Intelligence (AI)
Supervisor: Dr Kristian Guillaumier

1 Abstract

Lung auscultation is an important diagnostic technique in modern medicine, allowing doctors to assess respiratory health by listening to internal bodily sounds. This study investigates a combination of advanced machine learning (ML) techniques and lung auscultation, with the goal of improving diagnostic accuracy and, ultimately, the treatment of patients. Using a comprehensive Kaggle dataset containing diverse recordings spanning various respiratory conditions, this study attempts to uncover the acoustic signatures unique to pulmonary disorders and build robust classification models.

Respiratory sound data was analysed thoroughly using four different machine learning algorithms: convolutional neural networks (CNNs), artificial neural networks (ANNs), support vector machines (SVMs), and random forests (RFs). Each algorithm was chosen for its distinct ability: CNNs for handling structured grid-like data, ANNs for detecting intricate patterns, SVMs for precise binary classification, and RFs for robust collective learning. The study included several phases, including meticulous data pre-processing, feature extraction using Mel-frequency cepstral coefficients (MFCCs), and rigorous model training and evaluation, all carried out using a combination of Python and R programming languages.

The findings highlight a significant step forward in improving respiratory condition diagnosis. The ANN had the best performance, with an accuracy of 92% after 70 epochs and a batch size of 5. In comparison, the CNN achieved 90% accuracy while using a larger number of epochs (74) and batch size (92). Both models correctly classified lung auscultations, distinguishing a number of diseases from one another as well as healthy recordings. Furthermore, the Random Forest (RF) model had an accuracy of 74.28% in determining whether a recording was healthy or unhealthy. The lowest-performing model achieved an accuracy of 69.67% while also classifying recordings as healthy or unhealthy.

This study highlights the potential of ML to standardise interpretations and reduce diagnostic variability in clinical practice, ultimately contributing to the advancement of precision medicine in respiratory healthcare.

2 Introduction

Lung auscultation is an essential technique in modern medicine that provides valuable insights into respiratory health by carefully listening to internal bodily sounds. This practice is critical in medical diagnosis because it allows professionals to make appropriate choices about patient care. However, as technology advances, there is an increasing opportunity to integrate sophisticated machine learning (ML) techniques into this long-standing tradition, ushering in a new era of precision medicine. In addition to manual stethoscope-based auscultation, computer-aided auscultation systems are now being considered to improve diagnostic capabilities through automation and data analysis.

The human lung generates various sounds based on its physiological state. These sounds can help diagnose a variety of pulmonary conditions. However, deciphering and categorising these sounds remains challenging. To diagnose common ailments like asthma and respiratory tract infections, it's important to thoroughly examine respiratory sounds and identify distinguishing characteristics.

The incorporation of machine learning, signal processing, and data analytics into modern medicine presents a potentially successful path for improving the effectiveness of lung auscultation. Clinicians can use these cutting-edge technologies to improve diagnostic accuracy, streamline clinical workflows, and, ultimately, optimise patient outcomes. However, incorporating ML techniques into traditional auscultation presents academic challenges because it necessitates a thorough understanding of signal processing techniques, algorithmic frameworks, and domain-specific data.

Thus, the current study seeks to address the multifaceted aspects of lung auscultation, elucidating its significance in modern healthcare and outlining the academic challenges associated with using machine learning to improve diagnostic capabilities. This study will use a comprehensive collection of annotated respiratory sounds available on Kaggle¹. This dataset contains recordings from patients with a variety of respiratory conditions, providing a wide range of data for analysis. Thus, the research seeks to gain valuable insights into the acoustic signatures of pulmonary disorders, which will help create more reliable classification models.

Four machine learning algorithms will be used to compare respiratory sound data: Convolutional Neural Networks (CNNs), Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and Random Forests (RFs). ANNs are computational models based on the structure and functionality of the human brain. They have the remarkable ability to detect intricate patterns and relationships within datasets. SVMs, on the other hand, are an effective tool in supervised learning. They help to separate data points into distinct classes by optimising the margin between them in a high-dimensional space. Furthermore, CNNs are a deep learning (DL) architecture capable of handling structured grid-like data, such as images or spectrograms. CNNs use convolutional layers to extract hierarchical features, allowing for efficient processing. Finally, RFs provide a robust ensemble learning technique by combining multiple decision trees to improve predictive accuracy and reduce overfitting.

The document begins with a comprehensive explanation of medical terms related to lung auscultation and their significance in clinical practice. Next a thorough literature review will be conducted on machine learning techniques and their outcomes in previous studies. This document begins with a detailed explanation of medical terms related to lung auscultation and their clinical relevance. Following that, a thorough review of the literature will be conducted, with a focus on ML methodologies and previous research findings. This analysis is critical for developing an academic comparison, as it guides methodology selection and evaluation criteria. The implementation methodology will be outlined, covering data preprocessing, feature extraction, and model training. Finally, the results will be carefully evaluated to determine the effectiveness and suitability of the proposed approach.

¹<https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database>

3 Background

This chapter delves into lung auscultation, an important diagnostic technique in clinical practice for assessing respiratory function. It discusses normal and abnormal breath sounds, their importance in detecting respiratory conditions, and the limitations of traditional auscultation techniques. Furthermore, it discusses the use of ML algorithms to improve diagnostic accuracy and standardise interpretation.

3.1 Understanding Lung Auscultation

Lung auscultation is a fundamental diagnostic technique in clinical practice, allowing healthcare professionals to assess the respiratory system's function. This vital procedure entails carefully listening to internal sounds produced while breathing, which allows clinicians to gain valuable insights into the health and functionality of the lungs and airways. Clinicians can use a stethoscope to directly assess the sounds produced within the respiratory tract, allowing for real-time evaluation of pulmonary dynamics and the detection of any abnormalities that may indicate an underlying disease.

The value of lung auscultation derives from its ability to provide detailed information on various aspects of respiratory function. Air movement within the respiratory tract is an important consideration when performing auscultation. Normal, healthy, sounds of breathing, also known as vesicular breath sounds, consist of a smooth, continuous flow of air through the airways during both inhalation and exhalation. Changes in the intensity, duration, or quality of these breath sounds can indicate changes in airflow dynamics, such as airway obstruction or restriction, which could be indicative of asthma, bronchitis, pneumonia, amongst other conditions.

Furthermore, lung auscultation allows clinicians to assess lung tissue integrity and the presence of abnormal sounds that may indicate an underlying pathology. Abnormal respiratory sounds, or adventitious sounds, include wheezes, crackles (also known as rales), rhonchi, and stridor, amongst others. Each of these sounds have distinct characteristics and can aid in diagnostics. Wheezes, for example, are high-pitched melodic sounds produced during expiration that indicate narrowed airways, which are common in asthma and chronic obstructive pulmonary disease (COPD). Crackles, on the other hand, are brief, discontinuous sounds heard during inspiration that can indicate fluid-filled airways or alveolar collapse, both of which are frequently associated with conditions such as pneumonia and pulmonary fibrosis.

Lung auscultation also helps professionals to assess various lung regions, including the trachea, bronchi, and alveoli, in order to obtain a complete assessment of respiratory function. The trachea, located anteriorly in the neck, provides information about central airway dynamics as well as potential obstructions like foreign bodies or tumours. Airflow in the larger airways can be measured by auscultating the bronchi, which branch off the trachea and extend into the lungs. Finally, auscultation of the alveoli, the lungs' smallest functional units where gas exchange occurs, reveals the integrity of the distal airspaces and the presence of abnormalities such as consolidation or atelectasis.

The dataset's annotated diseases include asthma, bronchiectasis, bronchiolitis, chronic obstructive pulmonary disease (COPD), lower respiratory tract infection (LRTI), pneumonia, and upper respiratory tract infection (URTI). Asthma is characterised by airway inflammation and narrowing, which causes symptoms such as wheezing, coughing, and shortness of breath. Bronchiectasis is characterised by bronchial tube widening and scarring, which results in recurring infections and a persistent cough with excess mucus. Bronchiolitis is an acute viral infection that affects the small airways of infants, causing coughing, wheezing, and breathing difficulties. COPD is a progressive lung disease that includes chronic bronchitis and emphysema, which cause restricted airflow and difficulty breathing.

LRTI includes infections of the airways and lung parenchyma, such as bronchitis, pneumonia, and bronchiolitis. Pneumonia causes inflammation of the lungs' air sacs, resulting in symptoms such as coughing, fever, and breathing difficulty. URTI is a common upper airway infection that causes symptoms such as a sore throat and runny nose.

3.2 Challenges in Traditional Auscultation

Lung auscultation faces several challenges that limit its precision. The subjective nature of interpretation, as well as clinical variability, all contribute to diagnostic disparities. Despite the need for extensive training to address these challenges, such endeavours require a significant amount of resources. Furthermore, distinguishing subtle differences in sound presents a significant challenge, potentially leading to misdiagnoses or treatment delays. As a result, there is an increasing demand for objective analytical methodologies. Promising technological advancements, such as electronic stethoscopes and artificial intelligence (AI) algorithms, provide opportunities for standardising interpretation and improving diagnostic accuracy. These innovations have the potential to significantly improve patient care.

3.3 Application of Machine Learning in Sound Classification

Recently, the healthcare industry has seen a significant increase in the use of machine learning (ML) algorithms for various diagnostic applications. One area of particular interest is the classification of respiratory sounds, where ML techniques have great potential. Traditional lung auscultation relies heavily on clinicians' subjective assessments, which can lead to inconsistencies and variations in diagnoses. These limitations can be mitigated by using machine learning techniques, allowing for a more objective and automated approach to respiratory sound classification. ML algorithms can analyse large amounts of sound data, identify recurring patterns, and make accurate predictions, transforming the diagnostic process in respiratory medicine.

3.4 Recording Equipment & Acquisition Modes in Kaggle Dataset

The Kaggle dataset used in this study employs various acquisition modes and specialised recording equipment to accurately capture respiratory sounds. Sequential/single-channel (sc) and simultaneous /multi-channel (mc) modes have distinct advantages, including temporal resolution and spatial coverage. The recording equipment includes the AKG C417L Microphone, renowned for its sensitivity and fidelity in capturing subtle acoustic nuances, the 3M Littmann Classic II SE Stethoscope, a standard tool in clinical auscultation known for its reliability and durability, the 3M Littmann 3200 Electronic Stethoscope, featuring advanced digital amplification and noise reduction capabilities, and the WelchAllyn Meditron Master Elite Electronic Stethoscope, renowned for its precision.

3.5 Key Audio Features for CNNs and ANNs

Mel-frequency cepstral coefficients (MFCCs) are representations of a sound's short-term power spectrum that highlight perceptually relevant features by simulating the human ear's response. Chromagrams measure the intensity of pitches within a musical octave, providing information about the harmonic content. Mel-scaled spectrograms show the power spectrum on a Mel scale, which corresponds to human pitch perception and emphasises tonal elements. Spectral contrast is the difference in amplitude between spectral peaks and valleys, which reflects the timbral texture. Tonal centroid features (tonnetz) map harmonic relationships between pitches, creating a geometric representation of the audio's tonal properties. These features are essential in CNNs and ANNs as they convert raw audio signals into structured numerical data that the models can handle. They capture essential sound characteristics, allowing networks to learn and recognise patterns, ultimately improving performance in tasks like audio classification, speech recognition, and music analysis.

3.6 Understanding the different types of ML Methodologies

This subsection (3.6) of the paper provides a concise overview of fundamental ML algorithms, specifically Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Random Forests (RFs). These techniques play a pivotal role in the methodology of the project. For those seeking a more in-depth understanding, the referenced papers [1-8] provide additional insights. It is presumed that readers possess a foundational knowledge of these algorithms as they proceed through the paper.

3.6.1 Convolutional Neural Networks (CNNS)

CNNS are specialised deep learning models designed to handle structured grid-like data, such as images and audio spectrograms [1]. CNNs, which consist of several layers such as convolutional, pooling, and fully connected layers, are capable of extracting hierarchical features from spectrograms or other sound data representations. This capability allows CNNs to detect complex patterns and variations associated with various respiratory conditions [2]. Using CNNs on large datasets containing annotated respiratory sounds, models can be built with exceptional accuracy in classifying a wide range of respiratory sound patterns [1-2].

3.6.2 Artificial Neural Networks (ANNs)

ANNs are computational models inspired by the structural and functional complexity of the human brain. These networks are made up of interconnected nodes, or neurons, organised in layers to mimic the neural architecture of the brain [3]. ANNs, as machine learning algorithms, have the ability to detect intricate patterns in data through a training process. In the context of respiratory sound classification, ANNs are trained using labelled sound samples to identify and categorise various respiratory sounds, including wheezes, crackles, and normal breath sounds [3-4]. During the training process, the weights and biases of neural connections are adjusted to optimise performance in accurately categorising respiratory sounds [3-4].

3.6.3 Support Vector Machines (SVMs)

SVMs are a type of supervised learning model known for its success in binary classification tasks. SVMs work by determining the most optimal hyperplane that separates data points into distinct classes in a high-dimensional feature space [5]. SVMs excel at distinguishing between normal and abnormal respiratory sounds by detecting distinct patterns and features associated with each category [6]. SVMs are well-suited to the nuanced characteristics of respiratory sound analysis because of their ability to address intricate and non-linear associations within datasets [6]. SVMs enable robust categorization of respiratory sounds, allowing for timely identification and diagnosis of pulmonary disorders [5-6].

3.6.4 Random Forests (RFs)

RFs are an ensemble learning technique that can be applied to both classification and regression tasks [7]. They work by constructing a large number of decision trees during the training phase. Each tree in the forest is trained independently on a subset of the data, and the final prediction is calculated by aggregating the predictions of all individual trees [7]. RFs are well-known for their ability to handle high-dimensional data with complex interactions while remaining robust to overfitting. Random Forests can analyse a wide range of features extracted from sound recordings, allowing for accurate identification and categorization of various respiratory conditions such as wheezes, crackles, and normal breath sounds [7-8]. Random Forests' ensemble nature makes them particularly well-suited to tasks requiring both interpretability and performance [7-8].

4 Literature Review

This literature review investigates new approaches to respiratory sound classification in order to improve diagnostic methodologies in respiratory medicine. It covers the integration of ANNs and FNNs, SVM-based classification of VAR model parameters, and the importance of standardised protocols in ML models. It also emphasises the importance of data augmentation and the use of features such as MFCC in CNN architectures to improve classification accuracy. Finally, it demonstrates the robustness of machine learning classifiers, specifically random forests, in accurately classifying pulmonary diseases, indicating their potential as reliable tools in healthcare decision-making.

4.1 Alternative Respiratory Sounds Classification System Utilizing ANNs

In their thorough investigation, Oweis et al. [9] delves into the complexities of respiratory sound classification, shedding light on a novel combination of ANNs and fuzzy neural networks (FNNs), known as neuro-fuzzy systems. Their innovative methodology aims to overcome the challenges inherent in traditional auscultation methods, particularly subjective interpretation and inter-clinician variability.

The methodology section of the study describes in detail the architectures and training procedures for both ANN and FNN models. The ANN model is rigorously trained on a meticulously labelled dataset of respiratory sounds, with parameters adjusted iteratively to maximise performance. Similarly, the FNN model, also known as the Adaptive Neuro-Fuzzy Inference System (ANFIS), uses neural network learning characteristics to extract fuzzy system parameters. Impressively, this meticulous configuration produces a remarkable accuracy rate of 98.6%[9].

Furthermore, the authors in [9] highlight the importance of hyperparameters in developing the ANFIS model. The study demonstrates the superior precision and reduced misclassification instances exhibited by both ANN and FNN-based systems through rigorous evaluation and comparison with conventional methodologies.

Overall, the study highlights the transformative potential of ANN and FNN-based approaches in respiratory sound classification, indicating a paradigm shift in diagnostic methodologies in respiratory medicine. The implications for clinical practice are significant, with improved patient care outcomes and streamlined diagnostic processes in respiratory healthcare.

4.2 SVM Based Classification of VAR Model Parameters

The authors in [10] make a significant advancement in computerised diagnosis in respiratory medicine by using SVMs to classify vector autoregressive (VAR) model parameters derived from respiratory sounds. Their approach aims to improve diagnostic accuracy and streamline classification processes, representing a significant advancement in respiratory disorder diagnosis.

The study [10] thoroughly examines the extraction of VAR model parameters from respiratory sound data, followed by SVM-based classification. The SVM detects intricate patterns and classifies respiratory sounds with ease after training on meticulously labelled data. The results demonstrate the method's efficacy, with high sensitivity and specificity rates, particularly in certain classifications.

Despite the promising results, the study candidly acknowledges areas for improvement, particularly in distinguishing between bronchiectasis and healthy cases. To address this, the authors propose ways to improve, such as investigating alternative mathematical models, incorporating new features, and testing different classifiers. These recommendations aim to improve diagnostic performance and expand the use of computational techniques in clinical practice.

4.3 The Need for More Research

Garcia-Mendez et al. [11] conduct a comprehensive review of the characteristics, diagnostic accuracy, concerns, and data sources of ML models used for lung sound classification. Their comprehensive analysis reveals the widespread use of ANNs and SVMs in this domain, with accuracy rates ranging from 49.43% to 98% for abnormal sound types and 69.40% to 99.62% for disease classification.

Despite the promising results observed in various studies, the review reveals a widespread high risk of bias, particularly in patient selection and reference standards. Garcia-Mendez et al. [11] emphasise the importance of standardising recording and labelling procedures in public databases, arguing for greater reliability and reproducibility in this field of research. By addressing these concerns, the study hopes to increase trust and confidence in the findings of ML models in lung sound classification.

Furthermore, the study [11] highlights the enormous potential of ML models to accurately classify lung sounds using publicly available data. However, it identifies significant areas for improvement, particularly in terms of bias reduction and standardisation protocols. Addressing these challenges will allow the field to advance and refine ML-based approaches for diagnosing and classifying respiratory conditions.

4.4 The importance of Data Augmentation

In the academic paper [12], an in-depth investigation into refining CNN models was conducted, introducing novel methodologies that highlighted the importance of data augmentation in improving model robustness and generalisation in respiratory sound classification.

Data augmentation, a fundamental machine learning technique, involves artificially diversifying the training dataset by making changes to existing data samples [12]. In the context of respiratory sound classification, this augmentation goes beyond traditional methods to incorporate nuanced changes that reflect real-world scenarios such as time and pitch shifting. These techniques simulate variations in respiratory patterns, enriching the dataset to protect the model from overfitting and increasing adaptability to various respiratory anomalies.

Furthermore, the study investigated the complementary integration of SVMs and CNNs in the classification framework. SVMs provide a robust framework for discerning patterns in respiratory sound data, based on the structural risk minimization principle, which is critical in scenarios with nonlinear separability [12]. By combining SVMs' discerning classification capabilities with CNNs' expertise in feature extraction, an integrated strategy emerges that promises to outperform traditional classification methodologies.

Shivakumar's presentation [12] emphasises this interdisciplinary effort as a transformative step forward in respiratory sound classification paradigms, potentially expanding clinicians' diagnostic arsenal and improving patient care through data-driven insights. The study emphasises the profound implications of machine learning in augmenting clinical decision-making processes, ultimately improving healthcare delivery and patient outcomes, by elucidating the interplay between data augmentation, deep learning methodologies, and traditional classification frameworks.

4.5 Incorporating Mel-frequency cepstral coefficient (MFCC)

Aykanat et al. [13] investigated an innovative approach to classifying respiratory sounds recorded by an electronic stethoscope, focusing on the integration of Mel-frequency cepstral coefficient (MFCC) features within a convolutional neural network (CNN) architecture.

MFCC features, a key component in speech processing tasks, provide a compact representation that captures both spectral and temporal features required for analysing respiratory sounds [13]. These features are extracted from audio signals via a series of processing steps such as Fourier transformation, logarithmic compression, and discrete cosine transformation. By incorporating MFCC features into the CNN framework, the model gains access to rich information relevant to respiratory sound classification, allowing for more effective learning and representation of sound patterns.

The study tested four classification tasks and compared the performance of CNN and support vector machine (SVM) algorithms. Notably, the accuracy results showed that CNN and SVM performed competitively across different tasks, with CNN outperforming SVM [13]. Specifically, the experiments revealed accuracies of 86% for both CNN and SVM in distinguishing between healthy and pathological lung sounds, 76% for CNN and 75% for SVM in classifying specific types of respiratory sounds, 80% for both CNN and SVM in identifying single respiratory sound types, and finally 62% for both CNN and SVM in classifying audio types encompassing all sound categories [13].

These results demonstrate the effectiveness of incorporating MFCC features into CNN architectures for respiratory sound classification tasks. The proposed approach achieves promising results in accurately classifying and pre-diagnosing respiratory audio by combining the discriminative power of CNNs with the informative representation provided by MFCC features [13].

4.6 The use of other ML Classifiers (such as: Random Forest)

Balasubramanian and Rajadurai [14] made significant advances in the classification of pulmonary diseases by applying machine learning techniques to real-time lung sound data. The researchers introduced a computer-based automated system that uses machine learning classifiers, with a particular emphasis on the random forest algorithm's performance.

To improve classifier performance, the system used denoising techniques such as the discrete wavelet transform and variational mode decomposition. Furthermore, cepstral features such as MFCC and gammatone frequency cepstral coefficients were combined for classification. This comprehensive approach aimed to accurately distinguish between various pulmonary diseases, such as chronic obstructive pulmonary disease, asthma, bronchiectasis, and healthy individuals.

Four machine learning classifiers were tested: decision trees, k-nearest neighbours, linear discriminant analysis, and random forests. Notably, the random forest classifier performed best, with an accuracy of 99.72% and perfect recall, specificity, and f1 scores. These findings highlight the robustness and efficacy of random forest in accurately classifying pulmonary diseases based on lung sound data.

The study's findings highlight the computer-based system's potential as a dependable decision-making tool for pulmonary disease classification, especially in resource-limited settings. The superior performance of random forest suggests that it is appropriate for real-world applications in health-care, where precise and efficient disease classification is critical for effective patient management and treatment planning.

5 Methodology

This section discusses the methodology, describing the experiments and providing an overview of the code. It is divided into two sections: one covering the CNN & ANN experiments in Python, and the other focusing on the SVM and RF experiments performed in R.

5.1 Experimental Setup

Python version 3.9.6 was the primary programming language used in both the CNN and ANN sections to conduct experiments. Using a variety of libraries and packages, tasks ranging from data manipulation to audio processing, model evaluation, and neural network construction were completed efficiently.

These operations relied heavily on key modules such as 'os' for directory operations, 'pandas' for efficient data manipulation, 'librosa' for complex audio processing, and 'matplotlib' for insightful plotting. Furthermore, libraries such as 'sklearn' facilitated model evaluation, 'seaborn' improved data visualisation, and 'tensorflow.keras' enabled the creation of neural networks, resulting in a comprehensive toolkit for experimentation.

In contrast, the SVM and RF sections made use of R version 4.4.0 as the primary programming language. Specialised libraries and packages were used to manipulate data, process audio, evaluate models, and implement machine learning algorithms.

This domain's essential libraries included 'tidyverse', 'fs', 'purrr', 'dplyr', 'seewave', 'tuneR', 'reshape2', 'caTools' and 'e1071', all of which played critical roles in tasks such as data preprocessing, feature extraction, and the last for SVM model training. Similarly, the 'randomForest' library was useful for RF experiments, ensuring reliable model training and evaluation.

All experiments mentioned in 5, were carried out on a MacBook Air M2 (2022) with 8 GB of Unified memory, running macOS Sonoma 14.4.1. This hardware configuration provided the computational resources required to run experiments efficiently, resulting in timely completion. Furthermore, it's important to note that the dataset used for these experiments was sourced from Kaggle², which added real-world data to the research.

5.2 CNN & ANN

5.2.1 Data loading and preprocessing

The experiment's first phase was primarily concerned with data collection. Audio files and diagnosis data were obtained from the publicly available Kaggle dataset. The dataset contains a variety of respiratory conditions, including normal (healthy), wheeze, crackle, and other anomalies. To ensure consistency and quality, audio files were converted to a uniform format using Python libraries such as 'librosa' for reading and processing. Noise reduction and normalisation techniques were used to improve the quality of the audio signals.

Following standardisation, relevant audio features were extracted and used as input for the models. Mel-frequency cepstral coefficients (MFCCs) were calculated from audio signals. These features capture essential characteristics of respiratory sounds, allowing neural network models to classify them more effectively. The extraction process was carried out using 'librosa.feature.mfcc', which computes the MFCCs for each audio file, resulting in a reliable representation of the sound data.

²<https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database>

5.2.2 Data Visualisation

To gain insight into the dataset, various visualisation techniques were used. The distribution of classes was visualised using bar plots, which helped to understand the dataset's class imbalance. Identifying class imbalance is critical because it can influence model performance, potentially leading to biased predictions for the majority class.

Additionally, the audio file durations were analysed and visualised to ensure consistency across samples. This analysis ensured that the recording lengths were not significantly different, which could have an impact on feature extraction and model training. Visualisations were created using libraries such as 'matplotlib' and 'seaborn' to produce plots that provide a thorough understanding of the dataset's characteristics.

5.2.3 Model Building

The next step was to create the CNN and ANN models. To capture the spatial structures in the data, the CNN was designed with a strong architecture. The CNN architecture consisted of multiple convolutional layers with filters for extracting different features from audio signals. Each convolutional layer applied a set of filters to the input data, resulting in feature maps that highlight various aspects of the audio signals. The hierarchical extraction of features enabled the CNN to learn increasingly complex data representations.

Max-pooling layers were used after the convolutional layers to reduce the data's dimensionality. These select the maximum value from a set of neighbouring values in the feature map, resulting in a downsampled input representation. This process not only reduces computational load, but it also improves feature detection's robustness to translations and distortions in input data. Max-pooling layers improve model generalisation by retaining only the most prominent features, preventing overfitting.

To further prevent overfitting, dropout layers were added to the DL algorithm. Dropout is a regularisation technique that randomly sets a subset of the input units to zero during each training update. This process forces the network to learn redundant representations of the data, reducing reliance on specific neurons and improving the model's generalizability. A dropout rate of 0.3 is used (i.e., 30% of the units are dropped), but this can be adjusted depending on the model and dataset's specific requirements.

To perform the final classification, the network's fully connected layers aggregated the extracted features. These layers are densely connected, which means that each neuron in one layer is connected to every neuron in the next layer. This configuration enables a comprehensive combination of the features learned by the preceding layers, culminating in the output layer, which provides the final class probability. The output layer used the softmax activation function, which is appropriate for multi-class classification tasks.

The CNN model was built using the categorical cross-entropy loss function and the Adam optimizer. The categorical cross-entropy loss function is suitable for multi-class classification because it computes the difference between the predicted probability distribution and the true distribution. On the other hand, the Adam optimizer is an efficient gradient-based optimisation algorithm that adjusts the learning rate during training, making it ideal for large datasets and complex models. To achieve the best possible performance, hyperparameters such as the learning rate, batch size, and number of epochs were carefully adjusted.

Similarly, an ANN architecture was constructed with multiple dense layers and dropout layers. The ANN, which has a simpler architecture than the CNN, is based on fully connected layers, with each neuron connected to every neuron in the previous layer. This architecture enables the ANN to learn representations of data based on the weighted sums of input features.

The dense layers in the ANN were created to capture the intricate relationships between input features and output classes. Each dense layer is followed by an activation function, usually the ReLU (Rectified Linear Unit) activation function, which adds non-linearity to the model and allows it to learn more complex patterns.

Similar to the CNN, dropout layers were used as well as categorical cross-entropy loss function and the Adam optimizer. The ANN's hyperparameters, such as the number of layers, neurons in each layer, learning rate, batch size, and number of epochs, were fine-tuned to achieve peak performance.

Both models, were created to handle the complexities of respiratory sound data while producing accurate classifications. The study aimed to develop models that can effectively distinguish between different respiratory conditions by leveraging the strengths of convolutional operations in CNNs and dense connections in ANNs, with potential applications in automated respiratory disease diagnosis.

5.2.4 Model Training

Both CNN and ANN models were trained by feeding pre-processed datasets into their respective architectures. The dataset was divided into training and testing sets to keep track of the models' performance while training, preventing overfitting and ensuring generalizability. Early stopping monitored the validation loss and stopped the training process if it did not improve after a set number of epochs, thereby preventing overfitting. Model checkpointing saved the model's weights at the epoch with the lowest validation loss, ensuring that the best version of the model was retained. These strategies ensured that the models were trained effectively while avoiding overfitting, resulting in improved performance on unseen data.

5.2.5 Model Evaluation

Evaluating the models' performance was an important step in this study. Various performance metrics were used, including accuracy, loss, confusion matrix, ROC curves, and precision-recall curves. These metrics provided a comprehensive assessment of the models' abilities.

Confusion matrices were used to demonstrate the classification performance of each model. These matrices provided detailed insights into the true positives, true negatives, false positives, and false negatives for each class, highlighting where the models predicted correctly and incorrectly.

ROC curves were created to assess the models' ability to differentiate between classes. The ROC curve compares the true positive rate to the false positive rate at various threshold settings, giving a comprehensive picture of the models' performance across thresholds. Precision-recall curves were analysed to better understand the trade-offs between precision and recall at different threshold values. Precision assesses the proportion of true positive predictions among all positive predictions, whereas recall assesses the proportion of true positive predictions among all actual positives. These metrics are especially useful when dealing with imbalanced datasets, as they provide information about how well the models identify minority classes.

Using these metrics, a detailed comparison of the CNN and ANN models was performed, highlighting their strengths and areas for improvement. These visualisations provided a better understanding of the models' performance and guided future refinement.

5.3 SVM & RF

5.3.1 Loading necessary libraries

The first step involved loading the libraries needed for data manipulation, analysis, and modelling. Several R libraries were used, including 'tidyverse' for comprehensive data manipulation, 'fs' for efficient file system operations, 'purrr' for functional programming paradigms, 'dplyr' for streamlined data manipulation, 'seewave' and 'tuneR' for specialised audio processing, 'reshape2' for versatile data reshaping, 'caTools' for data splitting, 'e1071' for SVM implementation, and 'randomForest' for building the Random Forest classifier. These libraries provided the robust tools required for managing the various aspects of data preparation and model building in this study.

5.3.2 Data Preparation

Data preparation began with defining the file path to the respiratory sound database and verifying the directory's existence. The directory contained audio and text files required for this research. Ensuring the directory existed was critical to avoiding interruptions during the data loading process. The .txt files within the specified directory were then extracted into a list. If no .txt files were found, the process was halted with an appropriate error message informing the user of the missing data.

To improve efficiency, the .txt files were read and combined into a single data frame. The contents of each file were converted into data frames, which were then merged into a single cohesive data frame using 'purrr::map_dfr'. Following that, patient identifiers were extracted from file names using regular expressions. The number of rows in each data frame was calculated to aid in the accurate assignment of patient IDs. These IDs were written to a file called 'PatientID.txt', and the number of repetitions of each ID corresponded to the number of rows in the respective data frame, ensuring that patient records were accurately mapped to their ID.

5.3.3 Data manipulation and analysis

The combined data frame was further processed to select specific columns and compute new ones. Columns V1–V4 were selected, and new columns V5 (representing recording duration) and V6 (indicating normal or abnormal recordings based on conditions) were calculated. The columns were renamed to improve readability. This cleaned and structured data was combined with the patient IDs to form a comprehensive dataset suitable for further analysis.

5.3.4 Reading the .wav files

Processing the .wav files entailed extracting frequency data from audio recordings. A connection was established to write frequency data to the 'Freq.txt' file. All .wav files in the specified directory were listed, and each was read with the readWave function. Spectrograms of audio files were created using the 'spectro' function without plotting. The maximum frequency from each spectrogram was extracted and saved to the 'Freq.txt' file, with the number of repetitions corresponding to the text file entries. This process ensured that the frequency data from each recording was accurately captured and stored.

5.3.5 Integrating Frequency Data into Patient Records

The frequency data stored in 'Freq.txt' was read and entered into the patient's records. This step involved combining the patient and frequency data and renaming the columns to reflect their contents. This integration was critical in providing a comprehensive dataset that included both audio and demographic information for each patient.

5.3.6 Integrating Demographic Information into Patient Records

Demographic data was retrieved from a separate file and integrated into the patient record. This entailed grouping the demographic data by patient IDs and selecting the appropriate columns. The Sex and Age of each patient were updated using the matching IDs from the demographic data. The columns were then reordered so that Normal_VS_Abnormal came after Age, ensuring the dataset's logical structure. This integration gave each patient a comprehensive view of their demographic and audio characteristics.

5.3.7 Preparing the Patient Data Subset

A subset of the patient data was prepared by transforming and converting individual columns. To facilitate modelling, the Sex column was converted to binary values (1 for male and 0 for female), while the remaining columns were converted to numeric types. The Normal_VS_Abnormal column was converted into a factor with levels 0 and 1 to prepare the dataset for classification tasks. This step ensured that the data was properly formatted for training machine learning models.

5.3.8 Data Splitting and Scaling

To ensure a thorough evaluation of the models, the data was divided into training and test sets with a 75:25 split ratio. Features in both the training and test sets were scaled, with the exception of the target variable, to ensure that all features contributed equally to model training. This scaling process is required for algorithms such as SVM and Random Forest to perform optimally because it prevents any single feature from having a disproportionate influence on the model due to scale differences.

5.3.9 Model Creation and Prediction

The data cleaning process consisted of removing rows with missing values from both the training and test sets to ensure that the models were trained on complete and accurate data. This step was critical to maintaining the dataset's integrity and avoiding any biases or errors that missing values could introduce. Following data cleaning, the SVM model was built with the 'svm' function and a radial kernel. The radial kernel was chosen for its ability to handle non-linear relationships in data, which is common with complex biomedical signals like respiratory sounds. The SVM model was trained on the cleaned training set, and predictions were then made on the test set to assess performance.

Similarly, the Random Forest classifier was trained with the 'randomForest' function. This ensemble learning method combines several decision trees to improve classification accuracy and robustness. The Random Forest classifier was trained on the same cleaned training set before making predictions for the test set. Random Forest was chosen because of its ability to handle large datasets with higher dimensionality and its effectiveness in managing overfitting via decision tree aggregation.

5.3.10 Evaluation

Model performance was assessed using confusion matrices and accuracy measurements. Confusion matrices for both the SVM and Random Forest models were created, providing detailed information about classification performance, such as true positives, true negatives, false positives, and false negatives. Accuracy was calculated as the proportion of correct predictions out of all predictions, and the results were converted to percentages for ease of understanding. A bar chart was created to visually compare the accuracies of the SVM and Random Forest classifiers, resulting in a clear and concise summary of model performance. This evaluation step was critical for understanding each model's strengths and weaknesses, as well as identifying areas for future improvement.

6 Results

In this section, the results of four different machine learning techniques will be evaluated. Each model was meticulously trained and fine-tuned to perform specific tasks. The goal of rigorous analysis is to reveal not only individual models' performance metrics, but also to draw insightful comparisons between two types of machine learning technologies: deep learning models (CNN and ANN) and traditional machine learning algorithms (SVM and RF). The goal of examining their accuracies, robustness, and computational efficiency is to discover each approach's strengths and limitations, providing valuable insights for future ML applications and advancements.

This procedure involved converting audio data into spectrograms, which are graphical representations of the intensity variations of frequencies over time. This transformation makes it easier to extract useful insights for data analytics. Figure 1 illustrates a randomly chosen .wav file as a spectrogram.

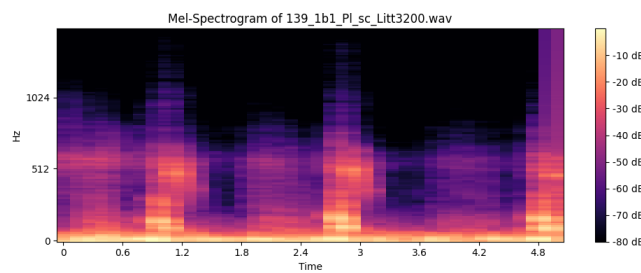


Figure 1: Sample Spectrogram generated by Jupyter Notebook

6.1 CNN

In this study, a Convolutional Neural Network (CNN) was used to classify respiratory conditions, with accuracy ranging from 86% to 90% across various training sessions. The optimal model configuration was established using the architecture shown in Figure 2. This model was compiled and trained over 74 epochs with a batch size of 92, resulting in a **final accuracy of 90% and a loss of 0.5765** as shown in Figure 3.

```
iAPT Assignment - CNN & ANN.ipynb

1 def build_model(input_shape, num_classes):
2     model = Sequential([
3         ConvD(64, kernel_size=5, activation='relu', input_shape=input_shape),
4         ConvD(128, kernel_size=5, activation='relu'),
5         MaxPoolingD(pool_size=2),
6         ConvD(256, kernel_size=5, activation='relu'),
7         Dropout(0.3),
8         Flatten(),
9         Dense(512, activation='relu'),
10        Dense(num_classes, activation='softmax')
11    ])
12    return model
13
14 def compile_and_train_model(model, X_train, y_train, X_test, y_test, epochs=74, batch_size=92):
15     model.compile(loss='categorical_crossentropy', optimizer=Adam(), metrics=['accuracy'])
16     history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=batch_size, verbose=1)
17     print("\033[92mCNN Model completed!\033[0m")
18     return history
19
20 # Define input shape and number of classes
21 input_shape = (193, 1)
22 num_classes = 6
23
24 # Build the model
25 model = build_model(input_shape, num_classes)
26
27 # Compile and train the model
28 history = compile_and_train_model(model, X_train, y_train, X_test, y_test)
```

Figure 2: Screenshot showing code snippet of CNN

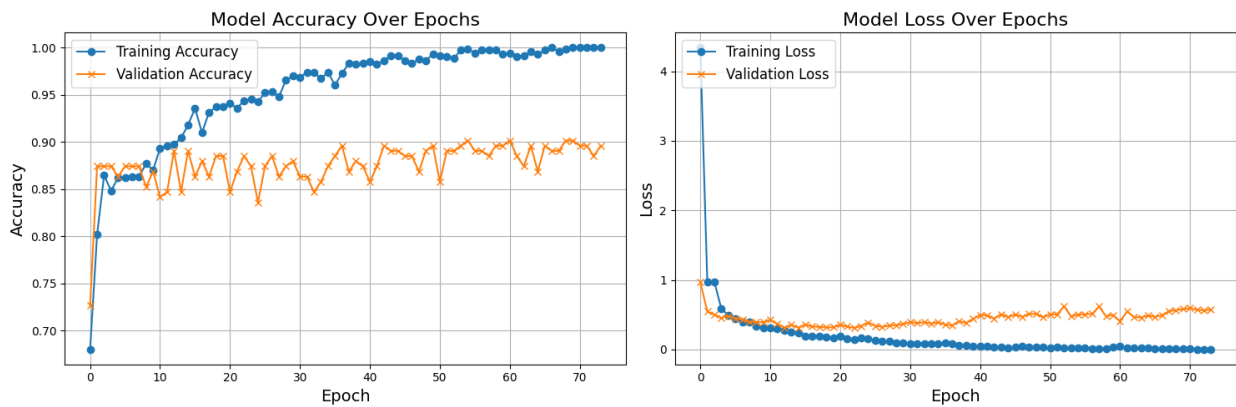


Figure 3: Model Accuracy and Loss Over Epochs

6.1.1 Confusion Matrix Evaluation

The confusion matrix in Figure 4 shows that the CNN model accurately predicted COPD, correctly identifying 97.5% of true COPD cases. This high sensitivity in detecting COPD demonstrates the model's robustness and dependability in recognising this condition, which is critical for prompt and effective treatment. However, the model struggled with the 'Healthy' and 'URTI' categories, accurately predicting only 14.3% and 0.0% of true cases, respectively. This implies that, while the model is good at identifying obvious pathological conditions like COPD, it may be less effective at distinguishing between subtler or less distinct conditions such as a healthy state or upper respiratory tract infections (URTI).

		Confusion Matrix					
True Label	COPD	97.5% 156/160		0.6% 1	0.6% 1	1.2% 2	
	Healthy	14.3% 1	14.3% 1/7	42.9% 3			28.6% 2
	URTI	50.0% 1	50.0% 1	0.0% 0/2			
	Bronchiectasis	40.0% 2			60.0% 3/5		
	Pneumonia	25.0% 1				75.0% 3/4	
	Bronchiolitis			60.0% 3	20.0% 1		20.0% 1/5
		Predicted Label					
		COPD	Healthy	URTI	Bronchiectasis	Pneumonia	Bronchiolitis

Figure 4: Confusion Matrix (CNN) Generated by Jupyter Notebook

The primary cause of this performance disparity is a dataset imbalance. Specifically, the dataset included 793 COPD samples but only 23 URTI samples, with even fewer samples for the other conditions. Figure 5 shows a bar chart that illustrates this imbalance. Such a skewed distribution of training data causes a bias in the model, which improves its ability to predict more frequent conditions while struggling with less frequent ones. This emphasises more representative training data to improve its performance across all categories.

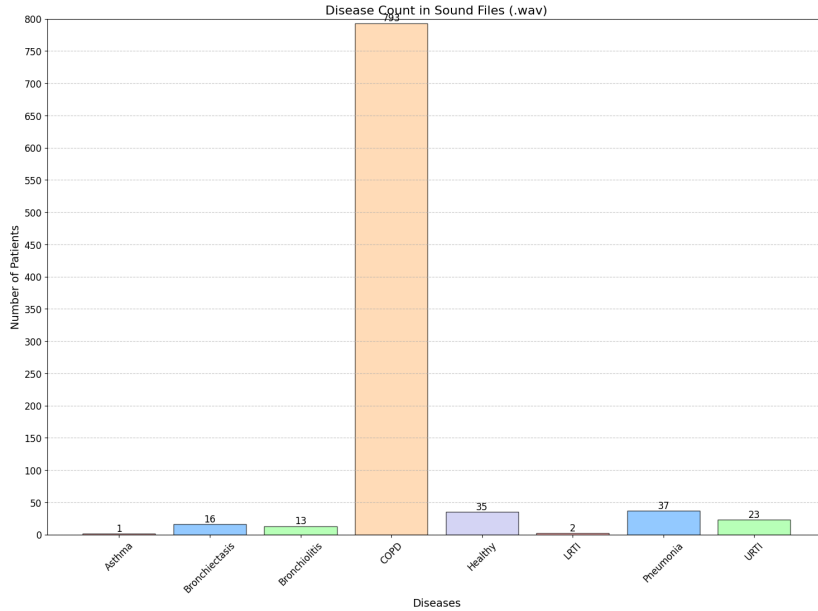


Figure 5: Bar Chart of Disease Count in Sound Files (.wav) generated by Jupyter Notebook

6.1.2 Receiver Operating Characteristic (ROC) Curves & Area Under Curve (AUC)

The ROC curves, depicted in Figure 6, provide a graphical representation of the true positive rate versus the false positive rate for each class. The areas under the ROC curves (AUC) indicate the following performances, as shown in Table 1:

Class	AUC
COPD	0.98
Healthy	0.97
URT	0.90
Bronchiectasis	0.97
Pneumonia	0.94
Bronchiolitis	0.95

Table 1: AUC values for different classes

The high AUC values once again indicate that the model is effective at distinguishing between the various classes, particularly for 'Healthy' and 'COPD' cases. This suggests that the model has a high overall ability to correctly classify instances of both positive (diseased) and negative (healthy) cases. The nearly perfect AUC for the 'Healthy' class indicates that the model is good at detecting non-diseased cases, which is important for avoiding unnecessary treatments or interventions.

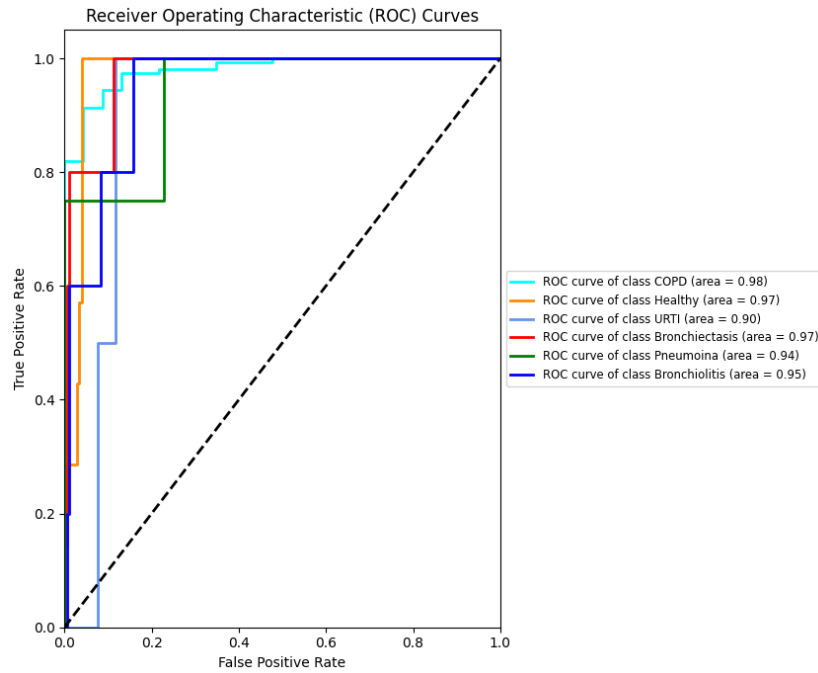


Figure 6: ROC curves & AUC for each Class (CNN) generated by Jupyter Notebook

6.1.3 Precision-Recall Curves

The precision-recall curves in Figure 7 demonstrate the trade-off between precision (the accuracy of positive predictions) and recall (the ability to find all positive samples) for each class. The results show:

- **COPD:** High precision and recall indicate reliable predictions with low false positives.
- **Healthy:** Lower precision and recall indicate difficulty in correct classification, potentially leading to both false positives and false negatives in identifying healthy individuals.
- **URTI, Bronchiectasis, Pneumonia, and Bronchiolitis:** Moderate precision and recall indicate potential for improvement in distinguishing these conditions. This implies that, while the model is effective, it may miss some true cases or misclassify some cases, potentially influencing clinical decision-making.

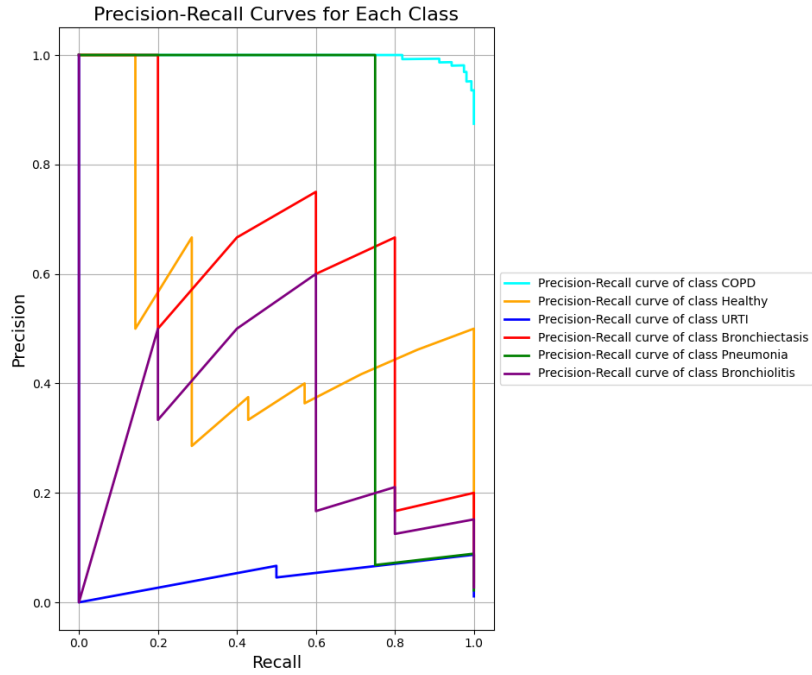


Figure 7: Precision-Recall Curves for each Class (CNN) generated by Jupyter Notebook

6.1.4 Discussion

The CNN model demonstrated a strong performance in predicting COPD and achieved overall good classification metrics across other respiratory conditions. However, the lower accuracy in distinguishing 'Healthy' and 'URTI' cases indicates a need for further refinement, possibly through additional data augmentation or architectural adjustments. The ROC and precision-recall curves affirm the robustness of the model in a clinical setting, highlighting its potential utility in aiding respiratory disease diagnosis.

6.2 ANN

Similarly, an Artificial Neural Network (ANN) was used to classify respiratory conditions, with accuracy ranging between 89% and 92% across different training sessions. The optimal model configuration was established using the architecture shown in Figure 8. This model was compiled and trained over 70 epochs with a batch size of 5, resulting in a **final accuracy of 92% and a loss of Loss: 0.2931**, as seen in Figure 9.

```
IAPT Assignment - CNN & ANN.ipynb

1 # Define the ANN model
2 def build_ann_model(input_shape, num_output_classes):
3     ann_model = Sequential([
4         Dense(256, activation='relu', input_shape=input_shape),
5         Dropout(0.3),
6         Dense(256, activation='relu'),
7         Dropout(0.3),
8         Dense(128, activation='relu'),
9         Dropout(0.3),
10        Dense(num_output_classes, activation='softmax')
11    ])
12    return ann_model
13
14 def compile_and_train_ann_model(ann_model, training_data, training_labels, validation_data, validation_labels, num_epochs=70, batch_size_value=5):
15     ann_model.compile(loss='categorical_crossentropy', optimizer=Adam(), metrics=['accuracy'])
16     training_history = ann_model.fit(training_data, training_labels, validation_data=(validation_data, validation_labels), epochs=num_epochs, batch_size=batch_size_value, verbose=1)
17     print("\033[92mANN Model training completed!\033[0m")
18     return training_history
19
20 # Define input shape and number of classes
21 input_shape_ann = (193,)
22 num_output_classes = 6
23
24 # Build the ANN model
25 ann_model = build_ann_model(input_shape_ann, num_output_classes)
26
27 # Compile and train the ANN model
28 training_history = compile_and_train_ann_model(ann_model, X_train, y_train, X_test, y_test)
```

Figure 8: Screenshot showing code snippet of ANN

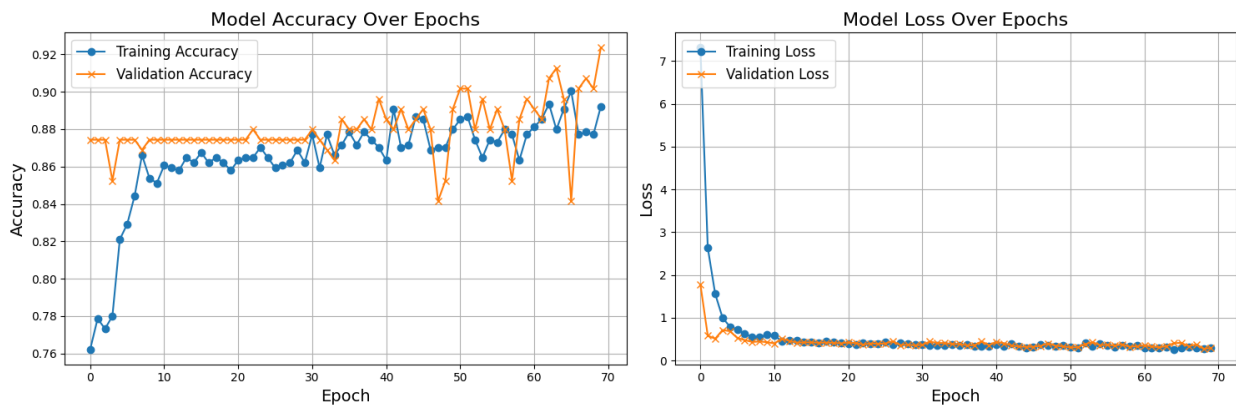


Figure 9: Model Accuracy and Loss Over Epochs

6.2.1 Confusion Matrix Evaluation

The confusion matrix in Figure 10 shows that the ANN model predicted COPD with high accuracy, identifying 99.4% of true COPD cases correctly. This high sensitivity in detecting COPD demonstrates the model's robustness and reliability in recognising this condition, which is critical for timely and effective treatment. The model also correctly identified 'Healthy' cases with 100% accuracy. However, there were some misclassifications in the other categories, indicating that there is still room for improvement in distinguishing these conditions.

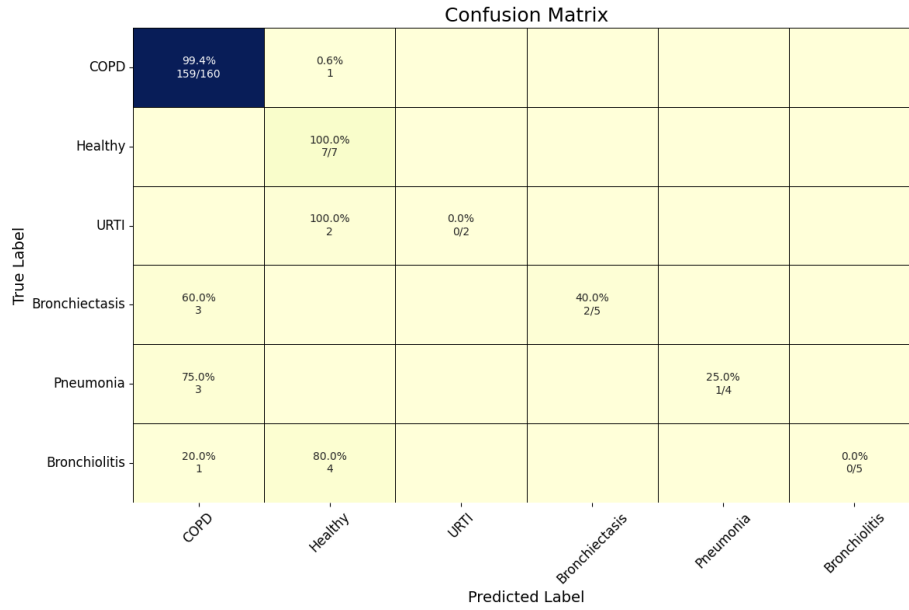


Figure 10: Confusion Matrix (ANN) Generated by Jupyter Notebook

6.2.2 Receiver Operating Characteristic (ROC) Curves & Area Under Curve (AUC)

Figure 11 shows the ROC curves, which represent the true positive rate versus the false positive rate for each class. The areas under the ROC curves (AUC) indicate the following performances, as shown in Table 2:

Class	AUC
COPD	0.98
Healthy	0.98
URTI	0.94
Bronchiectasis	0.96
Pneumonia	0.99
Bronchiolitis	0.90

Table 2: AUC values for different classes

Once again, the high AUC values indicate the model's ability to distinguish between different classes, particularly between 'Healthy', 'Pneumonia' and 'COPD' cases. This indicates a strong overall ability to accurately categorise instances of both positive (diseased) and negative (healthy) cases. This similarity is consistent with the results of the previous CNN study, demonstrating the model's effectiveness across multiple methodologies.

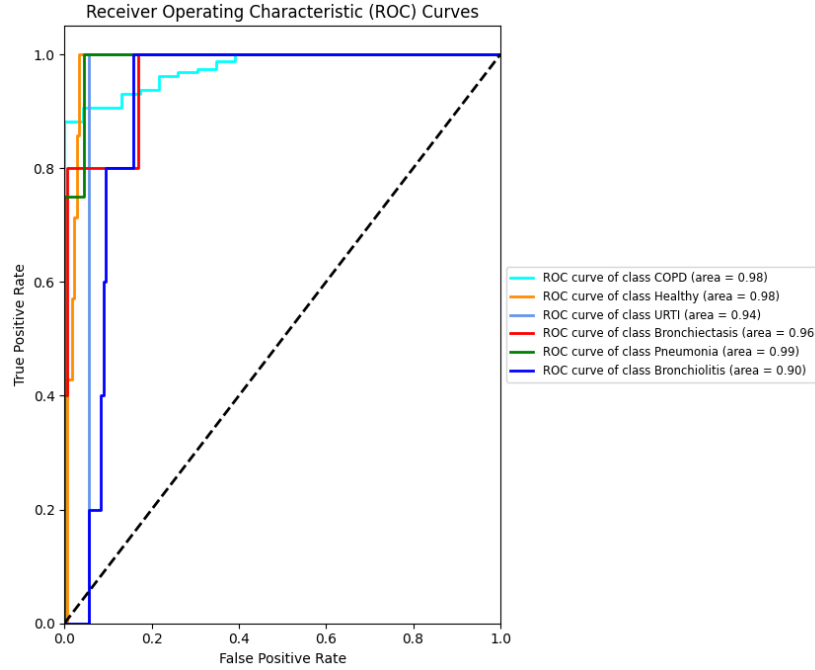


Figure 11: ROC curves & AUC for each Class (ANN) generated by Jupyter Notebook

6.2.3 Precision-Recall Curves

The precision-recall curves in Figure 12 highlight the trade-off between precision (the accuracy of the positive predictions) and recall (the ability to find all the positive samples) for each class. The results reveal:

- **COPD, Healthy and Pneumonia:** High precision and recall, indicating reliable predictions and a low rate of false positives.
- **Bronchiectasis & URTI:** Moderate precision and recall, suggesting room for improvement in distinguishing these conditions. This implies that the model, while effective, might miss some true cases or incorrectly classify some cases, potentially affecting clinical decision-making.

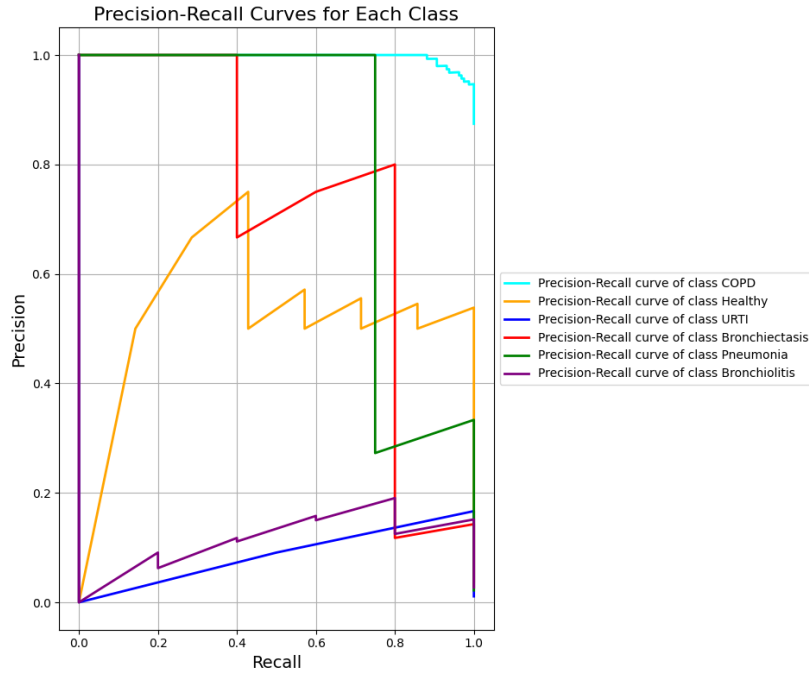


Figure 12: Precision-Recall Curves for each Class (ANN) generated by Jupyter Notebook

6.2.4 Discussion

The ANN model performed very well in predicting 'COPD' and 'Healthy' cases, indicating its potential use in clinical diagnostics. However, the model made significant misclassifications in 'Bronchiectasis', 'URTI' and 'Bronchiolitis', indicating the need for further refinement to improve differentiation between similar conditions. Enhancing the model with augmented datasets, advanced techniques such as ensemble learning, and additional clinical features may reduce false positives and negatives. Future research should once again focus on using a more diverse datasets and conducting extensive validation to ensure robustness and generalizability, with the goal of making the ANN a valuable tool in respiratory disease diagnosis and management.

6.3 Comparison of the two models (CNN & ANN)

A comparison of the CNN and ANN models for respiratory condition classification reveals both similarities and differences in performance metrics. The CNN model achieved a final accuracy of 90.16% and a loss of Loss: 0.5765, while the ANN model achieved a slightly higher accuracy of 92.35% and a lower loss of 0.2931. Figure 13 depicts these differences, demonstrating the ANN model's slight advantage in accuracy and lower loss over the CNN model. Despite the similar accuracy rates, the ANN model's loss performance indicates that it has a more stable learning process and may have better generalisation capabilities on previously unseen data.

In terms of specific condition classification, the confusion matrix evaluation shows that both models excel at detecting COPD, with the CNN correctly identifying 97.5% and the ANN correctly identifying 99.4% of true COPD cases. However, both models struggled with the 'Healthy' and 'URTI' categories, with the CNN correctly predicting only 14.3% of 'Healthy' cases and the ANN achieving 100% accuracy in 'Healthy' cases but making significant misclassifications in 'URTI'. The imbalance in the dataset, as shown in Figure 5, most likely contributed to the misclassifications. Both models achieved high AUC values for 'Healthy' and 'COPD' cases, indicating strong overall classification capabilities.

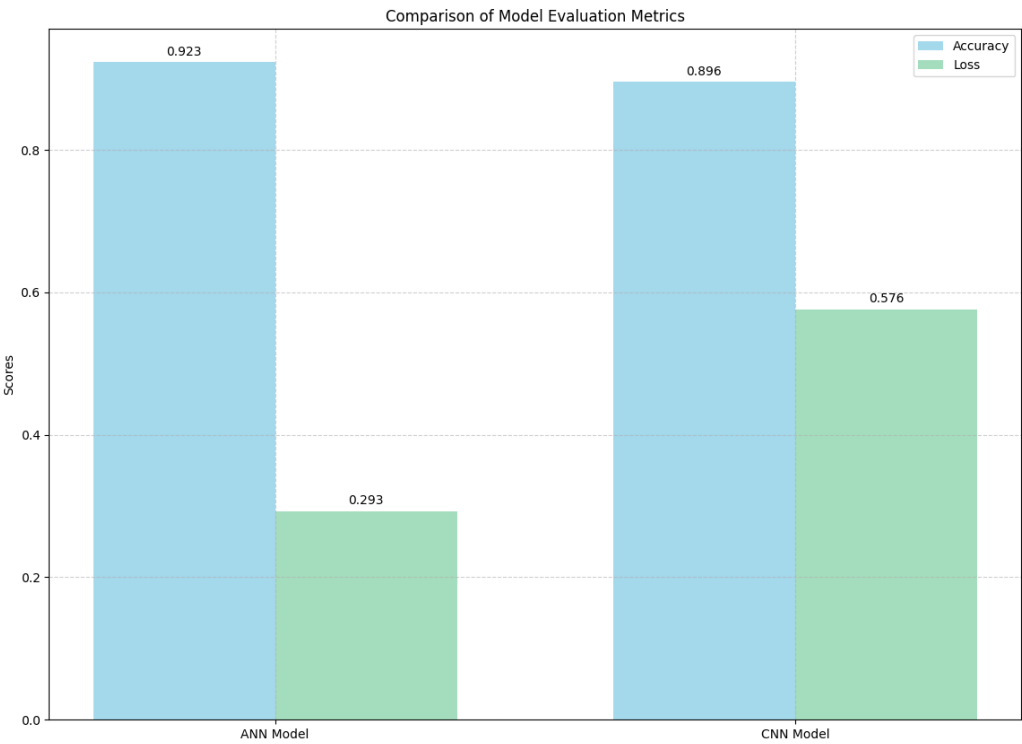


Figure 13: Comparison of Model Evaluation Metrics (CNN & ANN) generated by Jupyter Notebook

6.4 SVM

The SVM classifier was created using the radial kernel function on the training set. The code used to create the SVM classifier is as shown in Figure 14, outputting a **final accuracy of 70%**.

A screenshot of an R console window titled "SVM". It contains two code snippets. The first snippet creates an SVM classifier using the radial kernel function. The second snippet uses the trained classifier to make predictions on a test set.

```
{r}
# Create SVM classifier
svm_classifier <- svm(Normal_VS_Abnormal ~ .,
                      data = training_set,
                      type = 'C-classification',
                      kernel = 'radial')

# Make predictions using the SVM classifier
svm_predictions <- predict(svm_classifier, newdata = test_set[, -6])
```

Figure 14: Screenshot showing code snippet of SVM

6.4.1 Confusion Matrix Evaluation

In the context of respiratory sound classification, the confusion matrix shows how well the SVM model differentiated between healthy and ill individuals. In the given matrix (Figure 15), where 0 represents "healthy" and 1 represents "sick," the SVM model correctly classifies 592 instances as sick and 600 as healthy. However, it incorrectly identifies 302 healthy instances as sick and 217 sick instances as healthy. Thus the model's performance in identifying sick individuals (as represented by false negatives) requires careful consideration. False negatives could lead to undetected respiratory conditions, emphasising the need for further model refinement and evaluation to ensure accurate diagnosis and timely intervention in healthcare settings.

A screenshot of an R console window titled "Confusion Matrix Generation: SVM and Random Forest Classifiers". It contains two code snippets. The first snippet creates a confusion matrix for SVM predictions. The second snippet displays the confusion matrix for SVM predictions.

```
{r}
# Create Confusion Matrix for SVM predictions
cm_svm <- table(test_set[, 6], svm_predictions)

Display Confusion Matrix - SVM & RF

{r}
# Display Confusion Matrix for SVM
cat("\nConfusion Matrix for SVM predictions:\n")
print(cm_svm)
```

Confusion Matrix for SVM predictions:

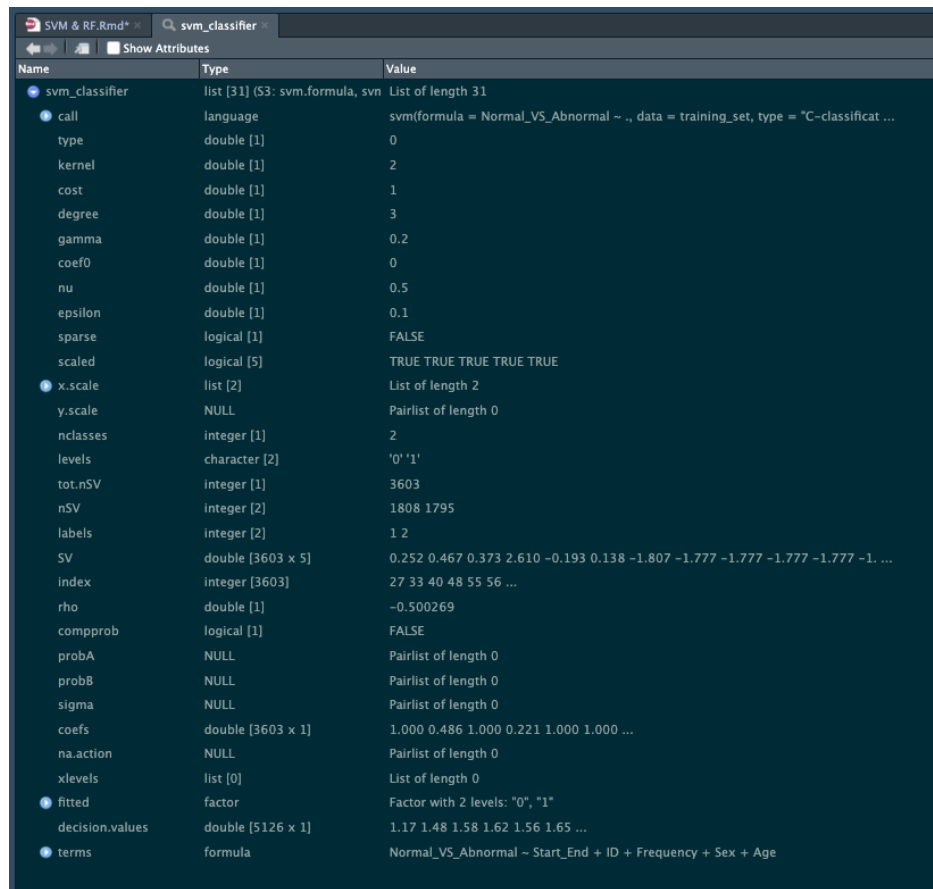
	svm_predictions	0	1
0	600	302	
1	217	592	

Figure 15: Confusion Matrix (SVM) Generated by R Markdown

6.4.2 Discussion

The SVM classifier classified respiratory sound data as normal or abnormal with an accuracy of 69.67%. While this accuracy suggests some predictive capability, it also highlights the difficulties in accurately distinguishing between normal and abnormal respiratory sounds based solely on the features used in the model.

One implication of this finding is that SVMs could be useful in assisting healthcare professionals in conducting preliminary assessments of respiratory conditions. However, the moderate accuracy indicates that additional features or more sophisticated modelling techniques may be required to improve classification performance.



The screenshot shows the RStudio interface with a data frame named 'svm_classifier'. The 'Show Attributes' window is open, displaying the following table of attributes:

Name	Type	Value
svm_classifier	list [31] (S3: svm.formula, svm	List of length 31
call	language	svm(formula = Normal_VS_Abnormal ~ ., data = training_set, type = "C-classificat ...
type	double [1]	0
kernel	double [1]	2
cost	double [1]	1
degree	double [1]	3
gamma	double [1]	0.2
coef0	double [1]	0
nu	double [1]	0.5
epsilon	double [1]	0.1
sparse	logical [1]	FALSE
scaled	logical [5]	TRUE TRUE TRUE TRUE TRUE
x.scale	list [2]	List of length 2
y.scale	NULL	Pairlist of length 0
nclasses	integer [1]	2
levels	character [2]	"0" "1"
tot.nSV	integer [1]	3603
nSV	integer [2]	1808 1795
labels	integer [2]	1 2
SV	double [3603 x 5]	0.252 0.467 0.373 2.610 -0.193 0.138 -1.807 -1.777 -1.777 -1.777 -1.777 -1. ...
index	integer [3603]	27 33 40 48 55 56 ...
rho	double [1]	-0.500269
compprob	logical [1]	FALSE
probA	NULL	Pairlist of length 0
probB	NULL	Pairlist of length 0
sigma	NULL	Pairlist of length 0
coefs	double [3603 x 1]	1.000 0.486 1.000 0.221 1.000 1.000 ...
na.action	NULL	Pairlist of length 0
xlevels	list [0]	List of length 0
fitted	factor	Factor with 2 levels: "0", "1"
decision.values	double [5126 x 1]	1.17 1.48 1.58 1.62 1.56 1.65 ...
terms	formula	Normal_VS_Abnormal ~ Start_End + ID + Frequency + Sex + Age

Figure 16: Screenshot showing code SVM Data Frame in RStudio

6.5 RF

The Random Forest classifier was trained using the 'randomForest' algorithm on the training set. The code snippet used to train the Random Forest classifier is shown in Figure 17, resulting in a **final accuracy of 74%**.



```
RF

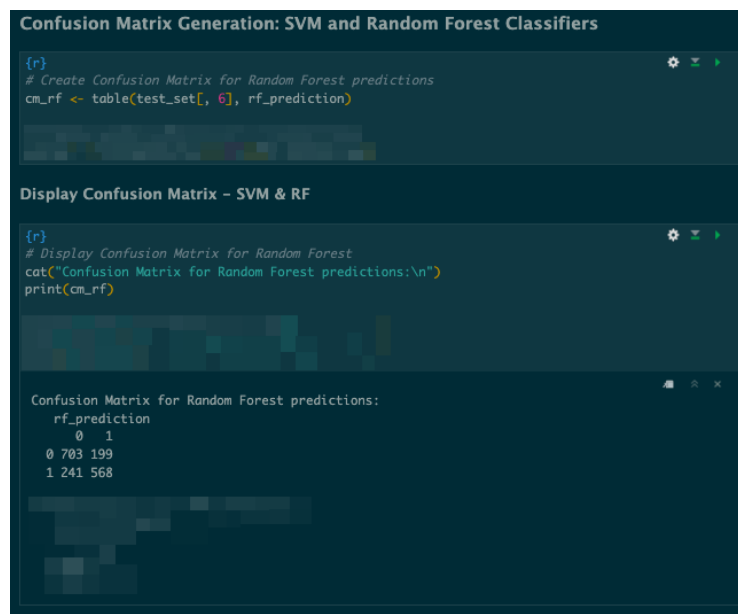
{r}
# Train random forest classifier
classifier_randomForest <- randomForest(x = training_set[-6],
                                         y = training_set$Normal_VS_Abnormal,
                                         ntree = 15)

# Predict using the trained classifier
rf_prediction <- predict(classifier_randomForest, newdata = test_set[-6])
```

Figure 17: Screenshot showing code snippet of RF

6.5.1 Confusion Matrix Evaluation

In the context of respiratory sound classification, the confusion matrix once again demonstrates the Random Forest model's ability to distinguish between healthy and ill individuals. In the provided matrix (Figure 18), the Random Forest model correctly classifies 703 instances as healthy and 568 as unhealthy. However, it incorrectly classifies 199 healthy instances as sick and 241 sick instances as healthy. This demonstrates the model's ability to accurately identify sick individuals (as evidenced by false negatives) and emphasises the need for further refinement and evaluation.



```
Confusion Matrix Generation: SVM and Random Forest Classifiers

{r}
# Create Confusion Matrix for Random Forest predictions
cm_rf <- table(test_set[, 6], rf_prediction)

Display Confusion Matrix - SVM & RF

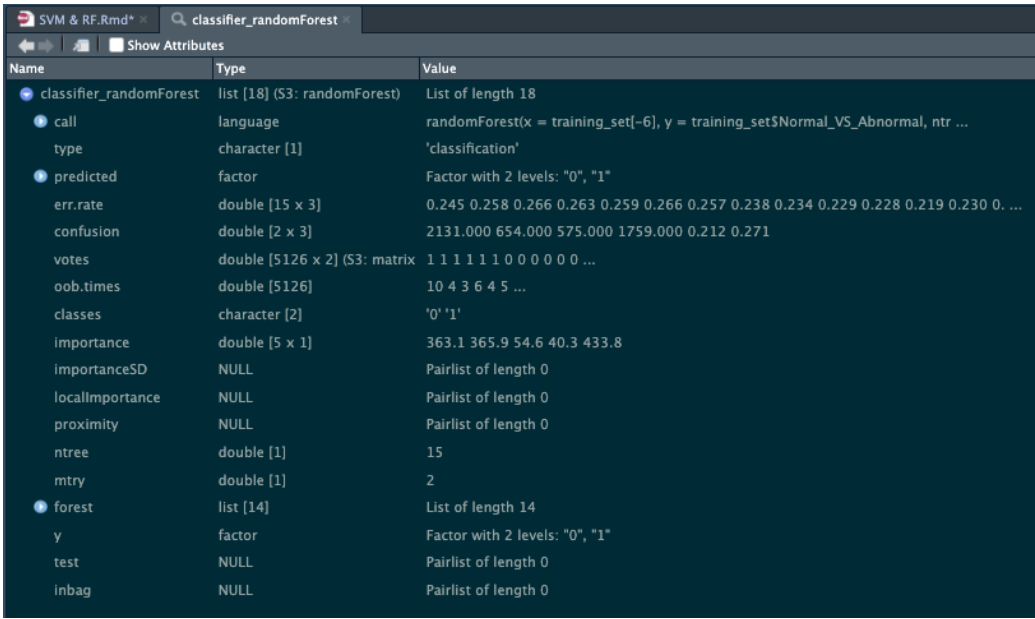
{r}
# Display Confusion Matrix for Random Forest
cat("Confusion Matrix for Random Forest predictions:\n")
print(cm_rf)

Confusion Matrix for Random Forest predictions:
 rf_prediction
 0    1
0 703 199
1 241 568
```

Figure 18: Confusion Matrix (RF) Generated by R Markdown

6.5.2 Discussion

The Random Forest classifier classifies respiratory sound data as normal or abnormal with an accuracy of 74.28%. This accuracy suggests that Random Forests have promising predictive capability, indicating that they could help healthcare professionals make preliminary assessments of respiratory conditions. However, more research is required to examine the model’s performance in various patient populations and respiratory conditions, as well as to identify additional features that could improve classification accuracy and robustness.



Name	Type	Value
classifer_randomForest	list [18] (S3: randomForest)	List of length 18
call	language	randomForest(x = training_set[-6], y = training_set\$Normal_VS_Abnormal, ntr ...
type	character [1]	'classification'
predicted	factor	Factor with 2 levels: "0", "1"
err.rate	double [15 x 3]	0.245 0.258 0.266 0.263 0.259 0.266 0.257 0.238 0.234 0.229 0.228 0.219 0.230 0. ...
confusion	double [2 x 3]	2131.000 654.000 575.000 1759.000 0.212 0.271
votes	double [5126 x 2] (S3: matrix)	1 1 1 1 1 0 0 0 0 0 ...
oob.times	double [5126]	10 4 3 6 4 5 ...
classes	character [2]	'0' '1'
importance	double [5 x 1]	363.1 365.9 54.6 40.3 433.8
importanceSD	NULL	Pairlist of length 0
localImportance	NULL	Pairlist of length 0
proximity	NULL	Pairlist of length 0
ntree	double [1]	15
mtry	double [1]	2
forest	list [14]	List of length 14
y	factor	Factor with 2 levels: "0", "1"
test	NULL	Pairlist of length 0
inbag	NULL	Pairlist of length 0

Figure 19: Screenshot showing code RF Data Frame in RStudio

6.6 Comparison of the two models (SVM & RF)

A performance comparison of the SVM and RF classifiers reveals that RF outperforms SVM in accurately distinguishing between healthy and sick individuals using respiratory sound data. Figure 20 shows that the Random Forest classifier achieved an accuracy of 74.28%, whereas the SVM classifier achieved an accuracy of 69.67%.

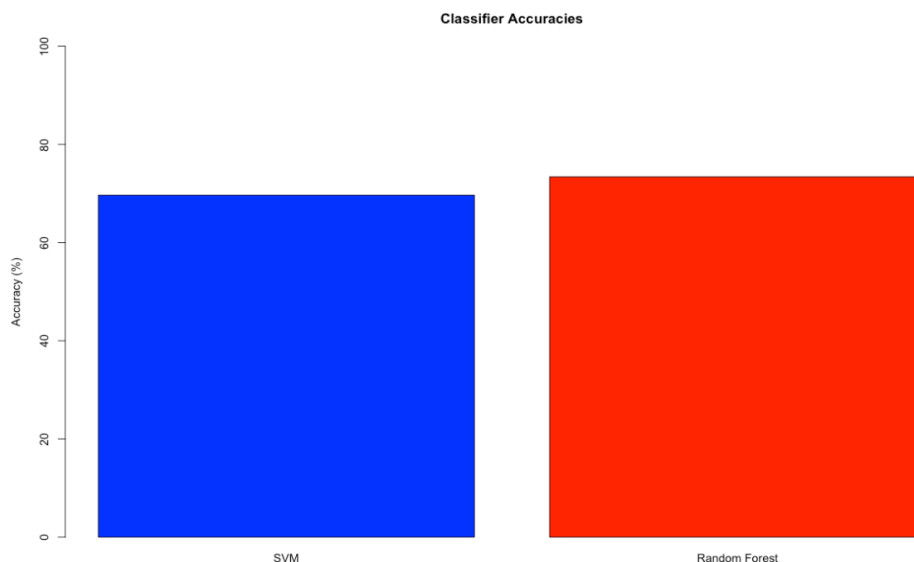


Figure 20: Comparison of Model Accuracy (SVM & RF) generated by R Markdown

The Random Forest classifier's higher accuracy can be attributed to the ensemble learning approach. Unlike SVM, which uses a single hyperplane to separate classes, Random Forest creates multiple decision trees and aggregates their predictions to reach a final decision. This process aids in the detection of complex patterns and interactions between features that SVM may miss.

Random Forest's superior performance can be attributed to its improved ability to handle high-dimensional data and complex feature interactions. In the context of respiratory sound classification, where the data may contain subtle and intricate differences between healthy and abnormal sounds, Random Forest's ensemble method is more robust and adaptable. Each tree in the forest casts a vote, lowering the likelihood of overfitting and improving generalisation to new data.

Furthermore, Random Forest's mechanism of averaging individual tree predictions reduces the impact of noisy or irrelevant features, which is especially useful in medical data where such issues are common. This ensemble approach produces more stable and reliable predictions, as demonstrated by the higher accuracy in Figure 22.

7 Future Research

Future research will focus on improving the performance and versatility of these models through a variety of approaches. Data augmentation methods such as time-stretching, pitch-shifting, and noise addition can be used to augment the training dataset, improving model robustness and generalisation.

Advanced feature engineering techniques like zero-crossing rate extraction, spectral roll-off analysis, and harmonicity assessment provide more detailed representations of audio signals, improving model accuracy.

Furthermore, optimising hyperparameters using approaches such as grid search, random search, or Bayesian optimisation can improve model performance. Furthermore, exposing the models to a variety of datasets from various sources will help to assess their robustness and applicability across different demographic and clinical scenarios.

Finally, integrating ensemble techniques, which combine multiple ML algorithms, shows promise for improving model performance. Ensemble models can effectively mitigate individual algorithm biases and errors by combining the various perspectives and strengths of different algorithms. This collaborative approach not only promotes more robust predictions, but it also encourages complementary collaboration among diverse models, which has the potential to unlock novel insights and improve overall predictive accuracy across a variety of datasets and real-world applications.

8 Conclusion

This section concludes the research findings by summarising the performance of various models in identifying respiratory diseases from sound data. The efficacy of CNNs, ANNs, SVMs, and RFs has been tested rigorously. Each model has distinct strengths and limitations, providing valuable insights into the field of automated disease diagnosis and monitoring.

8.1 CNN & ANN

The CNN and ANN models performed admirably in identifying respiratory diseases from sound data. The CNN, with its ability to capture spatial hierarchies and extract detailed features, achieved **90%** accuracy. Meanwhile, the simpler ANN model achieved **92%** accuracy. Because of their high accuracy and robust generalisation capabilities, these models have significant potential for use in real-world applications such as automated disease diagnosis and monitoring.

8.2 SVM & RF

In comparison, the SVM and RF classifiers produced lower accuracy rates than the neural network models. The SVM, which used a radial kernel, achieved an accuracy of **70%**, while the RF model achieved **74%**. Despite their lower accuracy, SVMs and RFs can provide useful insights and act as baseline classifiers for respiratory sound analysis. Further refinement and optimisation of these models may improve their performance and broaden their clinical application.

9 References

- [1] R. Yamashita, et al., "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, pp. 611–629, 2018. [Online]. Available: <https://doi.org/10.1007/s13244-018-0639-9>
- [2] P. Purwono, A. Ma'arif, W. Rahmانيar, H. Imam, H.I.K. Fathurrahman, A. Frisky, and Q.M.U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," vol. 2, pp. 739–748, Jan. 2023. [Online]. Available: <https://doi.org/10.31763/ijrcs.v2i4.888>
- [3] M.S.B.M.M.P.W., "Research Paper on Basic of Artificial Neural Network," *IJRITCC*, vol. 2, no. 1, pp. 96–100, Jan. 2014.
- [4] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *European Journal of Gastroenterology & Hepatology*, vol. 19, no. 1, pp. 1046–1054, Jan. 2008. [Online]. Available: <https://doi.org/10.1097/MEG.0b013e3282f198a0>
- [5] Y. Zhang, "Support Vector Machine Classification Algorithm and Its Application," in *Information Computing and Applications*, C. Liu, L. Wang, and A. Yang, Eds. Berlin, Heidelberg: Springer, 2012, vol. 308, *Communications in Computer and Information Science*, pp. 27–34. [Online]. Available: https://doi.org/10.1007/978-3-642-34041-3_27
- [6] T. Evgeniou and M. Pontil, "Support Vector Machines: Theory and Applications," in *Proceedings of the 14th Annual Conference on Computational Learning Theory*, 2001, vol. 2049, pp. 249–257. [Online]. Available: https://doi.org/10.1007/3-540-44673-7_12
- [7] A. Cutler, et al., "Random Forests," in *Machine Learning*, vol. 45, pp. 157–176, Jan. 2011. [Online]. Available: https://doi.org/10.1007/978-1-4419-9326-7_5
- [8] G. Biau, "Analysis of a Random Forests Model," *Journal of Machine Learning Research*, vol. 13, pp. 1063–1095, Apr. 2012. [Online]. Available: <https://www.jmlr.org/papers/volume13/biau12a/biau12a.pdf>
- [9] R. J. Oweis, et al., "An alternative respiratory sounds classification system utilizing artificial neural networks," *Biomedical Journal*, vol. 38, no. 2, pp. 153–161, 2015. [Online]. Available: <https://doi.org/10.4103/2319-4170.137773>
- [10] I. Sen, et al., "Computerized diagnosis of respiratory disorders: SVM based classification of VAR model parameters of respiratory sounds," *Methods of Information in Medicine*, vol. 53, no. 4, pp. 291–295, 2014. [Online]. Available: <https://doi.org/10.3414/ME13-02-0041>
- [11] J P. Garcia-Mendez, et al., "Machine learning for automated classification of abnormal lung sounds obtained from public databases: A systematic review," *Bioengineering*, vol. 10, no. 10, p. 1155, 2023. [Online]. Available: <https://doi.org/10.3390/bioengineering10101155>
- [12] V. Shivakumar, "Classification of Respiratory Sounds" Stanford University, School of Medicine, Nov. 2019. [Online]. Available: https://cs230.stanford.edu/projects_fall_2019/reports/26260886.pdf
- [13] Aykanat, M., et al. "Classification of lung sounds using convolutional neural networks." *J Image Video Proc.*, vol. 65, no. 2017, 11 September 2017. [Online]. Available: <https://doi.org/10.1186/s13640-017-0213-2>
- [14] S. Balasubramanian and P. Rajadurai, "Machine Learning-Based Classification of Pulmonary Diseases through Real-Time Lung Sounds," *Int. J. Eng. Technol. Innov.*, vol. 14, no. 1, pp. 85–102, Jan. 2024.