RESPIRATORY DISEASE DETECTION FROM RESPIRATORY SOUND AND MEDICAL HISTORY USING DEEP LEARNING TECHNIQUES

HIWOT HABTAMU ADERAW

A Thesis Submitted to Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering (Artificial Intelligence)

Office of Graduate Studies

Adama Science and Technology University

OCT 11, 2021

ADAMA, ETHIOPIA

RESPIRATORY DISEASE DETECTION FROM RESPIRATORY SOUND AND MEDICAL HISTORY USING DEEP LEARNING TECHNIQUES

HIWOT HABTAMU ADERAW

ADVISOR: DR. MESFIN ABEBE

A Thesis Submitted to Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

OCTOBER 2021

ADAMA, ETHIOPIA

Declaration

I hereby declare that this Master Thesis entitled "**Respiratory Disease Prediction from Respiratory Sound and Medical History Using Deep Learning Techniques**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

_____	_____	_____
Name of the Student	Signature	Date

Recommendation

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**Respiratory Disease Prediction from Respiratory Sound and Medical History Using Deep Learning Techniques**" prepared under my/our guidance by Hiwot Habtamu submitted in partial fulfillment of the requirements for the degree of Mater's of Science in **Computer Science and Engineering**. Therefore, I/we recommend the submission of the revised version of the thesis to the department following the applicable procedures.

_____	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-advisor	Signature	Date

Approval Sheet

I/we, the advisors of the thesis entitled "**Respiratory Disease Prediction from Respiratory Sound and Medical History using Deep Learning Techniques**" and developed by Hiwot Habtamu, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Major Advisor	Signature	Date
Co-advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Hiwot Habtamu have read and evaluated the thesis entitled "**Respiratory Disease Prediction from Respiratory Sound and Medical History Using Deep Learning Techniques**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in **Computer Science and Engineering**.

Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

Above all, I would like to thank the Almighty GOD for his blessing and help throughout my life. Next, I would like to express the deepest gratitude to my primary Advisor, Dr. Mesfin Abebe, for his guidance and invaluable support in every part of the thesis. My wider acknowledgment goes to Dr. Solomon Damite (Medical Doctor), for helpful medical suggestions and unwavering support in everything I need for. My sincere thanks go to Dr. Abate Girma (Pulmonologist), for the relentless assistance and patience that won't be underestimated. Nurse Simenesh Aderaw and Nurse Hareg W/Gebreal deserve my appreciation for their willingness, guidance, suggestion, and contribution. I am extremely grateful for all of my family members, you have been and always be nurturing and supportive. Particular recognition to AI SIG members for insightful suggestions and constructive criticism. Finally, I would like to extend my special thanks to my friends for their appreciation and valuable support.

Table of Contents

LIST OF FIGURES

LIST OF CODE FRAGMENTS

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ALGORITHMS

LIST OF ACRONYMS

Acc	Accuracy
AI	Artificial Intelligence
ANN	Artificial Neural Networks
CAA	Computerized Auscultation Analyses
CNN	Convolutional Neural Network
Colab	Colaboratory
COPD	Common Obtrusive Pulmonary Disease
CPU	Central Processing Unit
CT scan	Computed Tomography scan
DNN	Deep Neural Network
GMM	Gaussian Mixture Models
GAN	Generative Adversarial Network
GFCC	Gammaton Mixture Model
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
ICBH	International Challenge on Bio Health Informatics
KNN	K-Nearest Neighbors
MFCC	Mel-Frequency Cepstral Coefficient
ML	Machine Learning
PR	Pulse Rate
RAM	Random Access Memory
RNN	Recurrent Neural Network
RR	Respiratory Rate
SOB	Shortness of Breath
Sp	Specificity
Spec-Aug	Spectrogram Augmentation
SPHMMC	St. Paul Hospital Millennium Medical College
SVM	Support Vector Machine
TPU	Tensor Processing Unit
VTLP	Vocal Tract Length Perturbation
WHO	World Health Organization

ABSTRACT

Respiratory diseases are infectious diseases that affect the respiratory organ, usually the lung of the patient. Oftentimes, investigation errors occur in respiratory disease diagnosis. To solve this, many researchers use different machine learning(ML) and Deep Learning(DL) techniques. But the previous works use respiratory sound as a single factor to predict respiratory disease. However, there is a significant overlap between the sound and symptoms of respiratory diseases. Here, this thesis aims in using patient medical history data along with respiratory sound to predict respiratory disease. For this purpose, we have collected patient symptom data from St. Paul Hospital and retrieved the respiratory sound data from ICBHI publicly available repository. Those datasets were used for pulmonary disease and respiratory sound classification in which, the target from respiratory sound is used as a feature to the pulmonary disease classification. To improve the respiratory sound classification model, feature Extraction MFCC and Mel-spectrogram along with augmentation techniques Time-shift, VTLP, and Spec-Aug have experimented with a CNN model. Towards pulmonary pathosis, the DNN model is used for classification. And the model has been evaluated with 5-fold cross-validation. Through experiments, respiratory sound classifier model gives an accuracy of 73.15% without Spec-Aug augmentation. From these experiments, we have observed that Mel-spectrogram features give a good accuracy furthermore when we integrate it with the MFCC feature it gives us a better result. Also the augmentation techniques Time-shift and VTLP have provided us fine results while on the contrary, Spec-Aug augmentation technique gives modest accuracy, when in fact Spec-Aug has a much better speed of execution than other augmentation techniques. Whereas the pulmonary pathosis classifier gives 97% accuracy. Altogether the respiratory disease prediction performance has been evaluated with the help of a health expert, and the result obtained shows that our work correctly classifies 7 out of 10 subjects. Eventually, our work revealed that using patient symptoms along with respiratory sound enriches respiratory disease prediction. In addition, respiratory sound classification can benefit from Mel-spectrogram features; apart from that, the Spec-Aug augmentation technique is unsuitable.

Key words: *Respiratory Diseases, Respiratory sound, Artificial Intelligence, Deep Learning, Feature Extraction, Data Augmentation, Disease Prediction.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

In the early 18th century physicians use only eyes, hands, nose, ears, and handmade stethoscopes to diagnosis a patient But starting from the age of scientific discovery, a lot of advancements and development in technology are being used on the application of earlier patient diagnosing (Mayorga et al., 2010). The advancement in technology includes computing solutions and Artificial Intelligence(AI).

The significance of Artificial intelligence in medicine and Medicare is arising exponentially. In 2016, AI projects working with medicine drew in more speculation from the global economy than other projects (Mekov et al., 2020). In medicine, AI is referred to as the exertion of automated treatment and diagnosis. Increasing the use of AI allows many tasks to be performed automatically, which gives time for medical professionals to perform a variety of tasks, that they cannot be done automatically (Geddes, 2016). However, researchers are still digging into the potential of AI. One of the research areas that keeps AI researchers in Medicine is Respiratory Disease Diagnosing and Prediction.

Respiratory diseases are life-threatening diseases that attack the respiratory organ usually the lung of the patient. Those diseases are among the most ordinary causes of severe illness and death worldwide, furthermore, the problem is much higher in growing countries such as Ethiopia. Annually respiratory diseases cause about 3 million deaths around the world (Chambres et al., 2018). The situation is even getting terrible each year. In addition to this, the number of doctors to that of patients is incomparable, the ratio is 1 Doctor to 17,000 patients[1]. It is hardly the patients could get the amount of time they need from the doctors. Lack of medical resources in developing countries makes the problem even much higher. Therefore, using technologies like AI have a positive impact.

According to WHO 'the big five' respiratory diseases are COPD (common obtrusive pulmonary disease), Asthma, Acute lower respiratory tract infection, Tuberculosis, and Lung cancer (Edition,

[1] https://www.who.int/data/gho/data/indicators/indicator-details/GHO/medical-doctors-(per-10-000-population)

2019). In the case of Ethiopian statistics besides the burden from COPD and Asthma, lower respiratory infections such as pneumonia and Bronchiolitis hold a major dead list (Anbesse et al., 2020). Therefore, this thesis focuses on those four respiratory diseases; Asthma, Bronchiolitis, COPD and pneumonia.

By their nature, respiratory diseases can be curable or non-curable. But if the patient having a respiratory disease is diagnosed early in its acute level, there is a huge chance that curable diseases could be cured and non-curable diseases can be treated so that the patient won't suffer any further. To diagnose respiratory diseases in their earlier stage, respiratory sounds and other symptoms play a major role.

At the clinics, the conventional ways to check for respiratory diseases include auscultation, bronchoscopy, spirometry test, chest CT scan, chest x-ray, and routine sputum culture, where the most simple and direct method is auscultation, in which a stethoscope is used to check for Pulmonary sounds (Chambres et al., 2018). Pulmonary sounds are audible vibrations that occur in the lungs and airways (Gavriely et al., 1994). Normal pulmonary sounds are pulmonary sounds associated with the breathing of a normal person whereas pathological sounds also known as adventitious sounds are the indicators of a person with some respiratory diseases.

Adventitious respiratory sounds may present before changes in radiology examination, by measuring lung function and recording symptoms, which indicates the value of auscultation and patient symptom monitoring in respiratory disease detection. Studies were performed on the capability of the human ear to detect the crackle signal in auscultation; according to the results, the most common detection errors are caused by the intensity of the respiratory sound signal. Deep breaths mask more adventitious sound than superficial breaths and the similarity between the waveform and amplitude of some lung sounds. Considering this study, auscultations performed using the human ear can be supplemented using AI(Reichert et al., 2008).

Machine learning techniques are integrated with adventitious sound detection for much better and faster results. Nowadays Researchers are using different Machine and deep learning algorithms include empirical rule-based methods, support vector machines (SVMs), artificial neural networks (ANNs), Gaussian mixture models (GMMs), k-nearest neighbors (k-Nns), Convolutional Neural Network (CNN) (B. M. Rocha et al., 2019), and logistic regression models and more advanced

machine learning techniques. In this paper the potential of CNN, DNN are exploited for the exact purpose of predicting respiratory disease.

1.2. Motivation of the study

Undoubtedly health is a pillar for life, and normally healthy peoples are productive. Therefore, keeping human beings safe is a very crucial task. Worldwide respiratory diseases are by far the most killer disease. In growing countries like Ethiopia, the problem is even much higher. In lung disease investigation many miss diagnosis situations happen due to many reasons; here is a real-life scenario:

> *A 31 years old female has been suffering from respiratory disease due to being misdiagnosed to COPD While she has Asthma she starts taking capsules, but the capsules were working adversely; 3 years passed and the problem even gets much higher, after finally she sees a specialized doctor from abroad and gets diagnosed and recovered. If such things happen in countries capital where many hospitals and doctors are available, it is so imaginable how the problem will arise in rural areas where no many health centers, a lack of medicines, and a very little number of doctors.*

Therefore, minimizing such miss diagnosis situations and decreasing the number of people dying from lack of proper respiratory disease diagnosis is an ideal solution. It is highly motivating to make a solution for health experts in better decision-making.

1.3. Statement of the Problem

Digital data such as image sound and live moving images (video) of the patient are being widely used in medical checkups as a result of the advancement in medical diagnosis. Nevertheless, the inspection and analysis is done by doctors. Examination of a general practitioner to that of a specialist and sub-specialist doctor cannot be exactly the same, this is due to the fact that patient diagnosis is highly dependent on the experience and heuristic capability of the physician. As a result, miss and under-diagnosis happen. To conquer this, researchers made machine learning models for better decision-making. Those models were built using respiratory sound as a single factor. But the patient's symptoms and respiratory sounds have significant overlap between many

respiratory diseases. This overlap makes the task of respiratory disease identification difficult by resulting in miss and under-diagnosis. Let us see two basic cases of overlap.

> **Case 1:** A person with a "Wheeze" sound can have Asthma or COPD(Mayorga et al., 2010).
>
> **Case 2:** A person with a "Normal" breathing sound can be health or can have respiratory disease such as upper respiratory infection(Chambres et al., 2018).

Considering the two cases, depending only on the sound data gives confusion, besides that as stated by Myorga et al,2010; adventitious respiratory sounds alone are far from being an adequate element for diagnosis. Therefore, referring to patient's medical history gives a chance of getting a quality diagnosis.

Also, the features used in respiratory sound prediction models are classical frequency and MFCC features but those features fail to classify all adventitious sounds especially for crackle wheeze and crackle class. While a time-frequency feature Mel-spectrogram is recommended. The available respiratory sound datasets are not resource-rich and the classes are not balanced, hence they need augmentation techniques. Luckily, a relatively new augmentation technique Spec-Aug performs well for sound augmentation.

Knowing the above problems, this thesis aims to improve the respiratory sound prediction by applying Mel-spectrogram features and Spec-Aug augmentation techniques. Consequently, respiratory disease detection using sound analysis in integration with the medical history of the patient is intended.

1.4. Research Questions

This study answers the following research questions to address the problem specified above:

RQ-1: What Percentage of enhancement do Mel-spectrogram features give for respiratory sound prediction?

RQ-2: What is the effect of the Spec-Aug augmentation technique in respiratory sound prediction?

RQ-3: Can respiratory disease prediction be improved using the patient's medical history and breathing sound?

1.5. Objectives of the Study

1.5.1. General Objective

The General Objective of this study is to develop a respiratory disease prediction model using respiratory sound analysis and patient symptoms.

1.5.2. Specific Objective

To meet the general objective of the study; the following specific objectives have been established:

- To collect, prepare and analyze patient medical history data.
- To prepare the obtained ICBHI respiratory sound dataset.
- To experiment the proposed augmentation and feature extraction methods.
- To design the general architecture and flow of respiratory disease prediction.
- To compose model for respiratory sound and pulmonary disease classification.
- To evaluate and analyze the performance of the model.
- To develop and deploy a prototype of respiratory disease prediction in edge device.

1.6. Scope and Limitations of the Study

1.6.1. Scope of the study

This study concerns respiratory disease detection using adventitious sound analysis and patient's medical history to yield a better result. Convolutional and deep neural network models were utilized for the classification of adventitious sound and respiratory disease respectively. For adventitious sound classification, the study covers pre-processing, augmentation, and feature extraction on the already available respiratory sound dataset. But for respiratory disease prediction, there were no patient's medical history data available; therefore, this work shields the dataset collection, preparation and pre-processing, and eventually the diagnosis prediction.

1.6.2. Limitation of the Study

This work does not concern with the sub-division of each disease. For instance, pneumonia can be branched into hospital-acquired, community-acquired, viral, aspiration, fungal, atypical, and so on. Additionally, at hospitals patient signs, symptoms, and history is written in paragraph format, but for the ease of the study, we have work on the structured format of patient symptom data. Therefore, patient history in form of natural language cannot be handle by this work.

1.7. Significance of the Study

With some improvement and full integration of the model, this work can help in minimizing miss diagnosis. But it does not say that the work can be standalone or fully replace practitioners. However, it can help in better decision-making. In addition to this, the work can be a great input for scientists who are engaged with developing a humanoid medical robot to serve and diagnose patients. AI, machine learning, and deep learning algorithms are growing instantly almost in all sectors, for those who are passionate about health care could use this paper finding as a milestone, for further research and investigation.

1.8. Organization of the Thesis

The thesis is comprised of 6 chapters along with a conclusion and future work. The first Chapter mainly discussed the introductory, background of the study, statement of the problem and research questions to be answered, objectives, scope, and limitations of the study alongside significant of the study. The second Chapter deals with the literature review on respiratory diseases, algorithms used for respiratory disease classification, and what is already known in the state of art also what researchers have done already, and the gaps as related work. The third Chapter elaborates the methodologies, tools, and techniques used in this work. The fourth Chapter shows the architecture workflow and design of the proposed solution. The proposed solution designed in chapter 4 is shown as to how it is implemented using actual code in the fifth Chapter. Following this, the last Chapter sees the result obtained from the experiments and discussed how the research questions have been answered through comparisons of the work with previous works. Finally, a conclusion and future work are provided.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Introduction

The human body needs energy for all metabolic activities in each cell. Most of this energy is derived from chemical reactions, which can only take place in the presence of oxygen (O_2). Respiratory organs use muscles to inhale oxygen and exhale carbon dioxide by using airways to and away from the lung(Gavriely et al., 1994). The respiratory system is studied as the upper respiratory system and the lower respiratory system.

The upper respiratory tract enables smell and speech, also ensures that air passing to the lower respiratory tract is warm, moist, and clean. The tract contains the mouth or oral cavity, the nose (nasal cavity), the pharynx, and the voice-box(Health, 2015)(Peate & Studies, 2018).

The trachea, the right, and left primary bronchi, and all the lungs comprise the lower respiratory tract. The windpipe (trachea), a tubular vessel, takes air from the larynx towards the lungs. Any inhaled debris is trapped and pushed upwards towards the esophagus and pharynx, to be either swallowed or discharge. The major passages and structures of the lower respiratory tract include the **windpipe** (**trachea**) and within the lungs, the **bronchi**, **bronchioles**, and **alveoli**(Mason, 2017)(Peate & Studies, 2018)**.** All the respiratory organs are very useful for breathing.

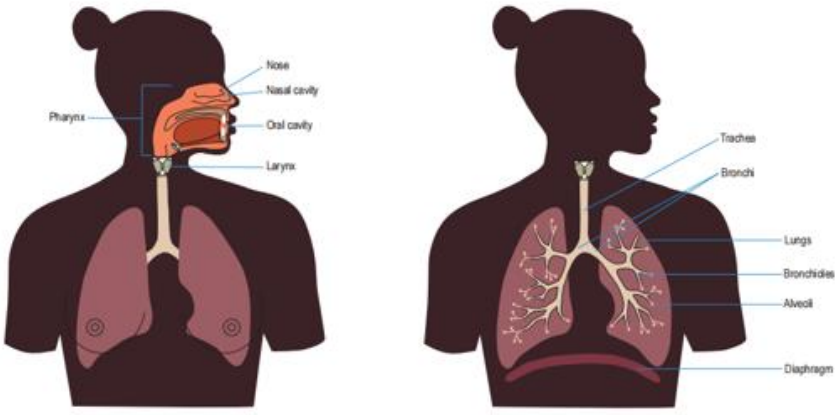


Figure 1:- Upper and Lower Respiratory system[2]

[2] https://www.magonlinelibrary.com/doi/respiratorySystem

2.1.1. Breathing

Breathing also known as pulmonary ventilation is a complex metabolic activity. Breathing is important for life and normal people cannot stay without oxygen for more than 3 minutes. because every cell in the human body needs oxygen to survive and grow. When a person breathes, the chest expands and the pressure in the lungs drops below atmospheric pressure, and air naturally enters the airways until the pressure difference disappears. The mode pulmonary ventilation rate is 12-16 breaths per minute. Each breath consists of three phases: inspiration, expiration, and pause. If any of the breathing processes are being either distracted or being difficult for inspiration and expiration, there is a full chance that respiratory diseases are present(Health, 2015).

A person having distracted breathing can experience adventitious respiratory sounds. Adventitious respiratory sounds are abnormal breathing sounds due to blockage or inflammation of respiratory organs(B. Rocha et al., 2017). The most common adventitious respiratory sounds are explained below.

***Wheezing*:** is a loud whistle. It sounds very clear when exhaling, but it can also be heard when inhaling in serious cases. It is caused by narrow airways or swelling. Wheezing is a sign of acute or chronic respiratory problems. Asthma and chronic obstructive pulmonary disease (COPD) are the most common causes of wheezing; it can also be an indicator of other respiratory and heart diseases(Reichert et al., 2008).

Crackle (Rales): is the sound that occurs when small air bubbles pass through the liquid. This could indicate a build-up of mucus or pus in the airways.(Messner et al., 2018) They can indicate fluid in the small airways.

> ***Fine crackle:*** is a crackle heard when a person is affected by pneumonia or heart failure.
> ***Coarse crackles:*** have a deeper and lengthy sound compared to fine crackles. Larger airways filled with fluid give this sound when air is entered through. COPD patients are most likely to have coarse crackle.
> ***Biphasic crackle:*** a two-phase crackling is a combination of fine and coarse crackling.

Stridor: is similar to wheezing, but the sound is usually louder than the wheezing; It can be identified by inhaling or exhaling, or both, and can identify an obstruction or narrowing of the upper airway. It means there is a narrowing or obstruction in the voice box called the larynx. If the

sound of exhaling can be heard, it means that the trachea or windpipe is narrowing. The trachea is the tube that connects the throat to the lungs(Pham et al., 1864).

Whooping: is an adventitious sound that can occur while coughing, when a person gasps for air. It's the high-pitched sound of air entering the airways. This sound is often heard with some phenomena, also known as whooping cough. Whooping cough is caused by bacteria. A patient with COPD has an increased risk of developing whooping cough (Pham et al., 2014).

Pleural rubbing: The lungs and the pulmonary cavity are covered by thin membranes called pleurae, which usually slide gently over each other to regulate breathing. when something is disrupting the membranes a pleural friction rub of explosive sound can be heard(B. Rocha et al., 2017).

2.2. Respiratory Diseases

Respiratory diseases are a major type of disease that usually affects the respiratory organ of the patient. Respiratory disease can range from acute to chronic(Peate & Studies, 2018). As this thesis is restricted to asthma, COPD, Pneumonia, and bronchiolitis, they are discussed as follows.

2.2.1. Asthma

Asthma is a condition in which the airways narrow and swell. The symptoms of asthma vary from one person to another, both in frequency and in severity. The symptoms include cough, wheezing or whistles, dyspnea or shortness of breath, Chest oppression**,** nasal symptoms such as irritation, sneezing, and blockage(Mortality & Collaborators, 2013). Several risk factors associated with asthma are allergy, genetic predisposition, tobacco use, exposure to environmental and occupational pollutions.

2.2.2. Chronic Obstructive Pulmonary Diseases

Chronic obstructive pulmonary disease (COPD) is a chronic incendiary lung disease that causes deterrent airflow from the lungs. Symptoms of COPD include fast breathing, chest pain, wheezing, a chronic cough with mucus(Badnjevic et al., 2018). Cigarette smoking is the leading cause of COPD additionally, fumes from burning fuel for cooking and heating in poorly air-conditioned homes is also a factor(Health, 2015). Second-hand tobacco smoke, also outdoor air pollutants, allergens, occupational agents, genetic factors, and aging are factors contributing to the occurrence of COPD(Gavriely et al., 1994).

2.2.3. Bronchiolitis

Bronchiolitis is an inflammatory bronchial reaction in young children and infants that is almost always caused by a virus. The first symptoms of Bronchiolitis include runny nose, congestion, cough, and mild fever(Peate & Studies, 2018). A few days later, the cough worsens and breathing gets faster, wheezing or crackling sounds when breathing out, widening nostrils, and grunting, sometimes no breathing, mal-nutrition, blue skin color around lips or fingertips(Friedman et al., 2014). Factors that raise the risk of getting bronchiolitis are schools, air pollution exposure, secondhand smoke, cardiopulmonary disease, weak immune systems, weakness of respiratory muscles, never having been breast-feds(Health, 2015).

2.2.4. Pneumonia

It is an infection in one or both lungs. The disease causes swelling in the air sacs of the lungs. Symptoms of pneumonia include fever, coldness, perspiration, crackle, cough, chest pain, shortness of breath(Pham et al., 2014). Severe symptoms that may occur in adults include the bluish color of the nails and Mental confusion. Severe symptoms that may occur in babies and young children: flaring nostrils with each breath, chest or ribs may suck in with each breath, and grunting sound with breathing. In addition to these on chest auscultation, crackles or crepitation sound heard. Adults over age 65 and children under age 2 are mostly affected with pneumonia**.** Other risk factor includes**,** being in the hospital, weakened immune system, being on a mechanical ventilator, having a chronic disease like COPD and heart disease, difficulty swallowing as a result stroke and malnutrition(Peate & Studies, 2018).

2.3. Methods of Respiratory Diseases Investigation

Physical examination is important because it reveals the presence of an area of swelling or blockage of an airway in the chest. The examinations include physical inspection and finding for masses, sensitive areas, anomalous breathing patterns, and auscultation or listening with a stethoscope to determine tone and loudness of breath sounds. Those sounds detected with auscultation may reveal deformity of the airways and the lung tissue. Mostly a healthcare professional hears breath sounds with a normal or electronic stethoscope(Mekov et al., 2020)(Gibson et al., 2013). Rarely, if the case is severe; the sound can be detected even without a stethoscope. For further investigation, the medical staff might call for other tests such as:

- ***Sputum Test*:** this examination of the sputum for bacteria helps in the identification of infectious organisms that needs a specific treatment; malignant cells can benefit from this sputum test.
- ***Spirometry:*** A device called a spirometer measures the flow of air in and out of the lungs also the volume of the air is estimated.
- ***Laryngoscopy:*** Larynx or voice box is explored using a small scoping device.
- ***Bronchoscopy:*** Similar to laryngoscopy, bronchoscopy provides imagining by exploring deeper into the lungs.
- ***Chest X-ray:*** In X-ray radiation is used to perceive a picture of the lungs. Any damage and inflammation of small air sacs can be checked easily.
- ***CT scan:*** More like an X-ray, but a CT scan provides detailed information about the lung.

Although all the above methods are very helpful the most straightforward method is Auscultation.

2.3.1. Auscultation

Sounds provoked from inside the body can be heard using a stethoscope, this mechanism is referred to as auscultation. A vast number of medical conditions can be diagnosed with the help of auscultation. Usually, the lung, cardiac, abdomen, and blood vessel sounds are investigated with auscultation. Anterior and posterior chest locations are the most commonly used areas for auscultation(Gavriely et al., 1994).

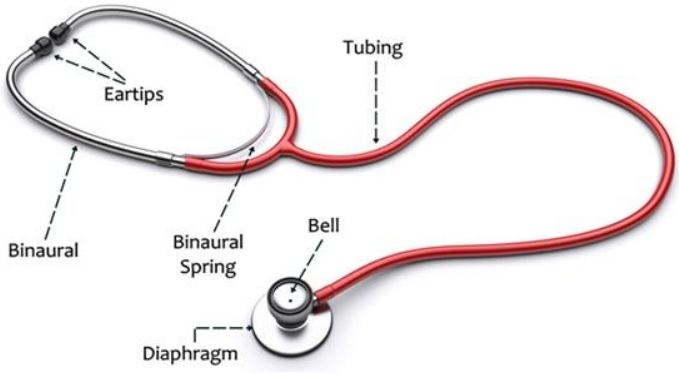


Figure 2- Stethoscope[3]

[3] https://www.nurselly.com/parts-of-a-stethoscope

The stethoscope consists of two parts, a diaphragm to listen to high-pitched sounds while the bell is for low-pitched sounds. Quite places are suitable to perform auscultation because the expert is able to perceive the sounds distinctly but auscultation using a stethoscope has some drawbacks.

2.3.1.1. Down Sides of Auscultation

When trying to detect and determine abnormal sounds it is a must that auscultation is performed by a medical expert. This is very hindering, the number of people capable of perfectly performing auscultation are limited. A lot of experience is needed to be an expert in auscultation. Determining the types of adventitious sounds heard for diagnosis reasoning can help inpatient monitoring. But unexperienced determination results in miss diagnosis due to underestimation and omission of symptoms(B. M. Rocha et al., 2019).

In addition to the investigation mechanisms mentioned above, the development of algorithms to identify or classify respiratory sounds has been a basis of recent research works. These works developed detection or classification methods by extracting certain features from the sounds and patient symptoms. The detection and classification methods used vary from empirically determined to the use of machine learning.

2.4. Machine Learning Algorithms

Machine learning is an inspection of algorithms that are mainly used for statistical models. One of the great thing about machine learning algorithms is that they are capable of automatically giving answers because they learn from data so that there is no need for explicit programming. Machine learning algorithms can fall into:

- ***Supervised:*** are machine learning algorithms that learn through experience. Those algorithms are mainly used for classification and regression problem types.
- ***Unsupervised:*** unlike supervised learning algorithms, unsupervised learning algorithms learn from data itself. Density estimation and clustering can highly benefit from the algorithm**.**
- ***Semi-supervised:*** this algorithm is a combination of both supervised and unsupervised machine learning techniques. In which the dataset is a mixture of data with examples and raw data.

- ***Reinforcement:*** it more resembles the animal's reflex action system. In this algorithm conditions known as environment and agents take part. Where the environment gives feedback whenever the agent gives the wrong answer, and it keeps this trend until the agent gets the answer right.

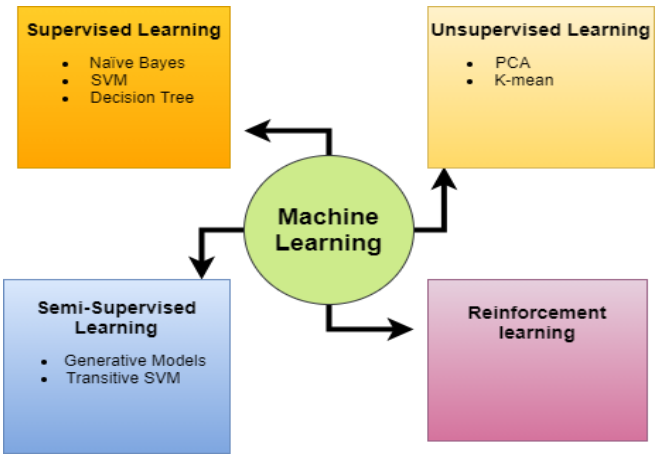


Figure 3- Categorization of Machine learning algorithms

2.5. Supervised Algorithms for Classification

Supervised learning is a learning pattern that acquires a relationship between the dependent features(output) and the independent features(input) through precedent samples. This learning mechanism is also known as inductive learning(Q. Liu & Wu, 2015). Classification is a major task in supervised learning algorithms(Rodriguez, 2020). There are a lot of researches on disease classification and also the way of classification is using the supervised method(Uddin et al., 2019). The classification can be achieved through machine learning or deep learning algorithms. The main difference between Machine learning and deep learning models is that; machine learning models are heavily dependent on well-defined features(Janiesch & Heinrich, 2021). Those features are extracted by other extractors. Whereas in Deep learning algorithms the model itself learns from the data pattern and it uses this pattern as a feature, this capacity of feature extraction is known as deep feature extraction. The following picture illustrates the idea.

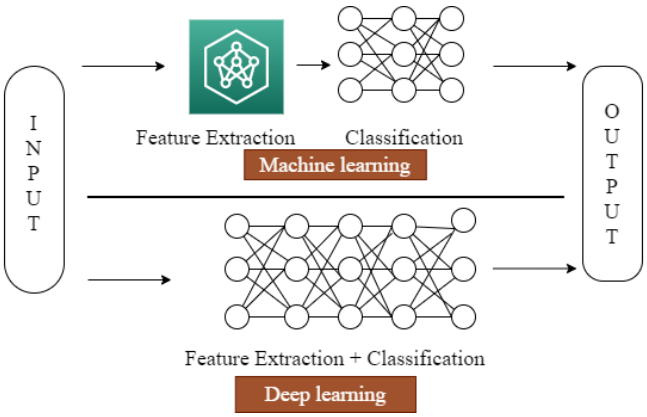


Figure 4- Overview of Machine Learning and Deep Learning

2.5.1. Machine Learning for Classification

Machine learning(ML) benefits the health expert and practitioners by determining multiplex models to extract knowledge in the medical sector. In individual patient supervision and care, ML models anticipate rules to help in decision-making (Mekov et al., 2020). After all the pre-processing including augmentation and feature extraction the data is used as an input to the classification algorithm and therefore the classification is held. But all the pre-processed data should not be fed at once because the model's performance should be measured so that some data should be kept as a test dataset. For sound and structured data classification, a lot of ML algorithms have been used(Demir, Sengur, et al., 2020). The most frequently used and the most impactful machine learning algorithms are discussed as follows.

2.5.1.1. Gaussian Mixture Model(GMM)

It is a statistical model in which a mixture of Gaussian random variables is used on the audio feature for analysis or synthesis. Mostly GMM is used in Fourier-based spectrums and a wide range of speech features. The most unbeatable usage of GMM is that for audio synthesis or speech formulation because it is relatively easy to feed the data so that the evaluation speed is maximized. But one of the shortcomings of the model is that in a non-linear mode of the data fold it is inefficient(Mayorga et al., 2010). For respiratory sound detection, it has been used but its performance has been outdated by other machine and deep learning algorithms.

2.5.1.2. Hidden Markov's model(HMM)

Starting from the classical eras of scientific research, HMM has been used. Fundamentally HMM is based on a theory to determine the series or best sequence of a model in states. It also includes the adjustment in the parameters. To obtain the best sequence of observations in time. In speech and audio recognition and processing HMM plays a vital role. Mostly in speech recognition and formulation has a major role. In respiratory sound prediction, HMM has been used as an improvement to that of GMM models. Hand-crafted features of the audio are used as an input to the HMM model and the model predicts the best combination of variables or sequences in the signal, hence it classifies the audio(Grewal et al., 2019). Nevertheless, HMM has been replaced by the state of the art Deep learning algorithms.

2.5.1.3. Logistic Regression (LR)

It is a predictive or classification model where the input features are categorized into the target class. Since many disease prediction models are meant to classify a person as healthy or not, they fall into binary class classification. Hereafter, the most widely used binary Logistic Regression model is exploited. A value of 1 for healthy and 0 for sick people. Besides this, logistic regression is still powerful for multi-class prediction, but it works best when the input is categorical data.

2.5.1.4. K- Nearest Neighbor (K-NN)

KNN is known as a lazy learner algorithm, because of its instance-based learning scheme. The algorithm is really sensitive to the structure of the data and its location so that it decides the classification based on the neighbors. Votes under the neighbors of the near element are calculated and it is a major point of classifications decision. This all procedure is dependent on the number of K values to consider; hence K is the number of neighbors to accept the vote from. This classification model has been used for disease classification, but this day's other algorithms are more preferable.

2.5.1.5. Support Vector Machine(SVM)

In SVM the data is categorized into two with a hyperplane of N-dimensions. The algorithm is more related to the multilayer perceptron of neural networks when sigmoid kernel is used. The algorithm can even be used as an alternative training model for multilayer perceptron, radian basis, and polynomial functions. Unlike standard neural network training that uses non-convex and

unconstrained minimization, SVM uses quadratic programming with a linear constant for solving the weights of the network. Because of the SVM performance still, disease prediction is highly implemented and archives astonishing accuracy.

2.5.2. Deep Learning for Classification

Inspired by the human brain, deep learning algorithms perform undoubtedly for classification problems. Deep learning extracts complex and high-level abstractions as a depiction of data through deterministic learning(Wehle, 2017). The invention of using spectral representations such as spectrogram paves a way in shifting to deep learning algorithms for audio classification. The deep learning ability in deep feature extraction makes it more suitable for temporal data classification. One of the areas that benefit from this is respiratory sound detection. Among other deep learning algorithms, Convolutional Neural Network and Deep Neural Network are the most widely used.

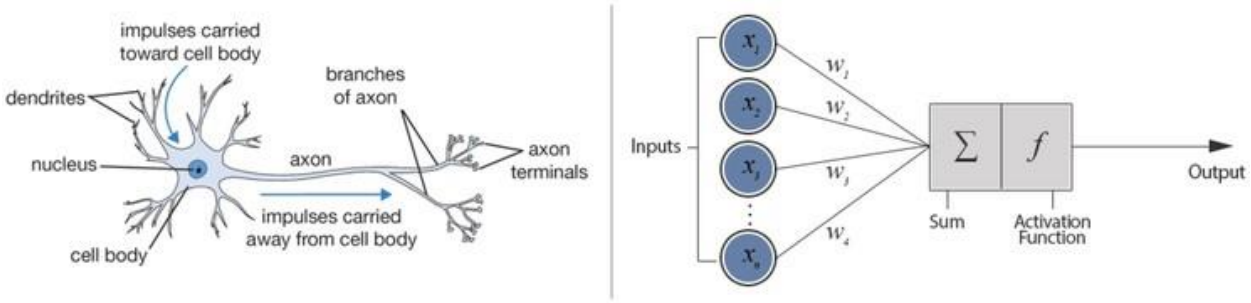


Figure 5- Neural Network[4]

Convolutional Neural Network*:* Firstly, CNN algorithms were evolved for image classification and analysis. But after the use and usage of image-like representation of the audio signal, CNN becomes a major algorithm in audio classification, image-like representations such as spectrogram, Mel-spectrogram, and MFCC. CNN model convolves each input then down sample and eventually makes a decision based on the dense layer at the end. But all the operations in CNN cannot fit the temporal audio data, because images are static and audio changes through time. Therefore, researchers customized the CNN model for each audio analysis task. In respiratory sound prediction, it is still the state of the art.

[4] https://www.azati.ai/disease-prediction-and-classification-with-neural-networks

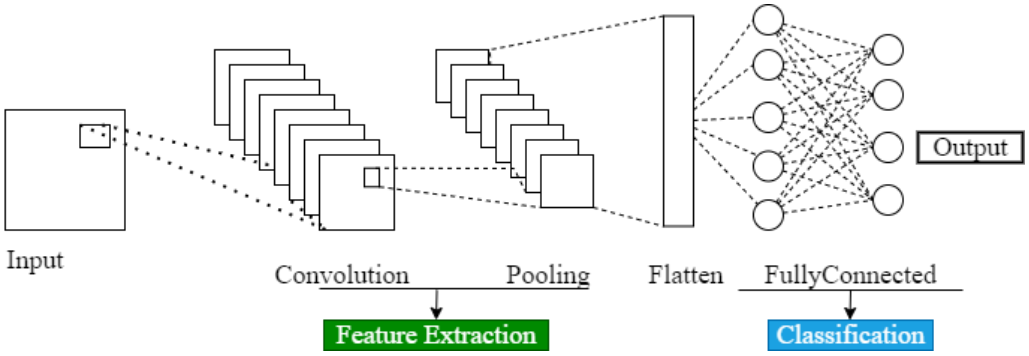


Figure 6- Convolutional Neural Network

Deep Neural Network: The ability of deep learning to learn from patterns of the input makes the algorithm more powerful than other machine learning algorithms. DNNs are neural networks aspired by the working scheme of the animal brain. The input mimics the input coming from the sense organs. And the hidden layer is more like the function of the central brain system. Deep Neural Networks are like the ANN but DNN can model more complex non-linear connections. Hence the name Deep, deep neural network architectures have a deep neural structure, have more deep neurons than ANN. The working of Deep Neural Networks in prediction is demonstrated as follows.

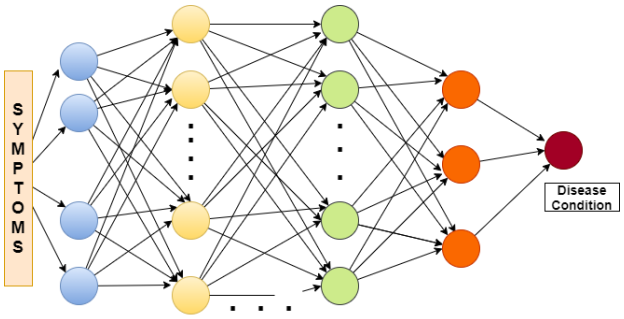


Figure 7- Deep Neural Network for Disease Prediction

2.5.3. Sound Data Augmentation

Sound augmentation is relatively new to that of the corresponding image augmentation. In image augmentation cropping the image, blurring or rotating the image can work fine as augmentation. But in audio augmentation, since it is temporal data such image augmentation techniques cannot work properly even in most powerful deep learning techniques(Madhu, 2019)(Iwana & Uchida, 2020). Here are some of the most widely used audio data augmentation techniques:

- ***Noise addition:*** Noise can be added to the input signal, activation function, gradients, or even to the weights. The noise added is usually Gaussian noise and it helps in augmenting through smoothing the input.
- ***Time Stretching:*** in time stretch the input signal is stretched up or stretched down using the stretching factor that means, which augments the signal by speeding up or slowing down the signal.
- ***Pitch shifting:*** in complementary to time-stretch, pitch shifting shifts or alters the pitch without changing the time of the signal.
- ***Spectrogram Augmentation:*** is the augmentation technique applied directly to the spectrogram feature of the audio.

2.5.4. Feature Extraction Methods

In the last decade detection and classification of adventitious respiratory sounds has been a major topic in many kinds of research. As in any other supervised machine learning classification trends; in adventitious respiratory sound classification, two major steps are included. The first and vital step is feature extraction, in which major audio features are extracted. Those extracted features are being used in the second step, which is the classification step. The most commonly used audio features extraction techniques and feature selection for structured data is explained as follows.

2.5.4.1. Sound Feature Extraction

Audio features extraction techniques used for classification can be sub-divided into time-domain, frequency domain, and time-frequency domain(Sharma et al., 2020).

1. **Time Domain Features**

   Time-domain features are audio features extracted by assuming the stationary behavior of the signal. The signal amplitude is used to calculate time-domain features. Majorly used time-domain features are; Mean it finds out the mean of the audio amplitude over the length of the signal, Variance measures variability using a squared average difference of the mean in the audio signal distribution. Standard Deviation is the square root of the variance. Mean Absolute Deviation averages the data points in their mean value deviation. Waveform length measures the accumulative length of the audio signal waveform in the time division. Zero crossing represents the frequency information of the signal over some time. Slope

Sign Change is mainly used for noise removal since this feature helps in getting the number of times that the signal changes its slope(Pham et al., 2014).

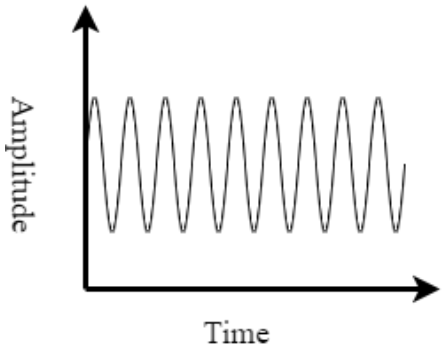


Figure 8- Time Domain Representation

2. **Frequency Domain Features**

Unlike time-domain features, frequency domain features mainly focus on the frequency of the signal. For frequency-domain features Power Spectral Density (PSD) is a majorly used extraction method. Some of the most common frequency domain features are; Linear Predictive Coding(LPC) it uses a linear predictive model to represent the spectral envelope of the audio signal and Peak Frequency(PF) approximates the most dominant peak frequency to calculate the underlying frequency.

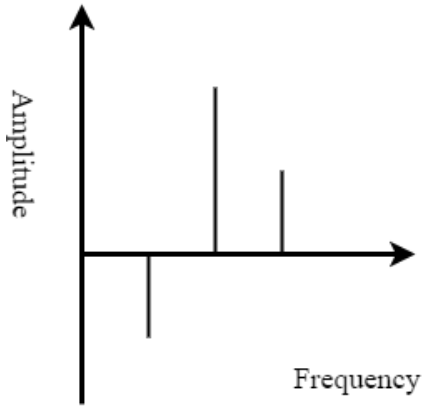


Figure 9- Frequency Domain Representation

3. **Time-frequency domain Features**

Time-frequency domain features aggregate the time and frequency behavior of the signal. Shor Time Fourier Transform(STFT) is mainly used to obtain time-frequency feature of

the signal. Spectrogram, constant Q-transfer, and Mel-spectrogram are the outputs from time-frequency domain transformation, and they are used to visualize the audio signal.

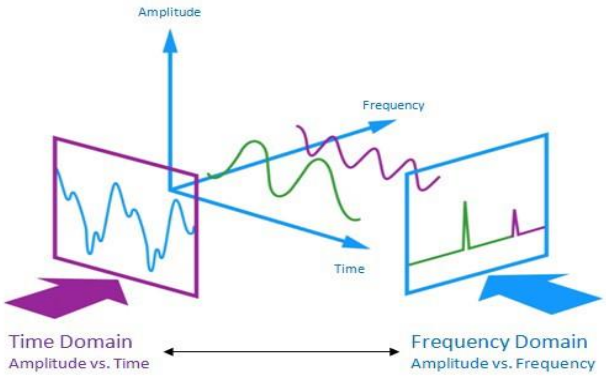


Figure 10- Time-Frequency Domain Representation[5]

2.5.4.2. Structured Data Feature Extraction and Selection

Before applying the classification algorithms extracting and selecting the features is very important and indeed very crucial. In this section, we have shown the most widely used feature selection methods for disease classification.

- **Extra Trees Classifier**

  To select the most important features among all the provided features. The Extra Trees classifier uses an ensemble tree method to observe the importance of each feature by randomly splitting all the observations.

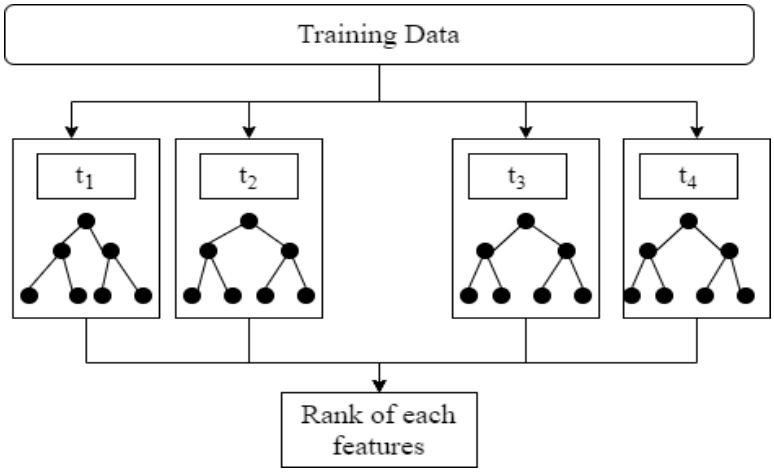


Figure 11- Extra Tree Classifier

[5] https://www. towardsdatascience.com/extract-features-of-music

❖ **Step Forward and Backward Feature Selection**

This feature selection method selects features by stepping forward and backward among the features. Whenever the feature is added to the model, the model's performance is calculated, if it has the highest score it is added to the model otherwise discarded. This is a step forward selection method; the reverse is a step backward selection method. Unlike the Extra trees classifier, this selection method is model-specific and hence works best with a random forest algorithm.

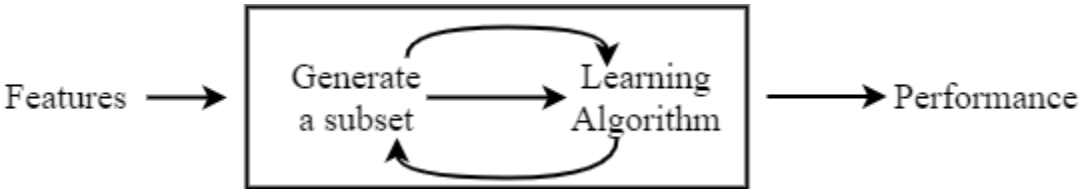


Figure 12- Step Forward and Backward Feature Selection

2.6. Related Work on Respiratory Disease Prediction

Some researchers have done researches on respiratory disease prediction. Among them, we have reviewed the papers that are believed to have a strong relationship with our thesis.

Gaëtan Chambers et al.(Chambres et al., 2018) rises the problem of using classification results based on sound content or micro-level to take a decision at the patient which is macro-level suggesting the patient sound sick or healthy. Authors extract 13 MFCC features of the respiratory sound using a library called essential. The extracted feature is used as an input to the Support Vector Machine (SVM) classifier. They also made two classifications the first classification is a multi-class classifier whereas the second classification is a binary class classifier that classifies a person as being healthy or not. The binary class classifier works good but multi-class classification fails especially for crackle sound. Additionally, the authors use the ICBHI dataset as it is, but since the dataset is imbalanced it leads to bad prediction. The best score during the test phase is Score = 49.86% while the method gave better results on the training phase. The closest results obtained during the train phase : Se = 0.4890, Sp = 0.7780 and Acc = 49.98%. Finally, the authors point out a very fundamental problem in predicting a respiratory disease, that is for some patients having normal respiratory sound making a decision on the type of disease is very difficult which needs some other information for better decision making.

N. Jakovljević and T. Lončar-Turukalo(Grewal et al., 2019) This paper aim to determine the respiratory sound as normal, wheeze or crackle. To achieve this, they have used ICBHI dataset. Mainly based on the legend work in GMM (Gaussian Mixture Model) by Mohammed Bahouran,2009, also authors aim to improve the model using Hiddne Markoves Model(HMM) technique. One of the great things about this paper is that they have used a noise filter for noise suppression. Band pass filter and High pass filter are the noise filter methods that they used. After the noise filtering, they extracted 30 MFCC features for the sound prediction. However, the model has modest performance and the argument they give is because it was tested on the real data.

Renyu Liu et al(R. Liu et al., 2019) authors aimed to detect the most common respiratory sounds wheeze and crackle but due to the problem of data imbalance, they have changed the classification into classifying the respiratory sound as adventitious or not. They haven't used any mechanism of handling data imbalance problem. Authors extract Mel-filter bank features and used the features as an input to their CNN model. In comparison with the previous works that are based on traditional ML algorithms, this work introduces to use CNN model which is capable of deep feature extraction. This ability leads to a better result. But one of the downsides of Mel-filter bank is unless in its first and second-order differentiation Mel-filter bank extracts only static features which is not suitable for temporal data that changed through time. They have used their own additional dataset as a supplementary to the ICBHI dataset but the model performs much higher for the ICBHI dataset than for their own dataset or the combination of two. The test result obtained from using those databases is Accuracy of 61.02% for Mixed databases. It shows that this work needs improvement.

Fathi Demir et al (Demir, Sengur, et al., 2020) the goal of authors was to efficiently classify respiratory disease. The classification is applied to the respiratory sound from ICBHI 2017 dataset. pre-trained CNN models such as AlexNet, VGG-16, and ResNet-50 have been experimented. Due to its performance VGG-16 model has been used as a feature extraction cooperatively with SVM model for classification. Spectrogram audio features have been extracted and resampled to be used as an input. Using 10-fold cross-validation the model gives an accuracy of 63.09% which has an improvement from the previous works. One of the gap in this research is that in the feature extraction, they have used a linear spectrogram feature which is less likely to mimic the human

auditory system. The other gap is, knowing that the dataset is imbalanced they haven't employ any measure about it.

Victor Basu et al (Basu & Rana, 2020) in this paper neural network model is implemented to classify respiratory sound that has been extracted using Mel-frequency Cepstral Coefficient, round 40 features have been extracted. The dataset used to train the model was from ICBHI2017.In respiratory disease diagnosis, the sound sample is obtained from around five different chest locations, but in this paper, only three locations were used. Authors suggest using different neural network architecture or maybe with different machine learning or deep learning algorithm for further improvements also different feature extraction techniques should be applied for further performance improvement. An improvement of the previous works by Victor Basu et al fills the gap of data imbalance by augmenting the data using time-stretch. The authors extract 40-MFCC features as an input to the CNN model to predict the type of disease. This work fails to detect the events in the respiratory sound rather they use the annotated audio files.

2.7.1 Summary of Related Works

Table 1: Summary of Related Work

Authors and Year	***Objective***	***Dataset Used***	***Features Used***	***Algorithm implemented***	***Limitation***
Gaëtan Chambers et al. (2018)	❖ Predict the type of respiratory sound. ❖ Predict the status as "Health" or "Not Healthy"	ICBHI 2017	13 MFCC	SVM	❖ Low performance. ❖ No measure for data imbalance problem. ❖ Does not predict the type of respiratory disease.
N. Jakovljević and T. Lončar-Turukalo (2019)	❖ Respiratory sound classification as Norma, Wheeze, and Crackle.	ICBHI 2017	30 MFCC	HMM	❖ Modest Accuracy. ❖ No measure for data imbalance problem. ❖ Haven't include both wheeze and crackle sound.
Renyu Liu et al. (2019)	❖ Detection of Wheeze and Crackle	ICBHI 2017 + Custom Pediatric	Mel-filter	CNN + SVM	❖ Use only static features.

					❖ Haven't include both wheeze and crackle sound. ❖ No measure for data imbalance.
Fathi Demir et al. (2020)	❖ Predict respiratory diseases.	ICBHI 2017	Spectrogram	CNN (VGG-16)	❖ Used less important feature. ❖ Fail for diseases with overlap sound. ❖ No measure for data imbalance.
Victro Basu et al. (2020)	❖ Predict respiratory disease.	ICBHI 2017	40 MFCC	CNN	❖ Since respiratory sound is a single feature, it fails for diseases with common sounds.

Many of the reviewed research papers focus on adventitious respiratory sound classification but the rest papers try to predict respiratory disease from the breath sound only. However, the researchers used the techniques without considering overlapping and variability of the respiratory sound on some common diseases. Some even fail to distinguish a person with normal respiratory sound but with pulmonary disease. This shows that to make the prediction more precise and powerful medical history as factors should be included in addition to the breathing sound. From the papers seen the best feature extraction used is MFCC but MFCC alone is weak especially for crackle class prediction (Pham et al., 2014). Therefore, integrating other feature extraction to that of MFCC is an ideal solution. All the papers used ICBHI 2017 dataset, but the dataset has data imbalance problem to overcome this, different augmentation techniques should be experienced. This thesis tries to fill the specified gaps.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

This chapter grants the methodologies, procedures, and techniques used in achieving the main aim of the thesis work, predicting respiratory disease from respiratory sound and patient symptoms. The first section of the chapter explains the study design and approaches. Secondly, the data collection and datasets used along with their pre-processing are elaborated. The third section contains the algorithms used in developing the learning models. Finally, the tools and techniques used along with the evaluation mechanisms used are exhibited.

3.2. Research Approach and Procedure of the Study

According to the nature of the problem at hand and the research questions of the thesis, this research falls into quantitative research which could get results through experiments. Because the approaches are research design approaches that include collection and analysis of data, quantifying the data, build a relationship, and mathematical models to solve phenomena or prove a theory(Donnelly, 2014).

To meet the goal of this research work, different procedures have been applied. Those are a set of actions or working steps to reach the final desired output of the planned work. The procedures include identifying the research area, then reviewing what has been done and what is already known by different researchers followed by gap investigation and research question formulation to tools and methodology selection along with dataset collection and preparation, in the end designing and implementing the optimal proposed solution are the main steps or procedures used in the operation of this thesis. A scenic view of the procedures is shown in the figure below.

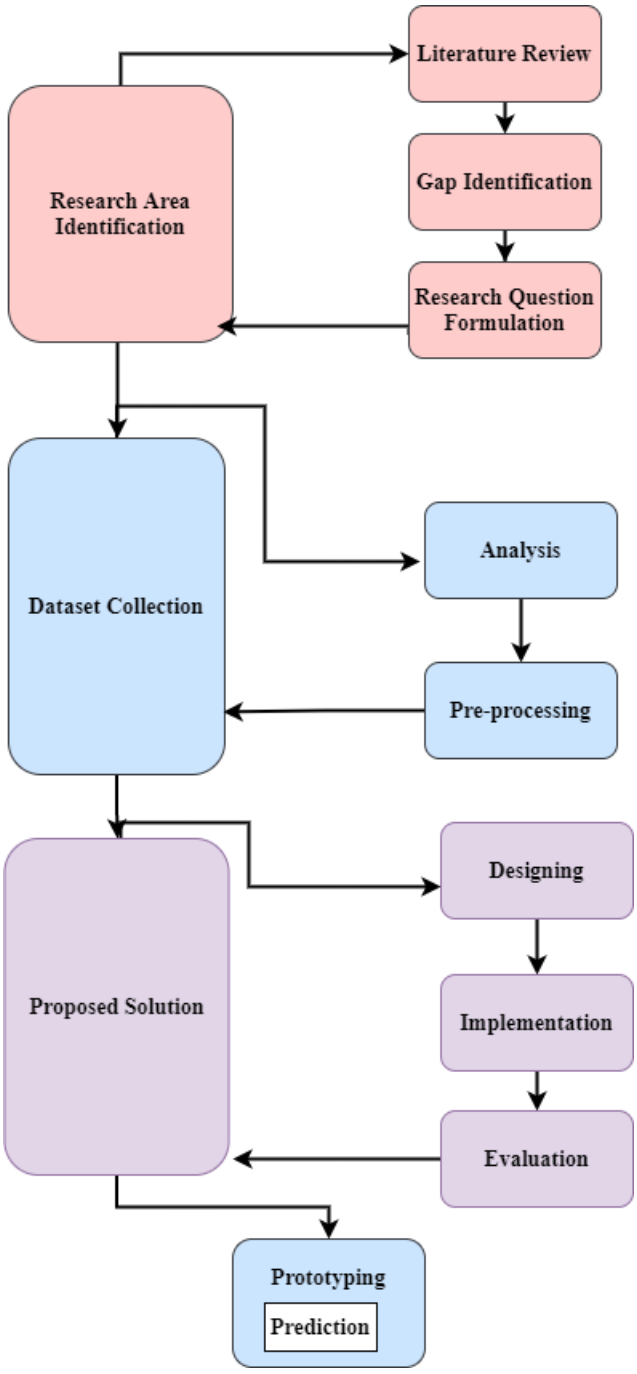


Figure 13- Procedures of the Research

3.3. Data Sources

Deep and Machine Learning algorithms are designed to develop models that learn from experience and examples. Those examples are collections of instances or observations also known as dataset. In order to train the learning model, it is a must to have a good dataset. Now a day's researchers

use either publicly available datasets or secondary datasets that have been prepared by the researcher themselves. In this thesis work, we have used two different datasets, the first one is a respiratory sound dataset. because of the universality of the respiratory sounds and due to the difficult situations to record the sound by ourselves we have used from publicly available ICBHI (International Conference on Bio Health Information) and the other patient symptom dataset was obtained from(SPMMC) St. Paul Hospital Millennium Medical College.

There are not many datasets available for respiratory sound data but there were some datasets that have been collected by some researchers, however, these researchers themselves claim that their collected dataset was not as effective as the ICBHI(R. Liu et al., 2019). Therefore, we have selected the diversified and trustworthy ICBHI dataset for respiratory sound data. ICBHI (International Conference on Biomedical and Health Informatics) is a seminar on biomedical issues and health informatic affairs. The initial category is in biosensors, electroceuticals, and other electronic health systems. The second category includes bioinformatics, big biodata, medical decision support system, and machine learning(Maglaveras & Paulo, 2017). The first version of the conference gives a challenge for different participants from different countries to come up with an algorithm that distinguishes respiratory sounds as non-adventitious or adventitious sounds that contain normal, wheeze, crackle, or wheeze and crackle sounds.

Next to Tikur Anbesa Medical College, SPHMC (St. Paul Hospital and Millennium Medical college is the largest specialized hospital in Ethiopia. The hospital was built in 1969 by Emperor Haileslase in collaboration with German Evangelical Church. The hospital was named St. Paul general specialized hospital until 2007 when the medical school opened in the new millennium era of the Ethiopian calendar(SPHMMC, 2019). The hospital has a chest clinic to treat Respiratory and chest-related pathologies as well as injuries. In the clinic there are specialized and sub-specialized pulmonologists, therefore collecting data from such a hospital is believed to give us quality data. The datasets collected from the above-mentioned sources have been explained below.

3.3.1. ICBHI Dataset Description

The ICBHI 2017 dataset has been prepared for the purpose of ICBHI respiratory challenge in 2017. The dataset was collected independently in two different countries by two research teams. The first research team from school of health science, university of Aveiro(ESSUA), Portugal. The other

research team was from Greece, Aristotle University of Thesolonke(AUTH) and University of Coimbra(UC). Two specialized pulmonologists and one cardiologist annotated the dataset(B. Rocha et al., 2017). Those research teams use different recording equipment such as AKG C417L Microphone, 3M Littmann Classic II SE, 3200 Electronic, WelchAllyn Meditron Electronic Stethosco

Figure 14-Audio recording equipment's[6]

Populations included in the dataset are both sex categories in all age groups, from elderly to children that are diagnosed with respiratory tract infection and some healthy patients. The dataset contains 6898 respiratory cycles of 920 recordings for 5.5hour long. The respiratory sounds are recorded from different chest locations in the range of 10-90 seconds. Chest locations are locations in which auscultation is performed, Trachea(Tc), Anterior left(Al), Anterior right(Ar), Posterior left(Pl), Posterior right(Pr), lateral left(Ll), Lateral right(Lr) are the seven chest locations. Example of the annotated respiratory sound along with chest location and recording equipment is shown below.

[A]. https://www.kpodj.com)

B. https://www.reddingmedical.com/classic-ii-se-stethoscope

C. https://www.medisave.eu

D. https://www.amazon.in/Welch-Allyn-Master-Electronic-Stethoscope

	start	end	crackles	weezels	pid	mode	filename
0	1.862	5.718	0	1	160	mc	160_1b3_Lr_mc_AKGC417L
1	5.718	9.725	0	1	160	mc	160_1b3_Lr_mc_AKGC417L
2	9.725	13.614	0	1	160	mc	160_1b3_Lr_mc_AKGC417L
3	13.614	17.671	0	1	160	mc	160_1b3_Lr_mc_AKGC417L
4	17.671	19.541	0	0	160	mc	160_1b3_Lr_mc_AKGC417L

Figure 15-Annotated respiratory Sound file

Table 2- Description of each column

No	***Column Name***	***Description***
1	Start	Beginning of respiratory cycle
2	End	End of the respiratory cycle
3	Crackles	Crackle's (presence=1, absence=0)
4	Wheezes	Wheeze's (presence=1, absence=0)
5	Pid	The patient identification number
6	Mode	The sound Acquisition mode
7	Filename	The audio file name

3.3.2. Patient Medical History Dataset Description

The patient medical history dataset is a dataset that contains the symptoms of the patients along with the necessary demographics and diagnosis information. The collected patient history data was provided by St. Paul Hospital and Millennium Medical College. We have collected the symptom of patients who have been diagnosed at the chest clinic of the hospital. The collected data includes 5year consecutive data (from 2016- 2020 GC.) of both hospitalized and outpatients. While collecting the data our inclusion criteria was patients with Asthma, COPD, Bronchiolitis, or pneumonia diagnosis. The patient history file contains all the information about

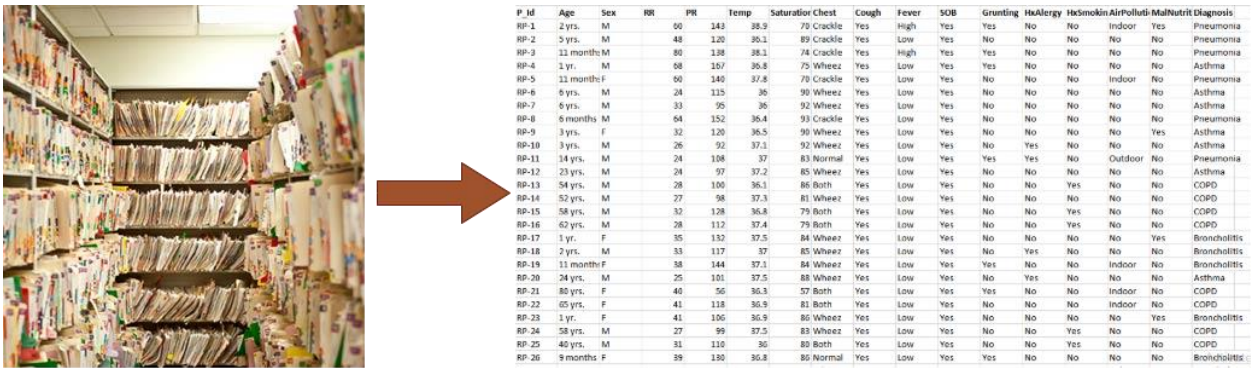
the patient, but all the information was not as valuable as the features used. To achieve that, a health expert (Medical Doctor) helps us in outlining the most important features and in reading the appropriate patient evaluation. Generally, the dataset building procedure looks as follows.

Step-1 HMIS (Health Management Information System)

HMIS is a medical record of patient's information and health data to be stored and retrieved for decision making and statistical purposes. We have used HMIS from adult chest clinic, emergency ward, and pediatric departments to retrieve card numbers of patients with the desired disease types.

Step-2 From Record center to Tabular data

By referring to the patient Id on the HMIS, from data record center we have got each patient's history in a hard copy; after reading each necessary information with the expert, we have prepared a tabular dataset of 16 features with their respective annotation in .csv file format. A sample dataset is given in *Appendix A.*



P_Id	Age	Sex	RR	PR	Temp	Saturation	Chest	Cough	Fever	SOB	Grunting	HxAlergy	HxSmokin	AirPolluti	MalNutrit	Diagnosis
RP-1	2 yrs.	M	60	143	38.9	70	Crackle	Yes	High	Yes	Yes	No	No	Indoor	Yes	Pneumonia
RP-2	5 yrs.	M	48	120	36.1	89	Crackle	Yes	Low	Yes	No	No	No	No	No	Pneumonia
RP-3	11 months	M	80	138	38.1	74	Crackle	Yes	High	Yes	Yes	No	No	No	No	Pneumonia
RP-4	1 yr.	M	68	167	36.8	75	Wheez	Yes	Low	Yes	Yes	No	No	No	No	Asthma
RP-5	11 months	F	60	140	37.8	70	Crackle	Yes	Low	Yes	No	No	No	Indoor	No	Pneumonia
RP-6	6 yrs.	M	24	115	36	90	Wheez	Yes	Low	Yes	No	No	No	No	No	Asthma
RP-7	6 yrs.	M	33	95	36	92	Wheez	Yes	Low	Yes	No	No	No	No	No	Asthma
RP-8	6 months	M	64	152	36.4	93	Crackle	Yes	Low	Yes	No	No	No	No	No	Pneumonia
RP-9	3 yrs.	F	32	120	36.5	90	Wheez	Yes	Low	Yes	No	No	No	No	Yes	Asthma
RP-10	3 yrs.	M	26	92	37.1	92	Wheez	Yes	Low	Yes	No	Yes	No	No	No	Asthma
RP-11	14 yrs.	M	24	108	37	83	Normal	Yes	Low	Yes	Yes	Yes	No	Outdoor	No	Pneumonia
RP-12	23 yrs.	M	24	97	37.2	85	Wheez	Yes	Low	Yes	No	No	No	No	No	Asthma
RP-13	54 yrs.	M	28	100	36.1	86	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-14	52 yrs.	M	27	98	37.3	81	Wheez	Yes	Low	Yes	No	No	No	No	No	COPD
RP-15	58 yrs.	M	32	128	36.8	79	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-16	62 yrs.	M	28	112	37.4	79	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-17	1 yr.	F	35	132	37.5	84	Wheez	Yes	Low	Yes	No	No	No	No	Yes	Bronchollitis
RP-18	2 yrs.	M	33	117	37	85	Wheez	Yes	Low	Yes	No	Yes	No	No	No	Bronchollitis
RP-19	11 months	F	38	144	37.1	84	Wheez	Yes	Low	Yes	Yes	No	No	Indoor	No	Bronchollitis
RP-20	24 yrs.	M	25	101	37.5	88	Wheez	Yes	Low	Yes	No	Yes	No	No	No	Asthma
RP-21	80 yrs.	F	40	56	36.3	57	Both	Yes	Low	Yes	Yes	No	No	Indoor	No	COPD
RP-22	65 yrs.	F	41	118	36.9	81	Both	Yes	Low	Yes	No	No	No	Indoor	No	COPD
RP-23	1 yr.	F	41	106	36.9	86	Wheez	Yes	Low	Yes	No	No	No	No	Yes	Bronchollitis
RP-24	58 yrs.	M	27	99	37.5	83	Wheez	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-25	40 yrs.	M	31	110	36	80	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-26	9 months	F	39	130	36.8	86	Normal	Yes	Low	Yes	Yes	No	No	No	No	Bronchollitis

Figure 16: Patient medical history data preparation

The prepared dataset generally contains 4167 elements. Since we have 4 different classes, we have made the dataset to have enough allocation. The dataset distribution ratio among the classes is shown below.

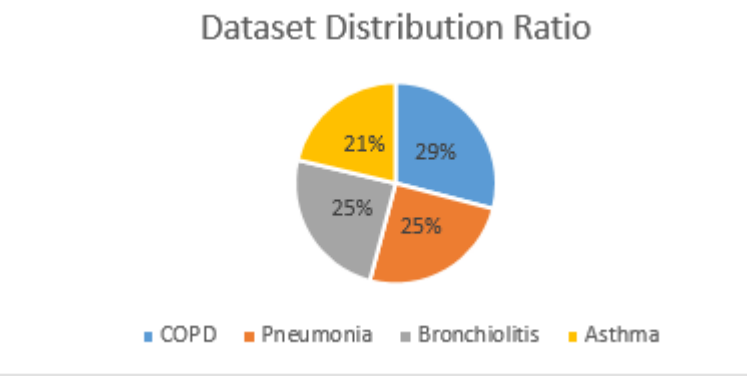


Figure 17- Patient medical history dataset distribution

Demographically the dataset contains somehow sufficient dispensation between male and female subjects. This is in consideration of not making bias in prediction. The total count and proportion of the two categories is manifested below.

Table 3- Gender distribution of the dataset

Gender	***Counts***	***Percentage***
Male	2375	57.0%
Female	1792	43.0%

3.4. Data Preprocessing Techniques

Transferring information that has been collected into some meaningful form is the main aim of data pre-processing. We have two different datasets, the patient medical history structured dataset and the respiratory sound time-variant dataset. Data pre-processing techniques are different based on the nature and type of the dataset. Therefore, we have used different pre-processing techniques separately as follows.

3.4.1. ICBHI Data Pre-processing

Pre-processing of audio data starts from reading the audio file and converting the file into a 32-bit Numpy float array. This can be achieved by quantizing the amplitude with a limited number

of bits. After reading the audio file other pre-processing techniques are used, here the pre-processing techniques are described briefly.

Segmentation: is a process of partitioning the audio signal to a specified period of time. The reason behind segmenting the audio file is that the annotation in the ICBHI dataset was done for some time interval ranging from 2sec to 20 seconds. Therefore, to match the annotation with the audio file we need to segment the audio concerning the length in the annotation.

Re-Sampling: originally every audible sound has its own sampling rate but for ease of use we should re-sample each audio file uniformly. Audio can be recorded as mono or stereo channels; channels are the position of each audio source in the audio signal. Those channels contain amplitude at a given instance time of the audio file. Sampling is the clipping of a continuous signal into discrete values in form of series

Slicing: slicing is trimming the audio file into an equal amount of period. The rationale in slicing is that input to the CNN model should be uniform in size for the purpose of learning and perform the prediction for one respiratory cycle. In the case of our dataset slicing is a dual beneficial one to get each breathing cycle and is also used as an augmentation tool.

Augmentation: Augmentations are manipulations that exposed the model to a wider array of training examples by creating different versions of similar content to the data. The ICBHI dataset has a problem of data imbalance, to overcome this augmentation plays a huge role. In addition to this, augmentation helps us to create other useful training samples that somehow resemble the real world(Salamon & Bello, 2017). For that reason, the model's ability to generalize increases. Among many augmentation techniques, we have chosen three augmentation techniques.

1. ***Time-stretch:*** alters tempo which is the measurement of how fast or slow the audio is performed, while keeping pitch the same(Iwana & Uchida, 2020). Time-stretch speeds up or slows down the audio signal depending on the stretch factor used. The advantage of time-shifting is that it preserves the pitch so that its ability to preserve information.

2. ***VTLP (Vocal Tract Length Perturbation):*** first implemented in the legendary paper by Navdeep Jaitly et al, 2013. VTLP improves augmentation for audio data. Its

significant and consistent gain makes the feature preferable to be used especially for datasets with limited transcribed training data(Jaitly & Hinton, 2013).

3. Spec-Aug: is an augmentation that works on the Mel-spectrogram feature of the audio signal directly. This method considers Mel-spectrogram as if it is image and performs augmentation both in time(x-axis) and frequency(y-axis)(Series, 2020). The augmentation was proposed by Daniel S. Park et al, it reaches to be the state of the art by outperforming all the remaining augmentations in speech(Park et al., 2019). Its simplicity and low computational cost make it more preferable.

Feature Extraction: Audio features are a description of sound or an audio signal that can basically be fed into statistical or ML models**.** In audio processing, feature extraction is an analogy to the inner ear in the human body that perceives audio signal and transmit electrical impulse to the brain(Knight et al., 2020). In feature extraction, certain features are extracted and provided for the learning model. Audio feature extraction starts from time-domain features to frequency-domain features and combining them in the time-frequency domain followed by deep features. To be benefited from time-frequency features as well as deep features we have used time-frequency features along with deep learnings ability on deep feature extraction(Sharma et al., 2020). The nominated feature for respiratory sound prediction are Mel-spectrogram and MFCC.

1. MFCC: Discrete cosine transform (DTC) on log power spectrum provides MFCCs. It is logarithmically Mel-scaled that resembles the human auditory system. This makes MFCC features key for different audio processing(Sharma et al., 2020). In audio analysis problem, Mel frequency ceptral coefficient (MFCC) have been serving as the supreme acoustic feature representation(Purwins et al., 2019). Because of this reason we have chosen MFCC for feature extraction.

2. Mel- spectrogram: Recently the concept of image processing is dominantly applying in audio and sound analysis. Therefore, instead of using manually selected audio features, image-like representations of audio is being used(Purwins et al., 2019) (Knight et al., 2020). Mel-spectrogram is one way of representing the audio signal in image by detailing the frequency configuration of the sound over the time domain. This feature is

said to be the best in speech and audio analysis, therefore we use the feature to exploit its effect in respiratory sound detection.

3.4.2. Patient Medical History Data Pre-processing

Structured data pre-processing is a crucial and most important step in developing a qualified data set. The pre-processing is formulated by data cleaning, transformation and feature selection. After all those pre-processing steps, the dataset keeps its formality and uniformity to be computed by the model Algorithm.

3.4.2.1. Data Cleaning and Unit Conversion

In data cleaning incorrect and inconsistent records are replaced and cleaned. Unit conversion is converting a data from certain measurement to the desired measurement. The patient history dataset contains all age groups from elders to infants of age less than a year. Due to this, the age column is recorded in two units, in a month and in a year. But for the purpose of appropriate manipulation, the units should be uniform. In order to achieve the unity of measurements, we have converted months to years.

3.4.2.2. Data Transformation

In tabular data, the representation of features is mostly heterogeneous. However, neural network models cannot handle categorical data as it is(Marais, 2019). To make every entry in the dataset suitable for the model training; one hot encoder and label encoding techniques are considered.

3.4.2.3. Feature Selection

All features did not contribute equally to the model prediction. Manually checking their contribution is very tedious rather there are algorithms for checking feature importance. Extra Tree Classifier is used in this work because of its simplicity and faster execution(Khalid et al., 2014).

3.5. Model Building Algorithms

Respiratory disease prediction from respiratory sound and other patient symptoms is a multi-class classification problem. Usually, those classification problems are handled using supervised machine learning and deep learning algorithms. In this research, we have chosen to work with deep learning algorithms. The main reason behind choosing deep learning is that it’s the ability to deep extract features than using only the already extracted features as in other traditional ML

algorithms; which eventually leads to better performance. Here the selected algorithms are concisely explained.

3.5.1. Convolutional Neural Network (CNN)

Formerly CNN algorithms were built for image processing in mind. Nevertheless, not long ago researchers have used the algorithm in sound analysis and classification. In audio classification, CNN has outwork the traditional ML algorithms(Demir, Turkoglu, et al., 2020). Our intention behind choosing the CNN model among many other corresponding DL algorithms is that CNN models have computational efficiency and speed where the comparative RNN models are less parallelize and therefore they are slower. RNNs can be very effective for audio synthesis but the problem we have is audio analysis therefore we have chosen to work with CNN models(Purwins et al., 2019). Also, its achievement for more than decades throughout those days, makes CNN models more desirable.

3.5.2. Deep Neural Network (DNN)

Deep Neural Network's ability to automate feature extraction is a huge advantage over tabular datasets. Also, It preserves the risk of dropping into common premises(Uddin et al., 2019)(Marais, 2019). But certainly, traditional ML algorithms perform undoubtedly for the tabular dataset. However, to make our work portable and accessible, we need to fit the model into edge devices. The first and the only method is using TensorFlow lite, which is used for deep learning models as a backend in the Keras library. Paradoxically other more complex deep learning methods would underperform for the tabular dataset. Therefore, we have chosen Deep Neural Network for disease prediction.

3.6. Design and Development Tools

Design and development tools are necessary and important gadgets to make documentation, implementation as well as prototyping very easy and handy.

3.6.1. Design Tool

Draw.io[7]***:*** is designing software that allows to create flowcharts and custom-made diagrams. The software has plenty of shapes with easy to use drag, drop functionality and tools to be used, in addition, it helps us to store the diagram in the cloud or the local device.

[7] https://www.app.diagrams.net

3.6.2. Development Tools

Google Drive[8]***:*** is a cloud-based file storage service owned and managed by Google. The drive helps in synchronization, so that anywhere, anytime the stored files can be accessed among all the sync devices.

Keras[9]***:*** is a high-level modular deep learning framework. It is designed for developers to ease of usability. Keras is written in python and it works on the top of Theano or TensorFlow libraries.

Librosa[10]***:*** is a python package, mainly used for music information retrieval(MRI) as great music and audio analysis tool.

Tensor flow[11]***:*** it is a numerical computation library developed by the Google team. Tensor flow represents data as a tensor, whereas a tensor is a multidimensional array. The library uses a graph to implement learning algorithms. In the graph, the tensor is used as an edge and mathematical functions are used as nodes. Tensor flow helps to deploy and run ML algorithms on CPU, GPU, and cluster.

Colab[12]***:*** is an abbreviation of Colaboratory, which is an open environment for Jupiter Notebook, completely running on the cloud. Colab does not require any setup and it has many python packages and libraries installed. The environment has both CPU and GPU runtime.

Android Studio[13]***:*** it is an IDE(Integrated Development Environment) on top of JetBrains's IntelliJ IDEA. Android studio is used to develop and build applications for Android devices.

3.7. Evaluation Methods

Evaluating or assessing one's work is a way of showing its performance. In this thesis, we have used two evaluation methods. The first one is algorithmic methods and the second one is using a health expert's evaluation.

[8]https://www.google.com/drive
[9] Keras.io
[10] https://librosa.org
[11] https://www.tensorflow.org
[12] Colab.research.google.com
[13] https://developer.android.com/studio

3.7.1. Algorithmic Evaluation Method

Machine learning algorithms learn from data but to evaluate how much the machine can generalize we have to test the algorithm using data that has not been on the training set. If the model has not been tested, a big problem in ML, which is overfitting may happen. To overcome this problem researchers and machine learning experts have implemented different methods, one of them was train-test split. However, this evaluation method is prone to overfitting. Afterward, train-test validation was introduced with better performance, nevertheless, it was not free from overfitting. currently, the best evaluation method among others is k-fold cross-validation(Sokolova & Lapalme, 2009). This evaluation method splits the whole data into k-folds. Every fold has to hold representative data from each class. Each k-group data is used as a train set k-1 times and one time as test data. This helps to see the model's performance and its ability to generalize. To discriminate the prediction power among model's we use evaluation metrics such as Accuracy, precision, recall, and F1-score.

Accuracy: it is a proportion of correct results in the total number of cases.

$$\text{Accuracy} = (TP + TN)/(TP+TN+FP+FN)$$

Precision: it gives the proportion of true positive cases among positive predictions.

$$\text{Precision} = TP/(TP+FP)$$

Recall: it shows how many actual positives are correctly classified.

$$\text{Recall} = (TP)/(TP+FN)$$

F1-score: it is the harmonic mean of precision and recall.

$$\text{F1-score} = 2^* (\text{precision}^*\text{recall}) / (\text{precision} + \text{recall})$$

Where:

False Positive (FP) = when predicted positive and actual is negative.

False Negative (FN) = when predicted negative and actual is positive.

True Positive (TP) = when predicted positive and actual is positive.

True Negative (TN) = when predicted negative and actual is negative.

3.6.2 Health Expert Evaluation Method

Using algorithmic methods respiratory sound prediction and Pulmonary pathosis prediction can be evaluated separately. But to get the acceptance and overall evaluation of the proposed

solution, we have arranged a health expert evaluation by a pulmonologist from St. Paul hospital. This assessment is exerted on newly unseen data. In the evaluation, our proposed solution's prediction power is compared to that of an experienced pulmonologist.

CHAPTER FOUR

4. Proposed Solution Design for Respiratory Disease Prediction

4.1. Chapter Overview

Previously the research procedure, methodologies, and tools used have been discussed. In this chapter, the architecture, design, and the working principle of the contemplated solution is elaborated. The main content of this chapter is divided into three segments. The first segment deals with the pavements to respiratory sound prediction. The second section discussed the strategy of combining the lung sound with patient's medical history to classify pulmonary pathosis, where pulmonary pathosis is a classification of disease based on structured data. Ultimately it gives the respiratory disease prediction. The last section is all about the structure and components of the proposed prototype.

4.2. Proposed work Architecture

As it is the objective of this study to predict respiratory disease from respiratory sound and patient medical history, respiratory sound, and patient medical history datasets have been used. Due to the nature of the data types, we cannot develop a single model that can work for both time-variant data and structured datasets. To overcome this situation, we have a model for the time-variant dataset and use the output from this model as an input to the tabular dataset classifier. Using this logic, the overall architecture of the proposed solution is shown below.

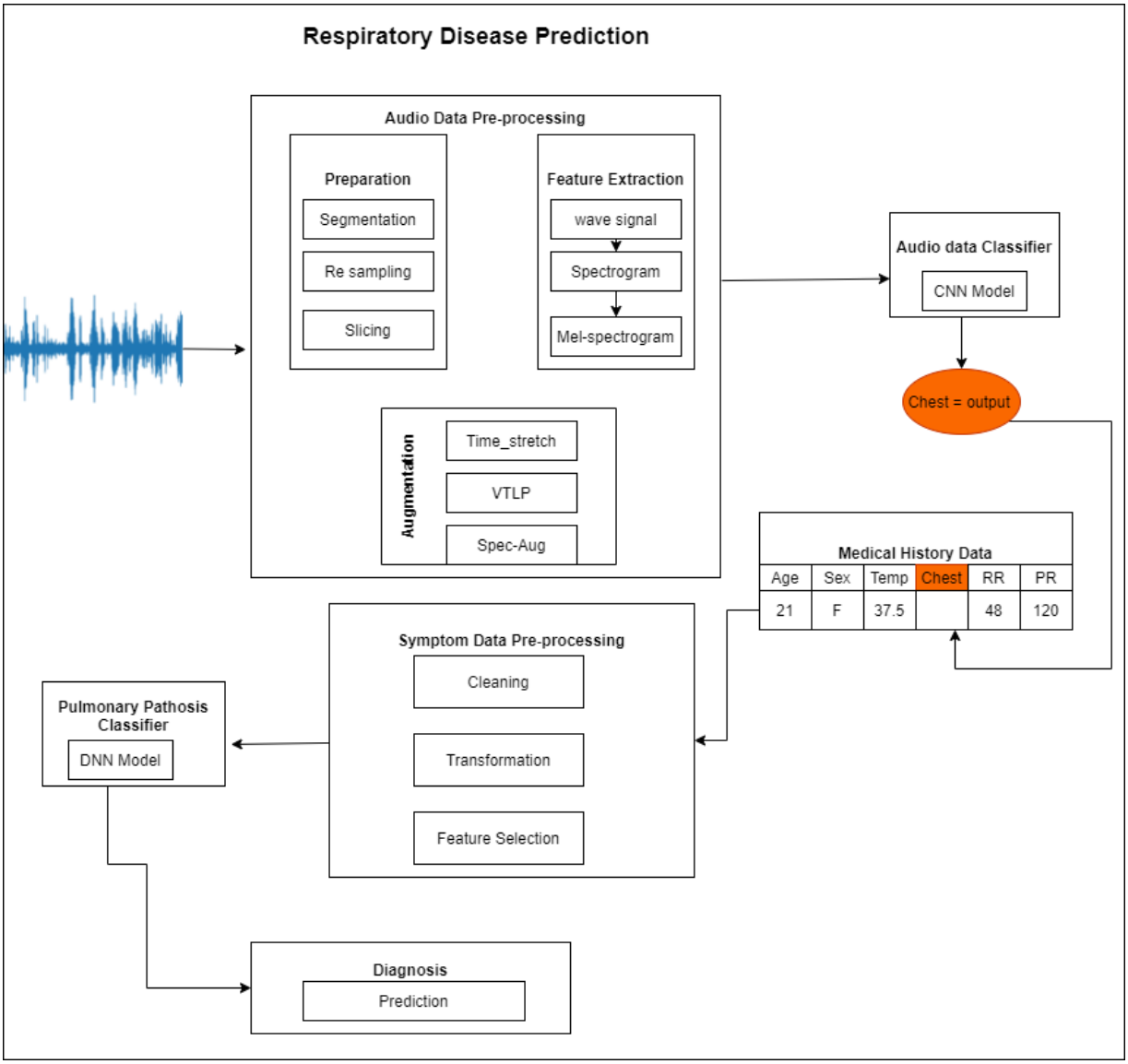


Figure 18: Overall architecture of the proposed solution

4.3. Data Pre-processing

This research works on two different datasets, patient symptom structured dataset and respiratory sound time-variant dataset. However, data pre-processing techniques are different based on the nature and type of the dataset. Therefore, we have used different pre-processing techniques separately as follows.

4.3.1. ICBHI Data Pre-processing

ICBHI dataset contains recorded audio data with different sample rates, different quantization, and different audio length. Therefore, the dataset needs to be pre-processed to make the data

optimized for the model training and overall expected result. The pre-processing step starts from reading, then re-sampling each audio file into the sample rate we set, also to make a balance between different classes data augmentation is performed; next to make the data suitable for model training, slicing the data in equal length in addition feature extraction are the core pre-processing steps. The pre-processing pipeline is shown below.

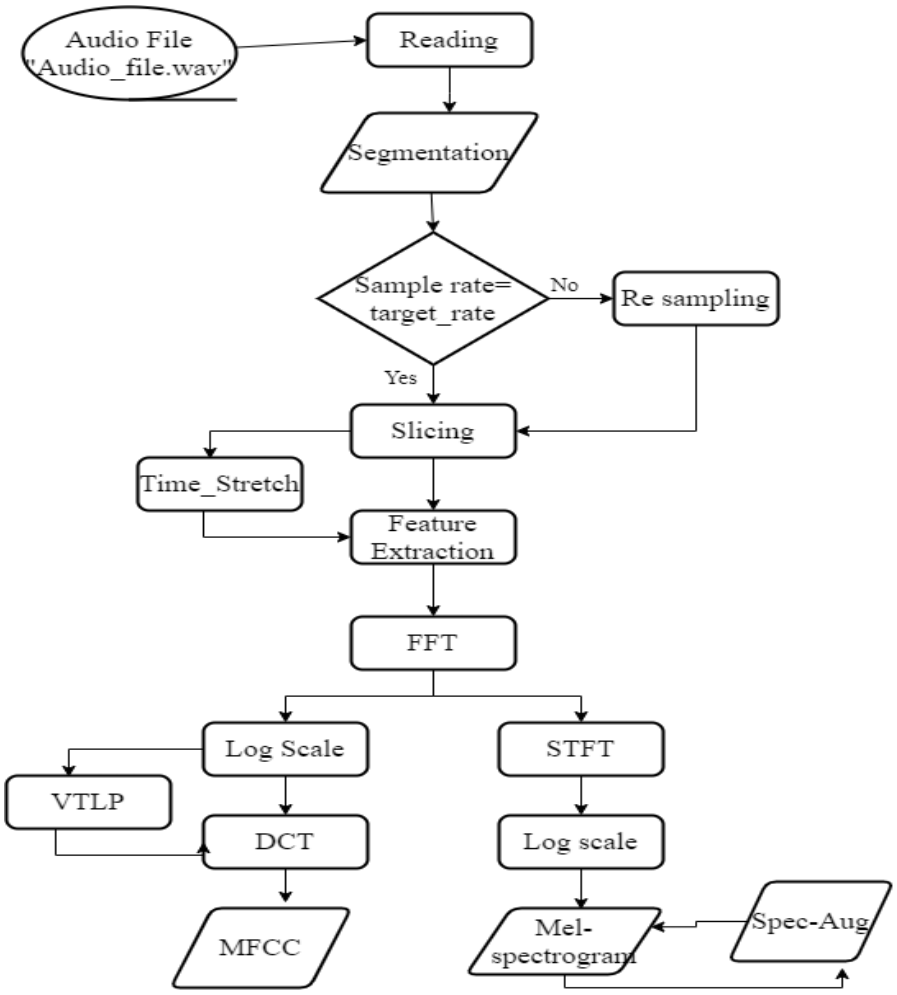


Figure 19-Audio data pre-processing pipeline

The respiratory sound file comes up with uncompressed 'wav' file format; wav files are digital files having stored the original data as-is in full format. Digital audio is recorded by taking sampling rate which is samples of the original sound wave at a specified rate. Reading a wav file is changing the signal into an array. The size of an array can be calculated as:

$$Signal = sr * T$$

Equation 1-Audio Size Calculation

Where: Signal is the size of an array, Sr is the sample rate of the audio file and T is the duration of the audio.

4.3.1.1. Segmentation

While the audio file contains the sound, metadata of each sound file contains annotated audio sample in a certain duration. To map with the annotated duration clip, we need to segment the audio file with respect to sound clip duration. Segmentation of the audio file helps us to map the class or label interpretation with the NumPy array of the appreciated audio file. The illustration and the algorithm is shown as follows.

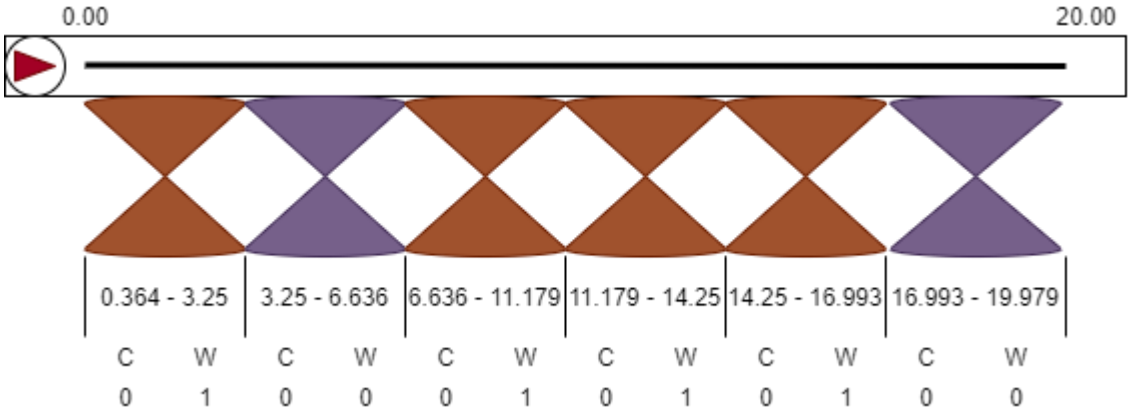


Figure 20-Illustration of Audio Segmentation

Where: C represents crackle, w represents wheeze, also 1 shows presence and 0 absence.

ALGORITHM 1: ALGORITHM TO SEGMENT AUDIO DATA

Input: Audio data along with Start and End index

Output: Segmented Audio data

Maximum index ← the length of audio data

Start ← min [start index, maximum index]

End ← min [end index, maximum index]

return audio data chopped with start and end length

Algorithm 1- Audio Data Segmentation

4.3.1.2. Re-Sampling

The audio files in the dataset are recorded with different instruments which gives the file a different sample rate because the way the devices digitize the audio data is not the same. The ICBHI dataset contains audio files with different sampling rates and frequencies, 4khz, 10khz, 44.1khz(Swaminathan et al., 2017). For the purpose of training the proposed model, all the records in the dataset should have the same sampling rate. The process of once again sampling the sound signal with the same sampling rate is called resampling.

ALGORITHM 2: ALGORITHM TO RESAMPLE AUDIO DATA

Input: Audio data along with its Sample rate, target sample rate

Output: Re-sampled Audio data

If(Sample rate is not equal to Target rate)

Multiplayer ← the division of target rate to sample rate of the audio.

New sample ← length of audio data multiplied with the multiplayer.

Resampled audio data ← interpolate new sample to the original audio.

End If

return audio data chopped with start and end length

Algorithm 2- Resampling Audio Data

4.3.1.3. Slicing

The ICBHI dataset contains different samples the maximum sample size is 16.0 sec and the minimum is 2 sec, but the proposed model needs to have an equal amount of sample size because the input to the model must be of the same size.

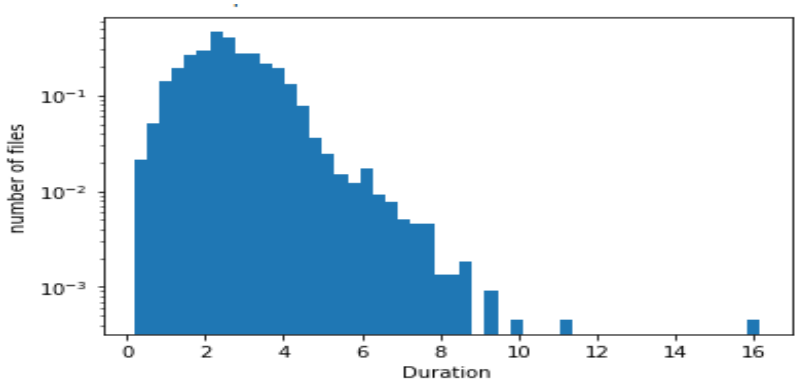


Figure 21: Duration of each sound file

As stated in the literature review *section 2.1.3* by average human beings tend to have 12-16 breathing cycles in a minute. If we take a minimum of 12 breathing cycles per minute a single breathing rate will have a duration of 5 sec.

12 cycle = 1 min =60 sec.

1 cycle = 60 sec / 12 cycle= 5 sec.

To get the optimal sample size in the amount of time, we have sliced the data for every breathing cycle. While slicing some audio signals might not have a sample size to be equally sliced at a rate of 5 sec, In such situations, we should pad the audio to zero(Silent).

ALGORITHM 3: ALGORITHM TO SLICE AUDIO DATA
Input: Segmented Audio data and Desired length of audio ***Output:*** Sliced Audio data ***Number of Slices*** ← total audio length divided by the desired length ***For variable x is in Number of Slices*** Slice length ← audio data divided by number of slices ***IF(sliced length is less than desired length)*** Pad the sliced audio with zeros or silent ***End If***

End for

return audio data sliced with the desired length

Algorithm 3-Slice audio data

4.3.1.4. Feature Extraction

Features can be obtained either in designed features or from the deep learning models. Despite the fact that CNN is best at learning features automatically; designed features also give us the information needed for the task we are trying to solve(I. Khan et al., 2019). Audio signal features fall into time-domain and frequency-domain or both. For the purpose of not losing both time and frequency information, we extract audio features using the time-frequency domain.

Time Domain: in the time domain audio features are extracted from amplitude-time representation or the waveform of raw audio signal.

Frequency domain: as the name suggests frequency domain gives emphasis to the frequency component of the audio signal. Frequency domain can be derived from time-domain by performing Fast Fourier transform(FFT). Frequency domain is represented as a frequency-magnitude graph; while magnitude reveals the contribution of each frequency to the overall audio data. The Mathematical formula to calculate the FFT and change the raw waveform to power spectrum and is shown below.

$$FFT\{x(t)\} = X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

Equation 2- Fast Fourier Transform

$$Sxx(f) = X(f)X^{*}(f) = |X(f)|^{2}$$

Equation 3- Equation to Obtain Power Spectrum

Where: x(t) is the time domain signal, X(f) is the FFT, ft is the frequency to be analyzed.
S_{xx} is the power spectrum, $x^{*}(f)$ is the complex conjugate of X(f).

Time-frequency domain: time-frequency is a combination of time-domain and frequency-domain of the audio data(signal). STFT (short time Fourier transform) is applied on the raw audio time-domain to yield time-frequency domain. Spectrogram, Mel-spectrogram Constant-

Q are the most common representation of time-frequency domain features. The mathematical expression and representation of discrete-time STFT is shown as follows.

$$STFT(x(t),\tau) = x(f,\tau) = \frac{1}{N}\sum_{t=1}^{N} x(t)w(t-\tau)e^{-2\pi jft/N}$$

Equation 4- Short-time Fourier transform

Where: f is the frequency, t is the time, τ shows the shift, and w is the Window Length.

Spectrograms are a visual representation of sound signals**. a** spectrogram is also known as a "heat map", is a representation of the sound signal in terms of its frequency and the strength, pitch, or loudness of the signal over a while. To convert a raw sound signal having amplitude over some time interval into a spectrogram of frequency over a while including its magnitude, we use a function of Short-Time Fourier Transform (STFT). Then by converting the linear spectrogram with a logarithmic function we can obtain Log-spectrogram, which is more close to the natural hearing schema. The output waveform, power spectrum, and Log-spectrum of the four respiratory sound types is Shown in *Appendix D.*

I. Mel-Spectrogram

Based on a psychoacoustic experiment, human beings tend to perceive audio in a non-linear frequency of logarithmic form as a decibel. This is because human beings are sensitive to lower frequency variations than higher frequencies. Mel-spectrograms are the perceptually relevant frequency representation(Sharma et al., 2020)(Breebaart et al., 2014). Because audio data processing in deep learning is mainly based on spectrograms, we have used the above STFT and convert the input 'wav' format sound signal to Mel-spectrogram. Some examples are shown below. Mel spectrogram consumes some parameters such as number of FFT, hop length, minimum frequency, maximum and maximum frequency. The parameters can be adjusted but based on previous studies the default value gives much better result.

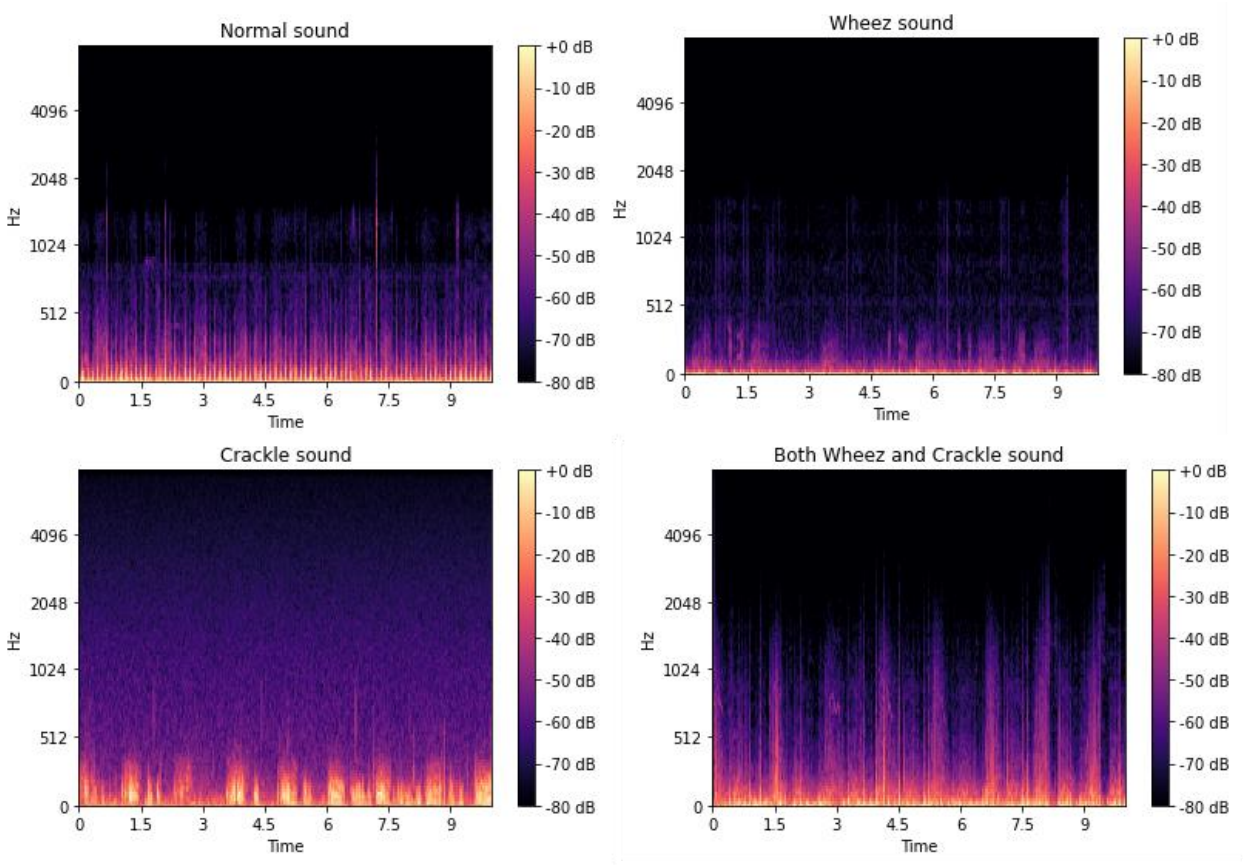


Figure 22- Mel-spectrogram of each sound type

II. MFCC (Mel Frequency Cepstral Coefficient)

MFCC is one of the most important features in audio analysis. In order to obtain the MFCC feature, the signal need to be windowed first, then DFT (Discrete Fourier transform is applied) also logarithm of the magnitude is taken along with the Mel scaled frequencies, finally, the DCT(Discrete Cosine Transform) is applied. The number of MFCC controls the number of features that need to be extracted(Karthikeyan & Mala, 2018).

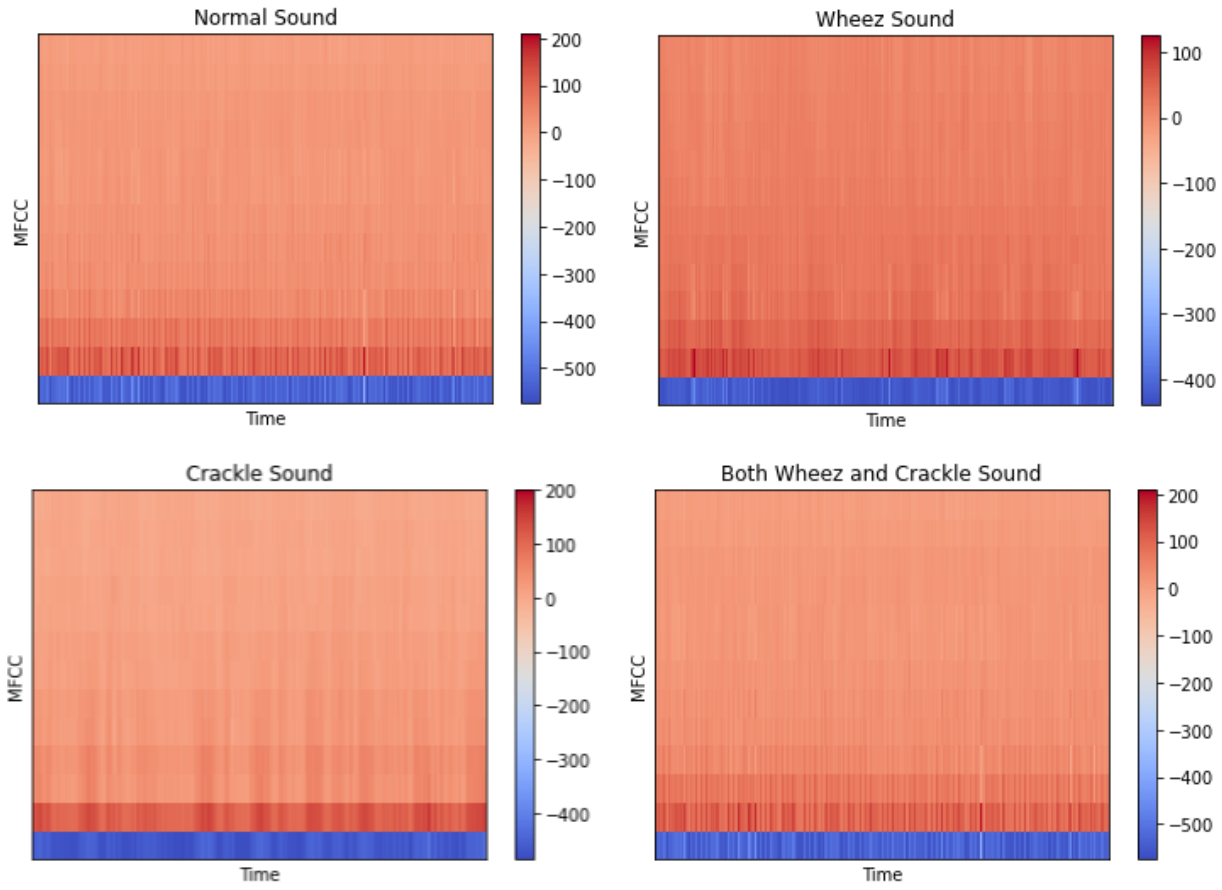


Figure 23- MFCC of each sound type

4.3.1.5. Augmentation

In this work, 3 different augmentation techniques are applied. Time-stretch in the audio signal prior to feature extraction also spectrogram augmentation and VTLP (Vocal Tract Length Perturbation) after feature extraction are explained on the way those augmentations work for respiratory sound classification.

Time-Stretch: keeping the pitch unchanged; this augmentation technique speeds up or slows down the audio sample(Nanni et al., 2020). The speed up and slow down are based on the amount of stretching rate we use if the rate is less than 1 then it slows down or if the stretch rate is greater than 1 then it speeds up. For the exact reason of not losing the sound events, we have stretched the time by 10 % with a repeat of a single factor.

ALGORITHM 4: ALGORITHM OF TIME STRETCH AUGMENTATION

Input: Audio time series, the maximum change

Output: Stretched Audio time series

If(y>1)

    # Speed up the audio time series by

    Stretch_factor = (+ve random number) * (maximum change /100)

    Audio data = sample rate * Stretch_factor

Else

    #Slow down the audio time series

    Stretch_factor = (-ve random number) * (maximum change /100)

    Audio data = sample rate * Stretch_factor

End If

return audio time series stretched by a specific rate

Algorithm 4- Time stretch

***Spectrogram Augmentation*:** as allude in *section 3.4.1* Spectrogram Augmentation or Spec-Aug considers spectrogram as an image for audio data. Nevertheless, image augmentation techniques cannot work directly to the spectrogram because of it is temporal data content. The Spec-Aug is applied in the Mel-spectrogram of the audio signal by time wrapping and frequency masking(Nanni et al., 2020)(Hwang et al., 2019). Frequency masking is performed by covering multiple frequency band signals in the horizontal axis whereas, time wrapping is performed by shielding multiple time band signals in the vertical axis(Park et al., 2019). This phenomenon allows Spec-Aug to focus on the entire spectrum. Spec-Aug for both crackle and wheeze class is exemplified in the following picture.

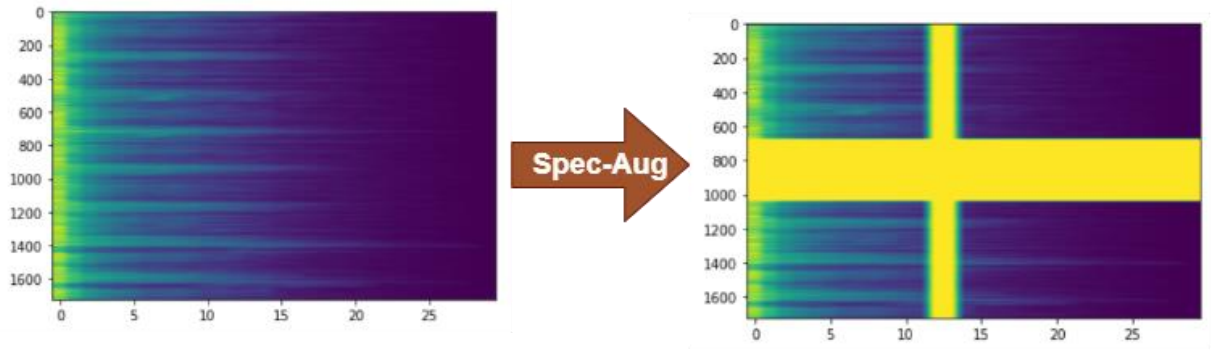


Figure 24- Exemplifying Spec-Aug on Both Crackle and Wheeze Class

ALGORITHM 5: ALGORITHM OF SPECTROGRAM AUGMENTATION

Input: Mel-spectrogram, Frequency and Time mask parameters

Output: Augmented Mel-spectrogram

#frequency shielding

F <- Frequency mask parameter

#$f \in [0, F]$

h <-Mel frequency channels

$f_o \in [0, h - f]$.

Masked frequency = $[f, f_o + f]$

 # time shielding

#T <- Time mask parameter

#$t \in [0, T]$

$t_o \in [0, T - t]$.

Masked time = $[t, t_o + t]$

return Mel-spectrogram shielded in frequency and time.

Algorithm 5- Spectrogram Augmentation

VTLP: The general principle of VTLP shift is that it generates a random wrap factor μ and wraps the frequency axis, such that a frequency is mapped to a new wrapped frequency**.** The VTLP shift is directly applied to the filter banks. Based on the previous studies the wrapping factor is set between 0.9 and 1.1 while beyond this interval gives non-realistic distortions(Jaitly & Hinton, 2013). The formula for conversion raw frequency to Mel-filter bank is as follows.

$$m(f) = 2595 * \log(1 + \frac{f}{700})$$

Equation 5- Frequency to Mel-scaled frequency

Where: f is frequency and $m(f)$ is Mel-scaled frequency

ALGORITHM 6: ALGORITHM OF VTLP AUGMENTATION

Input: Mel-filter bank, Wrapped factor (μ), boundary frequency(f_n)

Output: Augmented Mel-filter bank

If (frequency(f) $\leq \boldsymbol{f_n \frac{min(\mu,1)}{\mu}}$***)***

Wrapped frequency = $\boldsymbol{f * \mu}$

#map frequency the wrapped frequency

Else

#Slow down the audio time series

Wrapped frequency = (sampling frequency)/2 –($\boldsymbol{f_n * f}$).

End If

#map frequency the wrapped frequency

return augmented Mel-filter bank

Algorithm 6- VTLP augmentation

4.3.2. Patient Medical History Data Pre-Processing

As it can be recalled from *section 3.4.2* different pre-processing techniques are mentioned to make the data suitable for prediction. This section detailed how the mentioned pre-processing techniques are used.

4.3.2.1. Data Cleaning

Inconsistent data Normalization: The chest column in the patient medical history dataset holds the types of respiratory sound. The event has been captured in several alike words or synonyms such as “crepitation”, “crackle” or “Rales” for crackle and “wheeze” or “ronchi” for wheeze sound. They are exactly the same but for the learning algorithm they seem different, hence row data can’t be represented differently. we have normalized the synonyms words as follows.

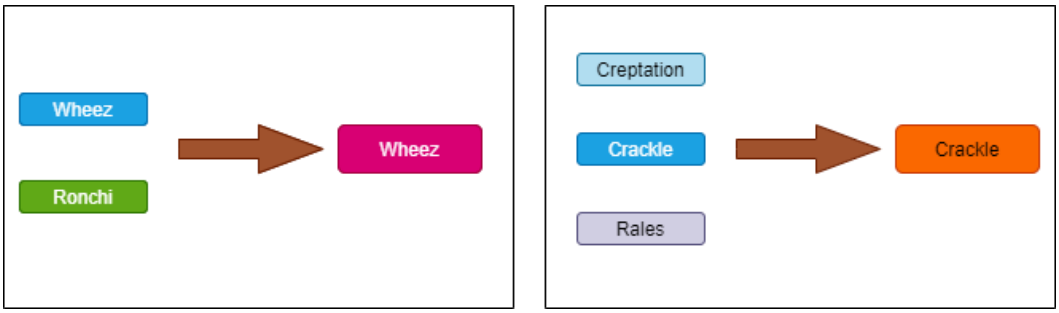


Figure 25- Chest Column Normalization

Unit Conversion: to make the variable of an attribute in the same unit of measurement we have used unit conversion. In our case patients below age 1 have the ages in months, which should be converted into year form with a very simple calculation of dividing each month by 12.

4.3.2.2. Data Transformation

As mentioned in *section 3.4.2* of the thesis, one hot and label encoding techniques are used. For ordinal data types, we have used label encoder in which it encodes the column by giving a rank of labels for each unique label. Unlike label encoder one hot encoder encodes by creating new column for each distinctive label. Our dataset is diversified by many feature modes. The table below keeps each feature along with its data type.

Table 4- Features and their data types

Parameter	***Symbol Used***	***Data Type***
Age	Age	Continuous
Sex	Sex	Categorical(Nominal)
Respiratory Rate	RR	Continuous
Pulse Rate	PR	Continuous
Saturation	So_2p	Continuous
Chest	Chest	Continuous(Nominal)
Cough	Cough	Categorical(Nominal)
Fever	Fever	Categorical(Ordinal)
Shortness of Breath	SOB	Categorical(Nominal)
Grunting	Gr	Categorical(Nominal)
History of Allergy	HxAllergy	Categorical(Nominal)
History of Smoking	HxSmoking	Categorical(Nominal)
History of Air Pollution	HxAirP	Categorical(Nominal)
Mal Nutrition	MN	Categorical(Nominal)
Diagnosis	Diagnosis	Categorical(Nominal)

4.3.2.3. Feature Selection

Once data cleaning is complete the entries on the dataset are considered reliable. The goal is for each patient to have the same event represented in the same manner for analysis. But, not all features play a big role in the prediction. To extract those features we have used an extra tree classifier algorithm. It creates many decision trees of the training dataset and the selection

process is held by majority voting of the decision trees. The algorithm outputs the value of each feature in relation to the class labels.

4.4. Model Building Algorithms

In order to build a respiratory disease prediction model, we have used DNN (Deep Neural Network) along with CNN (Convolutional Neural Network) algorithm to predict Pulmonary pathosis and respiratory sound type respectively. CNN is used for it is known to be the best Neural network for speech and sound processing(A. Khan et al., 2020). The detail of how we use those algorithms with their design and working principle is explained below.

4.4.1. Convolutional Neural Network for Respiratory Sound Classifier

The success of CNN algorithms in speech and audio processing is ahead of classical Machine learning algorithms and even ANN algorithms. Researchers suggest in using CNN for audio processing because of its major abilities in feature extraction and discrimination.

Respiratory sounds are sound signals that can be prepared and converted (using STFT) into spectrograms to be an input of the learning model. Taking the pulmonary sound wave as an input, the model used predicts either that respiratory sound is Wheeze, Crackle, Wheeze and Crackle, or Normal respiratory sound. The Architecture and design of the mentioned model is elucidated below.

Majorly CNN model we used is composed of Input layer, Convolutional layer, Fully Connected Layer and Output Layer, let's see each layer with their appearance in our model architecture.

Input Layer: the input layer of the CNN consumes the extracted features. The input features contain time-frequency patches with specified height and width. Therefore, the input has a shape of input height by input width.

Convolutional Layer: **the** heart of the CNN model is a convolutional layer. In the convolutional layer, a multi-level convolution operation is performed to extract the characteristics or features of the input using defined numbers and dimensions of kernel in each receptive field. In this layer, each neuron is used as a kernel. In building the CNN model we have used 7 convolutional layers and 2-layers of pooling or downsampling. The detail of each layer is described as follows.

Pooling Layer: The pooling layer is a layer in CNN models that happens next to feature extraction in the convolutional layer. After feature extraction, the location of the features is not that important, rather the aggregated information within its neighbors becomes presiding. In pooling layer down-sampling is performed for alike information in the neighborhood. The output of the pooling layer is the dominant information in that region (Demir, Turkoglu, et al., 2020). Because features are immutable for small distortion and translational shift. Pooling layer helps in reducing the size of feature maps for unaired feature set which in return helps by regulating the complexity of the model in increasing the model's ability to generalize and optimizing the model. Among other pooling mechanisms we have used MaxPooling because of its ability in extracting low level features.

Activation Function: Activation function helps the model to learn very complex patterns by serving as a decision function. There are different activation functions among them ReLu and Softmax are the famous activation functions used to ingrain features of non-linear combination. In the output layer we use Softmax activation function. However, in the middle of model function we used ReLu. The type of ReLu activation function we use is LeakyRelu. LeakyRelu is an advancement of ReLu that protects the model from dying neural connections or networks(Basu & Rana, 2020). Those neural networks usually dye because most of the weights become too negative and tends to have zero value.

$$f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x)$$

Equation 6- Leaky ReLu activation Function

Where: α is a small constant.

Fully Connected layer: At the end of the convolution and pooling layer all the extracted features are used as input to the fully connected layer. Fully connected layer is a dense feed-forward neural network that is connected from the last pooling layer, where the output of the pooling layer is flattened. Flatten makes the 3D-matrix in the above convolutional and pooling layer to be in a vector format(Dileep et al., 2020). Those vectors are inputs of fully connected layer. A fully connected layer with 4096 neurons with ReLu activation function and a dropout layer is used.

Output layer: next to the fully connected layer another layer with activation function of Softmax with 4 output nodes is used. The Softmax activation gives us the probability of each class in deciding the higher probability among them. The higher probability, the higher in deciding chance of input to fall in one of the given 4 classes.

ALGORITHM 7: ALGORITHM FOR RESPIRATORY SOUND CLASSIFICATION ON CNN

Input: Mel-Spectrogram of size(128,128)

Output: probability of Wheeze, Crackle, Both, Normal

Start

For(e >= 100)

//convolution

1. $c(i,j) = (mel * f)(7,11) = \sum_{m=1}^{128} \sum_{n=1}^{128} mel[m,n]\, f[\, i-m, j-n]$
2. $a = \max(0.1c[i,j], c[i,j])$
3. $c_1(i,j) = \left(\left((c[i,j] * f)(5,5) = \sum_{m=1}^{256} \sum_{n=1}^{256} c[m,n]\, f[\, i-m, j-n]\right)\right) a_i + b_i)$
4. $a^2 = \max(0.1c_1[i,j], c_1[i,j])$
5. $c_2 = max_{i=0,..256} c_1$

//dense layer

6. $z_i^1 = \sum_j^{4096} w_j^1\, c_j^1 + b_i$

   $a_i^1 = \max(0.0001 z_i^1, z_i^1)$

// output layer

7. j[1,4] $z_i^2 = \sum_j^{512} w_j^2\, a_j^1 + b_j$

   $p_i = \frac{e^{z_j}}{\sum_{j=1}^{4} e^{-z_j}}$ //softmax , $l(x,y) = \sum_i^4 y * \log(p_{i)}$

End

Algorithm 7- Respiratory Sound Classifier on CNN

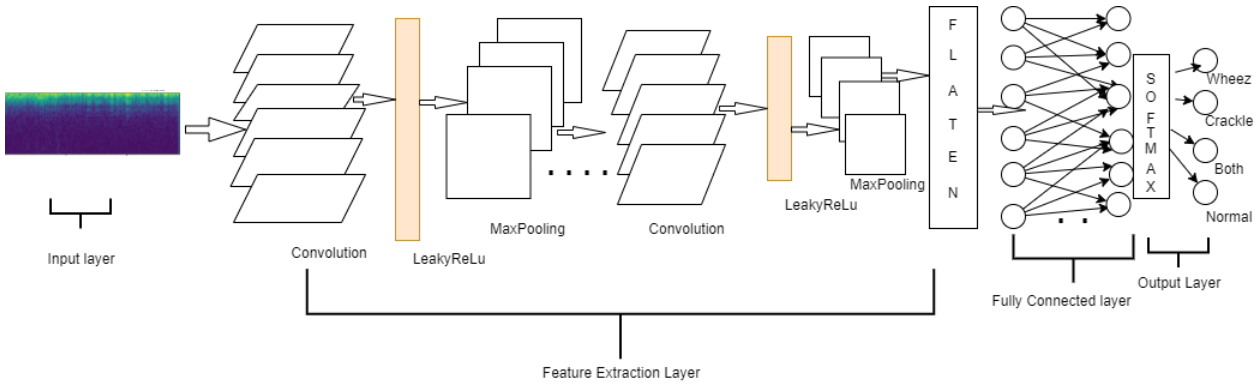


Figure 26: CNN model Architecture for Respiratory Sound Classifier

4.4.2. Deep Neural Network for Pulmonary Pathosis Classifier

Respiratory diseases prediction from patient medical history data is a multi-class classification problem. The medical history data could be prepared and appeared as structured or unstructured. In this work, we have prepared the dataset in a structured format. Taking the input structured data, the model used predicts either the patient is likely to have Asthma, Bronchiolitis, Pneumonia, or COPD. The design and structure of the DNN algorithm used is elaborated below.

Keras sequential model API that adds each layer sequentially has been used. The model is composed of an input layer with 20 nodes and 3 hidden layers. The first hidden layer contains 64 nodes, the next hidden layer contains 32 nodes, the third layer has 16 nodes and the last hidden layer is made up of 8 nodes. In all the hidden layers we use Relu and LeakyRelu activation functions interchangeably. The output layer holds 4 nodes with a sigmoid activation function, for sigmoid is the best activation function to be used in the output layer(Caliskan & Yuksel, 2017). The probabilistic output from the sigmoid activation function determines the output of the model. The design and architecture of the model is shown as follows.

ALGORITHM 8: ALGORITHM OF PULMONARY PATHOSIS CLASSIFIER ON DNN

Input: x with 20 nodes of symptoms

Output: probability of Asthma, Bronchiolit, COPD, Pneumonia

Start

For(numberofepoch<=50)

1. i[1,64], $o_i^1 = \sum_j^{20} w_j^1 \, x_j^1 + b_i$, $a_i^1 = \max(0, o_i^1)$

   // second layer

2. i[1,32], $o_i^2 = \sum_j^{64} w_j^2 \, a_j^1 + b_i$, $a_i^2 = \max(0, o_i^2)$

   // third layer

3. i[1,16], $o_i^3 = \sum_j^{32} w_j^3 \, a_j^2 + b_i$, $a_i^3 = \max(0, o_i^3)$

   // fourth layer

4. i[1,8], $o_i^4 = \sum_j^{16} w_j^4 \, a_j^3 + b_i$, $a_i^4 = \max(0, o_i^4)$

   // Output layer

5. i[1,4]$y_i = \sum_j^8 w_j^5 \, a_j^4 + b_i$, $p_i = \frac{1}{1-e^{-y_i}}$

6. $l(x, y) = \sum_i^4 y * \log(p_{i)}$ // categorical cross entropy

End

Algorithm 8- Pulmonary Pathosis Classifier on DNN

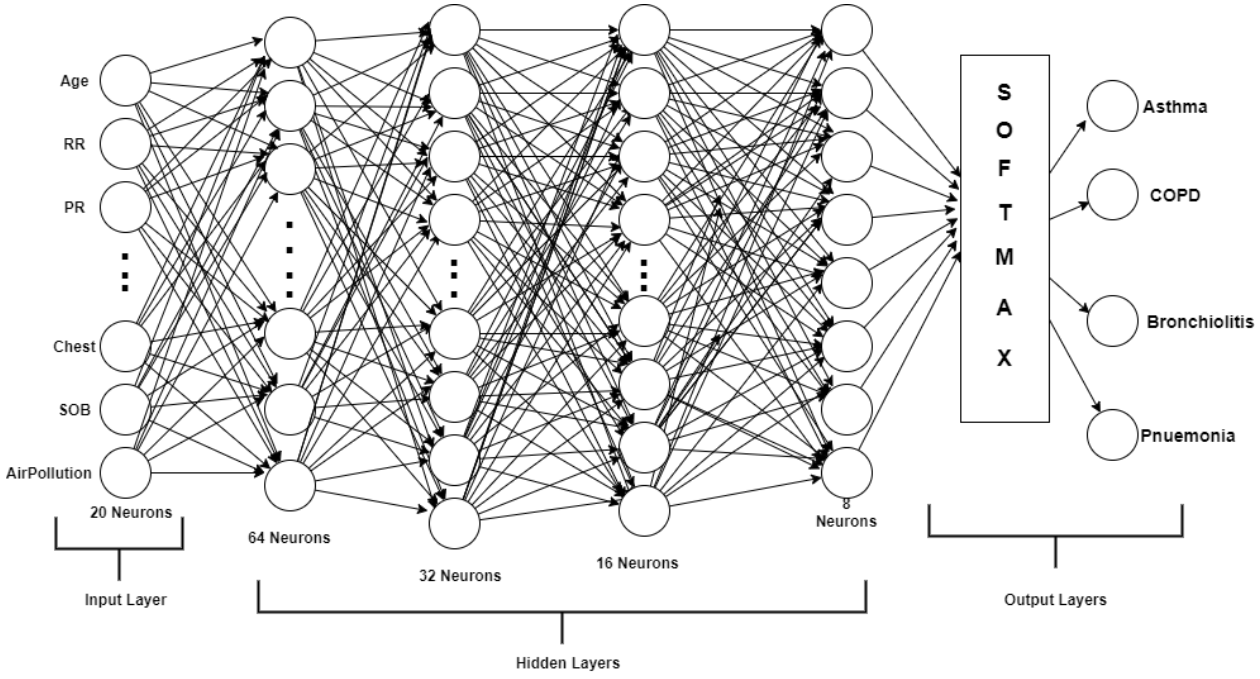


Figure 27: DNN Model for Pulmonary Pathosis Architecture

4.5. Proposed Model Prototyping Architecture

After building and training the models, the best among trained models is selected to be used for prototyping. The prototype is built for mobile devices that is because of its portability, ease to use, and privacy. Hence the model is running on the mobile device itself, internet connection and latency in connecting to the server are not issues. To develop this, the model we have built has to be converted into tensor flow lite as a special portable format of FlattBuffers[14] with .tflite file extension. Afterward, Tensor flow Lite Interpreter is used to make the.tflite model be compatible with different programing languages and platforms. Following this, the model is integrated with android studio to make it applicable for edge devices.

[14] https://google.github.io/flatbuffers

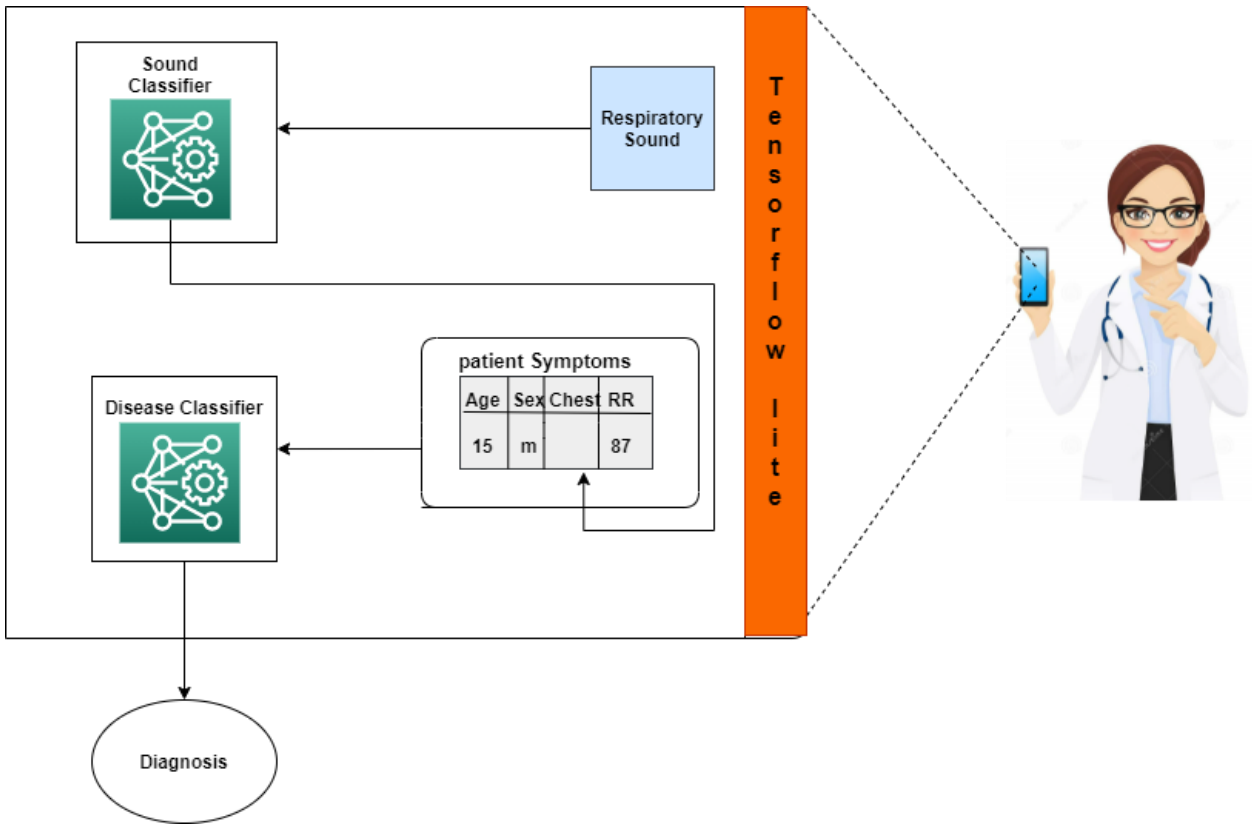


Figure 28-Prototype Architecture

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

In the previous chapter, we have seen the architecture, design, and working principle of the proposed respiratory disease prediction. This chapter resonates with the proposed solution in the form of actual code implementation. The implementation holds pre-processing, model building, testing and evaluation then finally prototyping. The detail of each implementation part is elucidated as follows.

5.2. Working Environment

Due to its popularity and richness in different libraries, we have chosen the python programming language to perform all the pre-processing and implementations. Python is an open-source language with plenty of contributors and support groups. In data science and machine learning, it is the leading programing language(Yli-heikkilä, 2019). Whereas java programming language in the android studio is used for prototyping.

5.2.1. Implementation and Deployment Environment

To implement the model, we have used google colab, it is a free GPU and TPU computation platform provided by Google. Google colab that offers Nivida K-80 GPU with 12 GB of RAM. It operates by connecting to the VMs (Virtual Machines) with a maximum lifetime of 12 hours in a certain period of time(Carneiro et al., 2018). Along with colab, a physical computer system with 8 GB RAM, Intel(R) Core i5, CPU 2.0 GHz Processor, Windows 10 64-bit operating system with NVIDIA GEFORCE graphics was used. The prototype was deployed using Android Studio v-4.2 and the physical android phone of android v.9 as a debugging tool.

5.2.2. Libraries and Header Files.

The modules are implemented using very helpful libraries and header files. Those libraries along with their respective version and description is presented.

Table 5- Libraries and header files

Library	***Version***	***Usage***
Keras	2.4.0	API for Python deep learning
Tensorflow	2.6.0	As a backend for Keras
Tflite	1.1.1	Framework for on-devise inference
Mathplotlib	3.4.3	Data visualization
Numpy	1.21.1	Multi-dimensional data manipulation
Panda	1.3.3	Data manipulation and visualization
Scipy	1.7.0	For statistical and mathematical routines
Seaborn	0.9.0	For statistical graphics
Sklearn.metrics	0.23.2	To calculate the model metrics
Librosa	0.8.1	For audio analysis

5.3. Respiratory sound classifier Implementation

In *sections 4.2* and *4.3* we have seen the design and working principle of the proposed respiratory sound classification solution. In this sub-section, we resonate the architecture with python code implementation. The implementation contains different pre-processing techniques as well as model building and evaluation. All the pre-processing and modeling algorithms are based on python and other built-in functions; where the necessary libraries used are loaded.

5.3.1. Pre-Processing Implementation

Pre-processing implementation includes reading the audio with annotated data, segmentation, resampling, and slicing followed by data augmentation to maintain the balance between different classes and simultaneously feature extraction. To perform all those preprocessing implementations we have started by loading the file containing audio data with its respective annotation file. Since we are working with colab, the data is obtained from google drive using the drive library.

5.3.1.1. Audio and annotation file reading

The annotation is a text file that comes up with the patient number escorted by its recording information (i.e. recording index, chest location, acquisition mode and recording equipment)

with the start and end timestamps of the portioned audio and their respective crackle and wheeze value. The annotation file is read and organized to match to the corresponding audio data.

As the audio data is in the wav file format we have read the data with the help of librosa library. The library returns the sample rate in samples per second and 32-bit NumPy array of the data. While reading, we haven't specified a constant sample rate, because the data set contains audio with different sample rates so that, we have preserved the self-sample rate of the audio data. However, we call the resample method to resample the data.

```
def read_wav_file(str_filename, target_rate):
    data,sample_rate = librosa.load(str_filename)
    if (sample_rate != target_rate):
         (_,data) = resample(sample_rate, data, target_rate)
    return (data, target_rate)
```

Code-Fragment 1- Reading Audio File

5.3.1.2. Audio Data Segment Implementation

The annotation file trims the value of crackle and wheeze in a certain non-uniform timestamp. To interpret the annotation of the audio files, we have segmented each audio file based on the start and end time of the annotated pair.

```
def segment_data(start, end, raw_data,  sample_rate):
    max_ind = len(raw_data)
    start_ind = min(int(start * sample_rate), max_ind)
    end_ind = min(int(end * sample_rate), max_ind)
    return raw_data[start_ind: end_ind]
```

Code-Fragment 2-Audio Data Segmentation

5.3.1.3. Audio Data Resampling Implementation

The audio samples do not have an equal sampling rate; but having a uniform sampling rate is very crucial for further pre-processing and model training. Here we implement the re-sampling of all the audio samples that have different sampling rate than the target sampling rate. Where target rate is the standard sampling rate suitable for our model training.

Interpolation: is estimation and analysis of numerical data that find new data points depend on the range of known data points linearly.

Linear spacing: is a way of spacing numerical arrays in a customized span, evenly or non-evenly. To meet the specified need, the 32-bit NumPy array of the sound file is spaced with 0-100 range.

```
def resample(current_rate, data, target_rate):
    x_original = np.linspace(0,100,len(data))
    x_resampled = np.linspace(0,100, int(len(data) * (target_rate / current_rate)))
    resampled = np.interp(x_resampled, x_original, data)
    return (target_rate, resampled.astype(np.float32))
```

Code-Fragment 3- Resampling Audio Data

5.3.1.4. Audio Data Slicing Implementation

Having a commensurate amount of data helps to execute the feature extraction and training in a good way. Because the duration is one of the factors that determine the size of audio data, which eventually sets the input size of the model. On the basis of that, audio data that has been interpreted in the form of array should have an equal span. Therefore, we have sliced each audio segment with 5 sec each. The python code to implement slicing is shown below.

```
def  slice_data(original, desiredLength, sampleRate):
    output_buffer_length = int(desiredLength * sampleRate)
    soundclip = original[0]
    n_samples = len(soundclip)
    total_length = n_samples / sampleRate #length of cycle in second
    n_slices = int(math.ceil(total_length / desiredLength))
    samples_per_slice = n_samples // n_slices
    #//floor devision
    src_start = 0
    output = [] #Holds the resultant slices
    for i in range(n_slices):
        src_end = min(src_start + samples_per_slice, n_samples)
        length = src_end - src_start
        copy = generate_padded_samples(soundclip[src_start:src_end],
                                       output_buffer_length)
        output.append((copy, original[1], original[2]))
        src_start += length
    return output
```

Code-Fragment 4- Audio Data Slicing

Considering the heterogeneity and variations in recording length, we get different reminders from the slice data. Therefore we have padded or filled the data less than 5sec with zeros using numpy zeros() function.

```
def generate_padded_samples(source, output_length):
    copy = np.zeros(output_length, dtype = np.float32)
    src_length = len(source)
    frac = src_length / output_length
    if(frac < 0.5):
        #tile forward sounds to fill empty space
        cursor = 0
        while(cursor + src_length) < output_length:
            copy[cursor:(cursor + src_length)] = source[:]
            cursor += src_length
    else:
        copy[:src_length] = source[:]
    #
    return copy
```

Code-Fragment 5- Padding the Audio Data

5.3.1.5. Augmentation

Augmentation is applied using librosa library, the library takes an audio file and corresponding annotation file in a multi-dimensional array format and outputs the determined audio together with enhanced or augmented data containing all the parameters used for deformation.

Time_stretch: As shown in the implementation below, to perform time stretch we have used the effects under librosa library. Since time stretch is implemented on the wave form of audio file, the time stretch function uses that wave form and a maximum percentage change of stretch as an input. The implementation returns the stretch audio wave form.

```
#Creates a copy of each time slice, and stretche or contract it
def time_stretch(original, sample_rate, max_percent_change):
    stretch_amount = max_percent_change / 100
    stretched = librosa.effects.time_stretch(original,stretch_amount)
    return stretched
```

Code-Fragment 6- Augmentation with Time Stretch

VTLP: The execution of VTLP shift is applied on the Mel-filter bank of the audio signal. Give the Mel-filter bank the VTLP shift returns it with a wrapped frequency.

```
import nlpaug.augmenter.audio as aug
augmenting_factor = aug.VtlpAug(sr)
augmented_audio_data = augmenting_factor.augment(data)
```

Code-Fragment 7- VTLP Augmentation

Spec-Aug: To implement the augmentation we have used spec_augment tensorflow library from SpecAugment. Because specAugment is exerted directly on Mel-spectrogram, we have passed the Mel-spectrogram to sepc_augmentation function. The working code fragment is shown below.

```
from specAugment import spec_augment_tensorflow

def Spec_augmentation(mel_spectrogram)
     warped_masked_spectrogram = spec_augment_tensorflow.spec_augment(mel_spectrogram=mel_spectrogram)
     print(warped_masked_spectrogram)
```

Code-Fragment 8- Augmentation with Spec-Aug

5.3.1.6. Feature extraction

Mel-spectrogram: In order to extract this feature we have utilized features.Mel spectrogram from librosa library. Parameters such as hop length and number of FFT has been used in their default values. Taking the audio signal with its respective sample rate as an input module. The output from this implementation is a Mel-scaled spectrogram. The python code implementation is shown below.

```
def MelSpectrogram_extraction(cycle_info, sample_rate, n_filters, vtlp_params):
    mel_spect = librosa.feature.melspectrogram(y=cycle_info[0], sr=sample_rate, n_fft=2048, hop_length=1024)
    mel_spect = librosa.power_to_db(mel_spect, ref=np.max)
    labels = [cycle_info[1], cycle_info[2]] #crackles, wheezes flags from the annotation
    return (mel_spect, label2onehot(labels))
      # 196x64x1 matrix
```

Code-Fragment 9- Mel-spectrogram Feature Extraction

MFCC: Considering the number of MFCCs and the audio signal with a specific sample rate, the librosa library computes the MFCC feature for each audio bundle. The implementation is shown below.

```
def Mfcc_extraction(cycle_info, sample_rate, n_mfcc):
    mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc)
    labels = [cycle_info[1], cycle_info[2]] #crackles, wheezes
    return (mfcc, label2onehot(labels))
```

Code-Fragment 10- MFCC Feature Extraction

5.3.2. CNN Model Implementation

Based on the architecture specified in section 4.3.1 of this thesis. The CNN model we have used for respiratory sound prediction is implemented with keras library. Sample code from the implementation is shown below.

```
model = Sequential()
model.add(Conv2D(128, [7,11], strides = [2,2], padding = 'SAME',
                 input_shape = (sample_height, sample_width, 1)))
model.add(LeakyReLU(alpha = 0.1))
model.add(MaxPool2D(padding = 'SAME'))
```

```
model.add(Conv2D(512, [1,1], padding = 'SAME', activation = 'relu'))
model.add(Conv2D(512, [3,3], padding = 'SAME',activation = 'relu'))
model.add(Conv2D(512, [1,1], padding = 'SAME'))
model.add(Conv2D(512, [3,3], padding = 'SAME', activation = 'relu'))
model.add(MaxPool2D(padding = 'SAME'))
model.add(Flatten())
```

```
model.add(Dense(512, activation = 'relu'))
model.add(Dense(4, activation = 'softmax'))

opt = tf.keras.optimizers.Adam(learning_rate=0.0001,
                               beta_1=0.9, beta_2=0.999,
                               epsilon=None, decay=0.00,
                               amsgrad=False)
model.compile(optimizer =  opt , loss = 'categorical_crossentropy',
              metrics = ['acc'])
```

Code-Fragment 11- CNN model Implementation

5.3.2.1. Model Testing and Evaluation

For the ease of data loading to the model, we have used Keras data generator library. Then the model is fed with the train and validation data using the generator. After fitting the model, we have used confusion matrix to see how well our model performs. All those implementations are shown below.

```
stats = model.fit(train_gen.generate_keras(batch_size),
                  steps_per_epoch = train_gen.n_available_samples() // batch_size,
                  validation_data = test_gen.generate_keras(batch_size),
                  validation_steps = test_gen.n_available_samples() // batch_size,
                  epochs = n_epochs)
```

```
test_set = test_gen.generate_keras(test_gen.n_available_samples()).__next__()
predictions = model.predict(test_set[0])
predictions = np.argmax(predictions, axis = 1)
labels = np.argmax(test_set[1], axis = 1)
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(labels, predictions, target_names = ['none','crackles','wheezes','both']))
print(confusion_matrix(labels, predictions))
```

Code-Fragment 12- Implementation of CNN Model Testing and Evaluation

5.4. Pulmonary Pathosis classifier Implementation

In *section 4.3.2* and *section 4.4.2* we have seen the pre-processing and model building architecture of pulmonary pathosis classification. In this section, we have shown the implementation details for each module.

5.4.1. Pre-processing implementation

In order to get the most out of the dataset collected, we have implemented different pre-processing techniques. Cleaning and normalization of duplicated data as well as handling inconsistent values and feature transformation followed by feature selection are the core pre-processing procedures. But to start with all the pre-processing steps the dataset must be loaded to the working environment. Since we are working with colab the dataset is retrieved from google derive using the drive library from google.colab package. Hereafter python's panda library is used to store the csv file into data frames as a vector form. The data retrieval and loading implementation can be seen as follows.

```
from google.colab import drive
drive.mount('/content/drive/')
file=('/content/drive/My Drive/PHxDataset1.csv')
```

```
import pandas as pd
import tensorflow as tf

df = pd.read_csv(file)
print(df.columns) # show columns
print(len(df))
```

Code-Fragment 13- Patient Symptom Data Loading

5.4.1.1. Data Cleaning and Normalization Implementation

Data cleaning includes eliminating duplicated rows and missing values. In this implementation step, the loaded dataset is checked for duplication using python's built-in **duplicate()** function. The function is able to determine and display the duplicated rows. **Isnull()** is another python's built in function to find out null vales in the dataset. Next, normalization of synonym words in chest column is performed.

5.4.1.2. Data Transformation

Since Machine learning models are bundles of mathematical operation; non-numerical data such as categorical ordinal and categorical nominal data should be converted into numeric format. To perform this, one-hot encoder is implemented for categorical nominal data types using **get_dummies()** function. Whereas categorical ordinal data types have been transformed with **labelencoder()** function.

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
```

```
#labels to implement label_encoder
labels = ['Sex','Cough','Fever','SOB','Grunting','HxAlergy','HxSmoking','MalNutrition']
```

```
label_encoder = LabelEncoder()
x[labels]= x[labels].apply(label_encoder.fit_transform)
```

```
from sklearn.preprocessing import OneHotEncoder
#encodes the column with 1 and 0
x = pd.get_dummies(x,columns=['Chest'])
```

```
x = pd.get_dummies(x,columns=['AirPollution'])
```

Code-Fragment 14- Implementation of Data Transformation

5.4.1.3. Feature Selection

After the data is transformed into numbers, columns transformed with one-hot encoding tends to have a new column for each different entry; which in return increases the column size. The transformed dataset contains 20 independent features but all the 20 features does not contribute equally for the prediction model. Therefore, feature selection is used to rank the importance of the feature for overall model prediction. We have identified the most and list important features using the **ExtraTreesClassifier** ensemble model of decision trees library.

```
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(x,y)
feature_importance = model.feature_importances
```

```
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=x.columns)
feat_importances.nlargest(20).plot(kind='barh', color = list('gggggggkkkkkkkkkkkk'))
plt.show()
```

Code-Fragment 15- Implementation of Feature Selection

5.4.1.4. Data Utility Function

The data set is composed of several independent features and a dependent attribute. To train the model we have teared the dataset in to x- independent features and y-dependent attribute by dropping the diagnosis column and assigning the diagnosis column to y.

```
#separating as dependent and independet feature
y=df['Diagnosis']  #Independet Feature
len(y)
x=df.drop(columns=['Diagnosis'])    #droping the independent feature
len(x)
```

Code-Fragment 16- Implementation of Data Utility

5.4.2. DNN Model Implementation

To construct the DNN classifier algorithm Keras sequential API model with tensor flow at the backend has been used. Since the model is sequential we have added layers using add function, which will append layers on the top of the previous layer. The input dimension, number of nodes in each layer and the activation function are expressed while adding the layers.

After constructing the model structure compiling the model is the next step; compiling a model assigns a loose function optimizer and metrics for model prediction. We have compiled our model using a loss of categorical-cross-entropy since the problem is multi-class classification. Adam optimizer is used with an accuracy metrics. Then the compile module compiles the model with the parameters we have sated.

Fitting a model measures its ability to generalize on unseen data. Keras' model.fit() module takes the training data to train the model and validation data to assess the model's ability. Rather than fitting all the training data at once, we have used batch_size to train the data in batches which decreases the computation complexity. Since we use K-fold cross-validation the model re-runs for each fold iteration. The python code to implement the proposed DNN model is shown below.

```
fold_no = 1
for train, test in kfold.split(x, y):
    model = Sequential()
    model.add(Dense(64, input_dim=18, activation='relu'))
    model.add(Dense(32, activation= 'relu'))
    model.add(Dense(16,activation='relu'))
    model.add(Dense(8,activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    #compile the model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    # Generate a print
    print('------------------------------------------------------------------------')
    print(f'Training for fold {fold_no} ...')

    # Fit data to model
    history = model.fit(x.loc[train], y[train],
              batch_size=batch_size,
              epochs=no_epochs,
              verbose=verbosity)

    # Generate generalization metrics
    scores = model.evaluate(x.iloc[test], y[test], verbose=0)
    print(f'Score for fold {fold_no}: {model.metrics_names[0]} of {scores[0]};
           {model.metrics_names[1]} of {scores[1]*100}%')
    acc_per_fold.append(scores[1] * 100)
    loss_per_fold.append(scores[0])

    # Increase fold number
    fold_no = fold_no + 1
```

Code-Fragment 17- Implementation of the DNN Model

5.4.2.1. Model Testing and Evaluation

For model testing and evaluation, K-fold cross-validation is implemented. Holding the performance value in each fold iteration, the model testing and evaluation returns a confusion matrix with the corresponding accuracy for each class.

```
#y_unique = y_test.unique()
cm = confusion_matrix(y_test_new,y_pred_new)
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
cm_df = pd.DataFrame(cm, index =['Asthma','Bronchiolit','COPD','Pnumonia'],
                     columns =['Asthma','Bronchiolit','COPD','Pnumonia'])
#
plt.figure(figsize=(5,4))
sns.heatmap(cm_df,annot=True,fmt='.2%',cmap='copper')
#sns.heatmap(cm/np.sum(cm), annot=cm_df,
 #           fmt='.2%', cmap='Blues')
labels = ['Asthma','Bronchioliti','COPD','Pnumonia']
labels = np.asarray(labels).reshape(4,1)
#sns.heatmap(cm, annot=labels, fmt='', cmap='Blues')
```

Code-Fragment 18- Testing and Evaluation for DNN Model

5.5. prototype Implementation

The core process of prototype implementation is making the model compatible for the prototyping devise, in our case android system. To do that, the implemented Keras model needs to be converted into tflite file extension. After interpreting the converted tflite module, the model is set for end-device execution.

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
tf_lite_model_name =tf_lite_name
open(tf_lite_model_name, "wb"). write(tflite_model)
```

```
interpreter = tf.lite.Interpreter(model_path=tf_lite_name)
```

Code-Fragment 19- Prototype Implementation.

CHAPTER SIX

6. RESULTS AND DISCUSSIONS

6.1. Chapter Overview

So far we have seen the problem statement, the methodology we use, the proposed solution with its architecture, and implementations in detail. In this chapter, the result obtained from executions and experimentation with different hyperparameter tunings, feature extraction, and augmentation methods is elaborated. Also, discussion on the outcome and overall accomplishment of the proposed solution along with expert evaluation and comparison with previous works are articulated.

6.2. Respiratory Disease Prediction

Respiratory disease prediction is one of the application areas of deep learning in medicine. The prediction is comprised of pulmonary pathosis classification and adventitious respiratory sound prediction, which predicts the presence and absence of adventitious sounds along with the type of respiratory sound. The outcome from respiratory sound prediction is used as a symptom or feature to be included with other patient symptom data. The combination of all the features leads to well-observed respiratory disease prediction. The result obtained from the sound classification model, pathosis classification model, and the overall prediction performance is explained below.

6.2.1. Respiratory sound classifier

Breath sound or respiratory sound contains crucial information about the type of adventitious sound the person is having. The CNN model uses that information to learn about the data and make predictions accordingly. It is believed that the model's learning process is highly altered by feature extraction and augmentation techniques used. For feature extraction MFCC and Mel-spectrogram features; also for augmentation time stretch, Spec-Aug, and VTLP techniques have been experimented. The experiment is conducted in hierarchical form, which means the experiment with a promising outcome is used as a base for the next experiment. The result from each inspection is elucidated below.

6.2.1.1. Augmentation Results

To recall from *chapter 4 section 4.3.1.5* of this thesis, without the data loses its class one can augment it to make it in a different form. Therefore, the model would think of it as new data. The augmentation techniques used in this paper are Time-shift, Spec-Aug, and VTLP shift. To preserve the best result, we have observed the effect of each augmentation technique. Since the classes with fewer data are wheeze and Both sound, we have done 4 experiments to augment each data using the mentioned augmentation techniques.

Experiment 1: time stretch or Time-shift augmentation technique was applied to a few waveform features of the two Wheeze and Both classes because they need to have an adequate amount of data. In order not to lose some information, we set the maximum percentage change to be 10. Using this the Time -stretch stretches the signal in a positive (Speed up) and stretches to the negative (slow down) by randomly choosing the stretching factor. Following that the model's overall accuracy becomes 68.0%.

Experiment 2: this experiment was observed using only the Spec-Aug augmentation technique. Randomly chosen Mel-spectrograms from Wheeze class and Both class has been augmented with frequency and time masking. The output from employed Spec-Aug is accuracy of 54.7%. However, there was an improvement in the model training speed than other experiments.

Experiment 3: VTLP augments the data by wrapping the frequency with a wrapping factor. The wrapping factor we use is random in the range of 0.9 and 1.1 because it is assumed that the best wrapping factor for VTLP lays in between(Jaitly & Hinton, 2013). Since frequency wrapping is very fast it leads the model to get a good accuracy of 69.5%.

Experiment 4: here the experiment was done by combining the features. Since the Spec-Aug gives us the low result, we have done the 4^{th} experiment using Time-stretch and VTLP shift augmentations. This aggregated augmentation technique gives us an accuracy of 70.0%.
Lastly the result obtained from each augmentation experiment is shown in the table below.

Table 6- Augmentation Experiment Result

Experiment	***Augmentation Used***	***Accuracy***
1	Time-Stretch	68.0%

2	Spec-Aug	54.7%
3	VTLP	69.5%
4	Time-Stretch + VTLP	**70.0%**

6.2.1.2. Feature Extraction Results

As mentioned in the literature review section of this thesis; previous researchers use time-domain features and frequency domain alone; as an improvement to this time-frequency domain, features started to be used. Many researchers in lung sound analysis use MFCC features and it has shown a great improvement. However, researchers in sound and audio analysis mentioned that Mel-spectrogram features are performing very powerfully in the time-frequency domain(Huzaifah, 2020). Therefore, to exploit the effect of Mel-spectrogram features in respiratory sound prediction; we have done 3 experiments.

Experiment 1: the first experiment was based on MFCC features. To get the optimal number of MFCC we have observed the effect of different MFCC numbers. Making 40 and 13 as a boundary we have selected 2 random numbers in between to see the effect. The result shows that as the number of MFCC increases the model's performance also increases. when the number of MFCC is 40 the model gives good accuracy but the training time it takes is 17 minutes, while MFCC of 30 gives the same accuracy with 10 minutes. Therefore, we have chosen MFCC of 30. The number of MFCC along with its accuracy is shown in the table below.

Table 7- Number of MFCC and its effect on the performance

Number of MFCC	**Accuracy**	**Training time**
40	68.5 %	17 min
30	68.5 %	10 min
20	62.5%	8 min
13	68.0%	7 min

Experiment 2: - the second experiment considers Mel-spectrogram for feature extraction. As the result obtained shows, Mel-spectrogram features give a good performance point. By varying the value of Mel-spectrogram hyperparameters arbitrarily, we have obtained poor results. While the number of the hyperparameters is stetted to the default the model gives an accuracy of 70.0%.

Experiment 3: - in this experiment both MFCC and Mel-spectrogram features have been used. It accommodates MFCC feature extraction with their best performing variables also Mel-spectrogram feature extraction with the finest default parameters. By concatenating the two models we have obtained an average accuracy of 71.2%.

Table 8- Feature Extraction Experiment Results

Experiment	***Feature Used***	***Accuracy***
1	Mel-spectrogram	70.0%
2	MFCC	68.5%
3	Mel-spectrogram + MFCC	**71.2%**

6.2.1.3. Feature Extraction and Augmentation Results

As inferred from the above experiments feature extraction of respiratory sound benefits from the combination of Mel-spectrogram and MFCC features. Also in the augmentation experiment, the best result obtained was when time-shift is combined with VTLP. Since the augmentation and feature extractions are coinciding with one another, experiments done with feature extraction includes augmentation techniques and vice versa. This indeed helps us in choosing the best feature extraction along with the augmentation technique used. The following table shows the summary of experiments in the features and augmentation techniques used, together with the achieved accuracy in each operation.

Table 9- Feature Extraction and Augmentation Results

Experiment	Feature Extraction		Augmentation			Accuracy
	MFCC	Mel-spectrogram	Time-stretch	Spec-Aug	VTLP	
1	X		X		X	68.5%
2		X	X	X		54.7%
3	X	X	X			69.0%
4	X				X	69.98%
5		X	X		X	70.0%
6		X		X	X	53.45%
7		X	X	X		54.0%
8	X	X	X		X	**71.2%**

Having seen the effect of features and augmentation techniques, we have chosen the best performing model with its accuracy score. However, there are several hyperparameters that need to be tuned. The result of the tuning is explained below.

6.2.1.4. Hyperparameter Tuning for Respiratory Sound Classifier Model

To exploit the ability of the model hyperparameters should be selected consciously. There are different algorithms to tune hyperparameters one of them is the Grid search algorithm. Since Grid search goes through all the possible combinations of hyperparameters it consumes computational time. Therefore, we have applied a manual hyperparameter tuning method is applied by leaning on the shoulder of the common rule of thumps.

Table 10- Clusters of Hyper Parameter for Tuning

Hyper parameter Tuning Clusters						
	Lr	***Af***	***Ks***	***Dp***	***Bs***	***Ep***
Cluster 1	0.0001	Softmax	3 x 3	0.6	200	30
Cluster 2	0.0001	Relu	5 x 5	0.5	128	35
Cluster 3	0.001	ReLu	7 x 7	0.3	128	50
Cluster 4	0.00001	Softmax	5 x 5	0.4	80	40

Cluster 5	0.001	Softmax	3 x 3	0.5	128	60
Cluster 6	0.001	Relu	5x5	0.5	110	27
Cluster 7	0.01	Softmax	3 x 3	0.3	128	100

Where: Lr is Learning Rate, Af is Activation Function, Ks is kernel Size, Dp is Dropout, Bs is batch Size and Ep is the number of epochs.

This demonstration hunts for the best combination of hyper parameters. The result on Manual tuning using the above-stated cluster or bundle of variables for hyper parameters is shown below as overall accuracy and average F1-score of each classes.

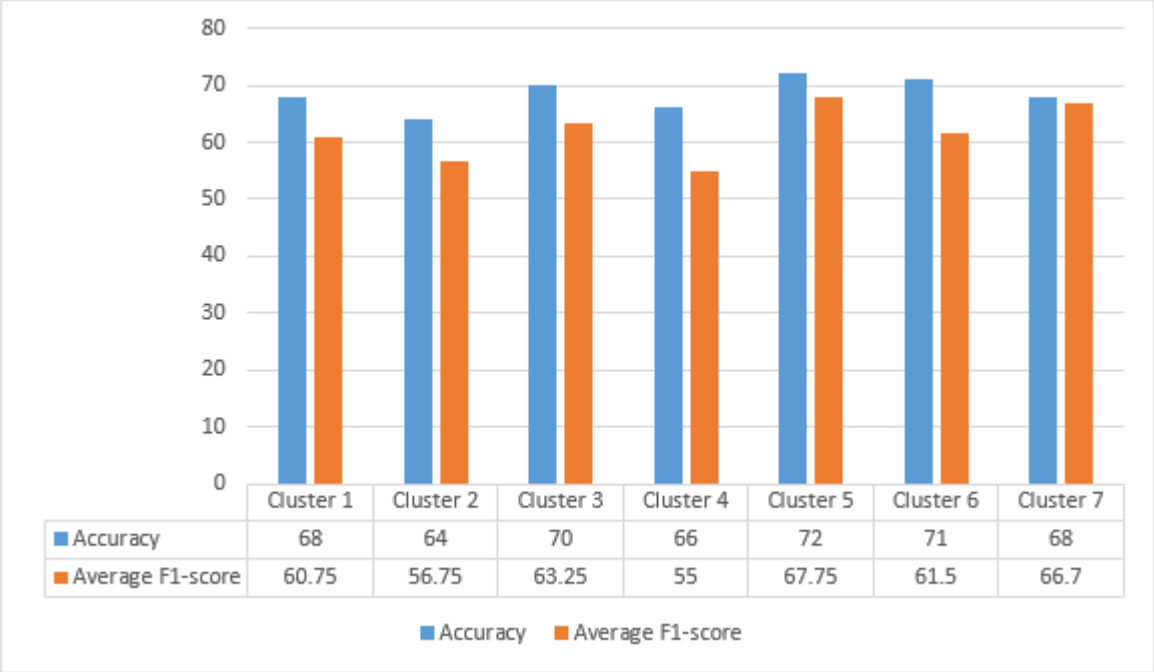


Figure 29 - Performance of each Hyper Parameter Cluster

As shown above, among the inspected cluster of hyperparameters cluster 5 with learning rate 0.001, activation function Softmax, kernel size 3 x 3, dropout 0.5 and number of epochs 60 gives the highest accuracy. Therefore, we have chosen it as the final respiratory sound classifier. The cross validation reveals that the model works good for Crackle and None classes but shows modest accuracy for Wheeze and Both classes.The cross validation result is shown below.

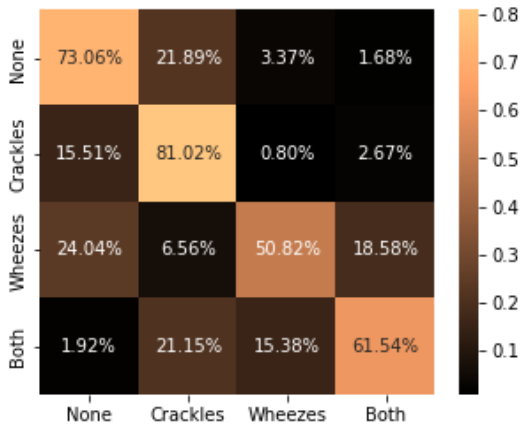


Figure 30- Cross Validation Result of Respiratory Sound Classifier

One way of checking if the model over fits or under fit is using the loose curve of the model. when we first train our model without using regularization technique moderate overfitting problem was seen. Then we have applied L2 regularization technique as a kernel regularize and bias regularizer. The loss curve of the model before and after applying L2 regularization technique is shown as follows.

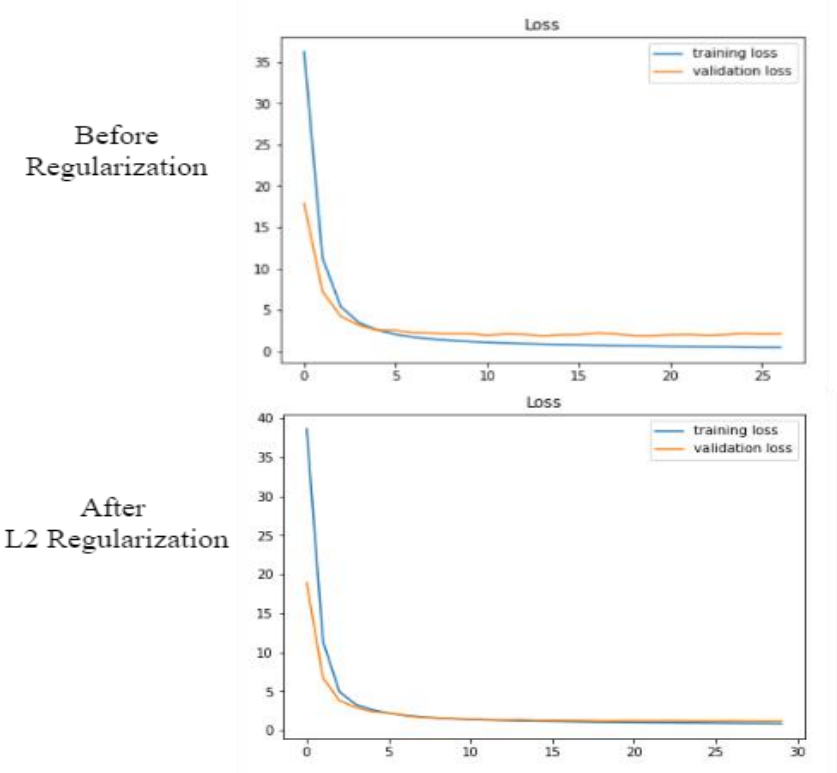


Figure 31- Loose Curve Before and After Regularization

6.2.2. Pulmonary Pathosis Classifier Result

Taking the patient's symptoms history along with the respiratory sound type; the pulmonary pathosis classifier classifies the type of pulmonary pathosis. To obtain the best result from the learning model we have run different methods and techniques that are believed to have an impact on the accuracy and overall performance of the model. First, we have demonstrated the importance of feature selection. Then by interchanging the number of k in K-fold cross-validation escorted by hyperparameter tuning results are delineated.

Data pre-processing is a very crucial activity in which the model may not work if some pre-processing such as data-transformations are applied, but there are other pre-processing steps in which we implement them because they are believed to affect the performance of the model in the most positive way. Originally the dataset we used has 21 attributes 20 independent features and 1 dependent feature. All of the attributes do not have an equal impact on the dependent feature. Using the features selection mechanism, we have extracted the effects of each attribute and find out the least relevant.

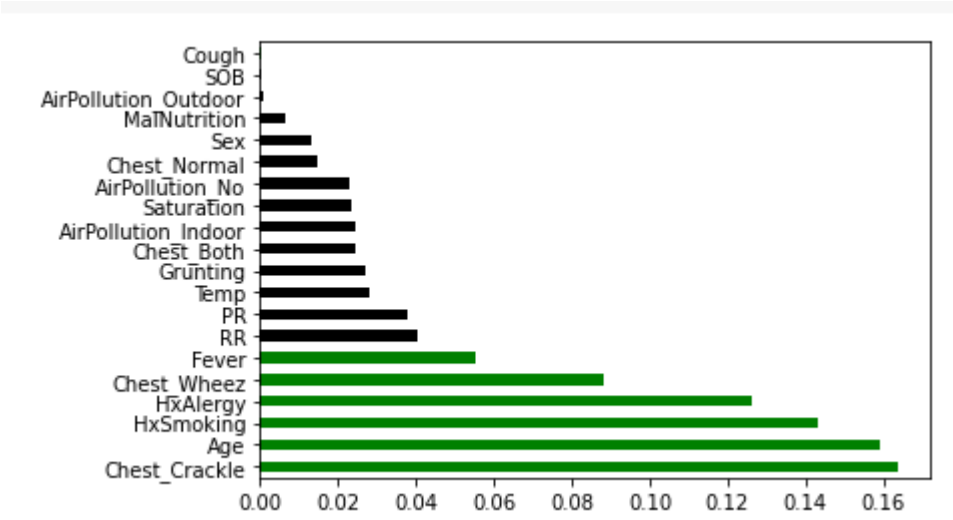


Figure 32- Result of Feature Selection

Based on our feature selection we have taken out the columns that have 0.00 relationship with the dependent variable “Diagnosis”. Based on this, we have removed (shortness of breath) “SOB” and “Cough” and examine the effect of feature selection on the performance of the algorithm. As shown in the table below, the accuracy and computational time of the model

changed due to changes in the number of features. Therefore, we excluded the 2 less important features.

Table 11- Effect of Feature Selection

Input Dimension	Accuracy	Computational Time
20	95.8	2 minutes
18	96.0	1 minutes

6.2.2.1. K-Fold Cross Validation Results with Different K-Values

As we have discussed in *section 3.6* of this thesis, one way of exploiting the model's performance and overcoming the overfitting problem is using k-Fold cross-validation. But the perfect value of k is not well known, therefore we have trained our model with different k-values and select the best among them. K-values can be any positive integers but, the default in Sklearn is 3. However, as other researchers suggest 3,5, and 10 are the most impactful(Broenlee, n.d.). The result on the confusion matrix shows the actual and predicted value of the model's performance. The result is based on the class level and interpreted as follows.

When k=3: the whole dataset is split into 3 subclasses in which the first two classes are used as training sets for the first iteration and the next iteration continuous till it touches all the clusters as training and test sets. The cross-validation experiment k=3, has got less performance in Bronchiolitis and Asthma class but gives a good accuracy for Pneumonia class. In addition, an overall accuracy of 95% is obtained. The confusion metrics as well as the classification report is shown below.

Table 12- Classification Report for K=3

Class	Precision	Recall	F1-score	Overall Accuracy
Asthma	0.94	0.85	0.89	
Bronchiolitis	0.91	0.97	0.94	
COPD	0.95	0.98	0.97	
Pneumonia	1.00	0.98	0.99	
				0.95

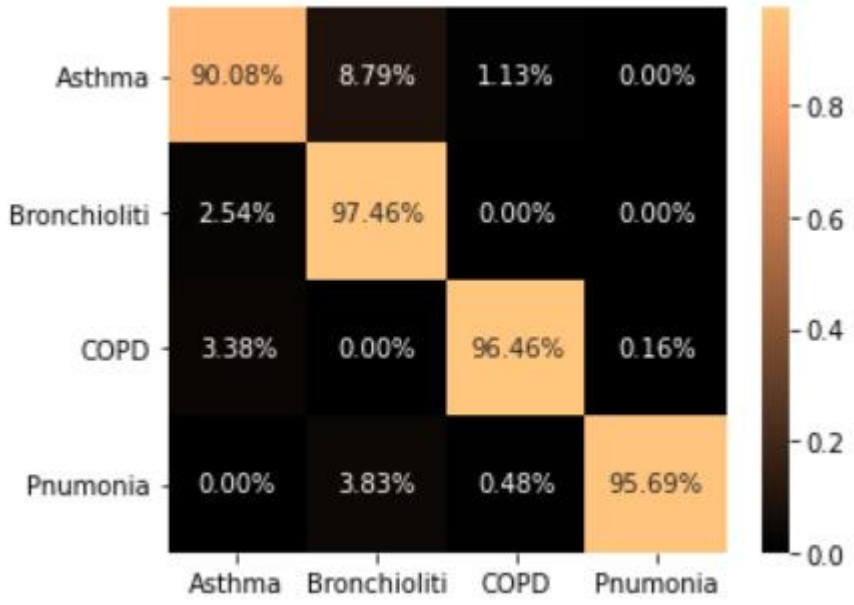


Figure 33- Cross-Validation Result for K=3

When k =5: The dataset that has been split for 5 folds iterates to make the prediction by assigning each fold as a test and the rest as a training. The outcome from this experiment shows that the model performs best for classes pneumonia and COPD whereas for classes Asthma and Bronchiolitis the model performs slightly low. The overall performance of the model making the k fold 5 times gives an accuracy of 96%. The confusion matrix and the classification report is shown below.

Table 13- Classification Report for K=5

Class	**Precision**	**Recall**	**F1-score**	**Overall Accuracy**
Asthma	0.90	0.93	0.92	
Bronchiolitis	0.93	0.95	0.94	
COPD	0.99	0.97	0.98	
Pneumonia	1.00	0.98	0.99	
				0.96

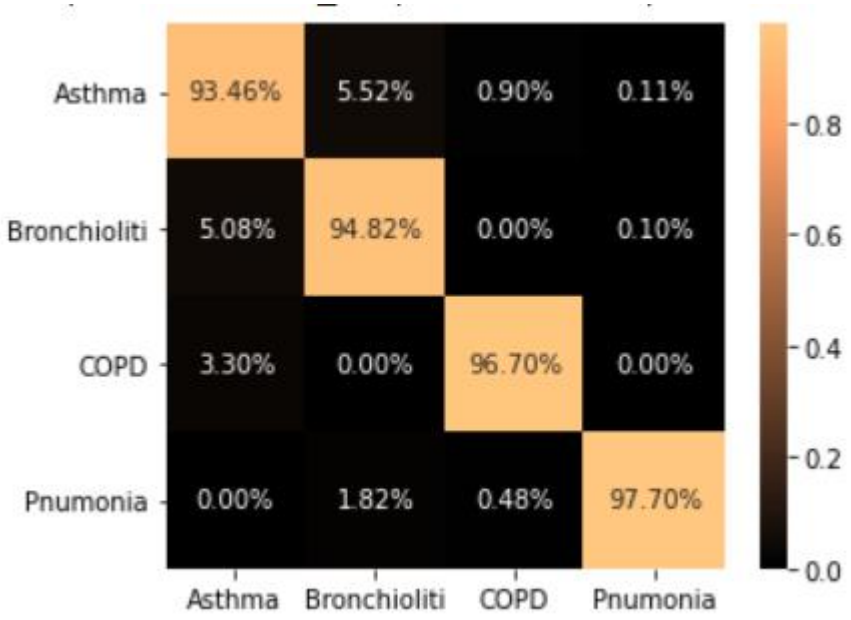


Figure 34- Cross-Validation Result for K=5

When k =10: The same as other K-fold cross validations, 10-fold have 10 evaluation loop. In each loop it swaps the test set between each folds. The evaluation from each fold is aggregated and the average is considered. As shown in the confusion matrix and the classification report this experiment shows good achievement except for Bronchiolitis class. Finally, 95% is observed as the average accuracy for all the classes.

Table 14- Classification Report for K=10

Class	**Precision**	**Recall**	**F1-score**	**Overall Accuracy**
Asthma	0.94	0.84	0.89	
Bronchiolitis	0.88	0.99	0.93	
COPD	0.99	0.97	0.98	
Pneumonia	1.00	0.99	0.99	
				0.95

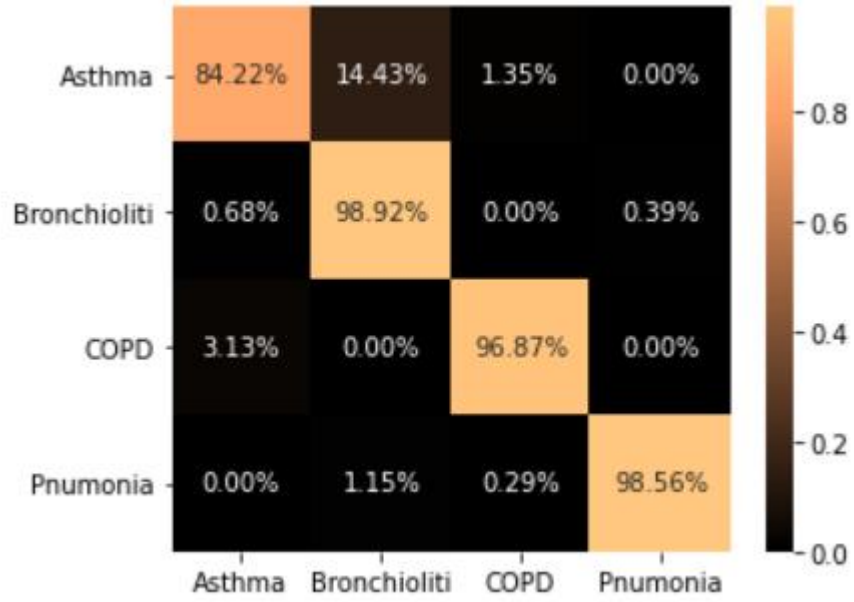


Figure 35 - Cross-Validation Result for K=10

From the above assessment on different K-values, the learning function exerts much better performance for all 4 classes and also elevated in the overall accuracy than the rest k-fold values. Seeing this potential, we have chosen to train our model using cross-validation of 5 folds.

6.2.2.2. Hyper Parameter Tuning Results

Using the Keras_Tunner hyperparameter tuning algorithm we have got a combination of best-performing hyperparameters. The algorithm chooses between different learning rate values, the number of neurons, and the number of epochs. The output from the algorithm was 160 neurons in the first layer, 32 neurons in the second layer, 16 neurons in the third hidden layer. From the chosen learning rates the algorithm best finds 0.001 as a learning rate and 100 number of epochs. The tuned hyperparameters give 97% accuracy; it holds an improvement of 1% on the model. The value of hyperparameters after and before tuning is presented below.

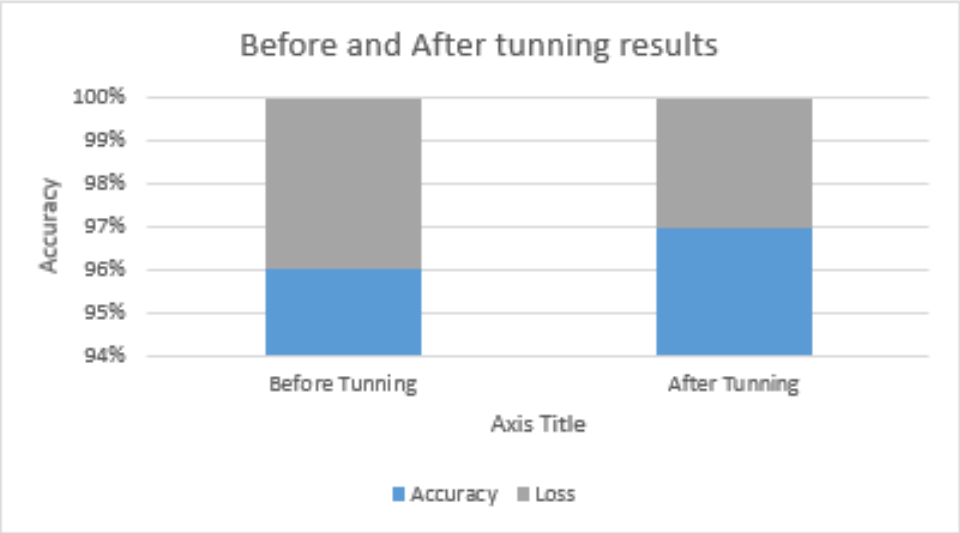


Figure 36-Hyper parameter Tuning Result

6.3. Overall Prediction Results Based on Expert Evaluation

From both respiratory sound and pulmonary pathosis classifiers, we have chosen the best performing model based on the obtained experiment result. However, to see the overall performance of the model we have gathered 10 new medical histories of patients along with their breathing sound that has been recorded with Boya microphone. The detailed medical history of each subject is provided in *Appendix E*. A volunteer Pulmonologist from St. Paul Hospital grants the diagnostic assessment for each subject. Simultaneously the new data was anticipated by the provided models. Here the result acquired from the expert pulmonologist and the proposed model is disclosed.

Table 15- Comparison for Expert and Proposed Model Prediction

Subject	***Expert Evaluation***	***Proposed Model's Prediction***	
Subject 1	Bronchiolitis	Bronchiolitis	1.00
Subject 2	Asthma	Bronchiolits	0.67
Subject 3	Asthma	Asthma	1.00
Subject 4	COPD	COPD	1.00
Subject 5	COPD	COPD	1.00
Subject 6	Bronchiolits	Asthma	0.33
Subject 7	Pneumonia	Pneumonia	1.00

Subject 8	Asthma	Asthma	1.00
Subject 9	COPD	COPD	1.00
Subject 10	Pnumonia	COPD	0.67
Average			0.867

In the table above the diagnosis from the expert and our proposed solution is indicated, the red cover indicates where our model fails to predict the true value. Despite the fact that the breathing sound is recorded with a low-quality microphone still the model is capable of predicting, surprisingly the model misses only 4 tests of total 30 testing sets from 10 subjects. This result is very promising therefore we can say that our model's overall performance is moderately accurate.

6.4. Discussion

An initial objective of this thesis work is to predict respiratory disease. The prediction starts with the classification of respiratory sound as wheeze, crackle, normal, or crackle and wheeze. To improve the respiratory sound classifier, we have used different feature extraction and augmentation techniques. As shown in *table 7* the CNN model gives a good performance result when the MFCC features are combined with Mel-spectrogram features. Whereas among the augmentation experiments VTLP and Time-shift give good accuracy than Spec-Aug augmentation. In *table 9* the combinations of augmentation and feature extractions is given. From the table, we can see that the best combination of features and augmentation techniques is when MFCC with Mel-spectrogram along with Time-shift and VTLP provides the best accuracy to the model. This was also improved by hyper-parameter tuning. In the case of the pulmonary pathosis classifier, the DNN model was evaluated through 3,5 and 10 K-values of cross-validation. And the result in *table 13* and *figure 30* shows that cross-validation with k value of 5 gives the best accuracy. Therefore, for the general respiratory diseases prediction, we have chosen a sound classifier with MFCC and Mel-spectrogram features along with Time-shift and VTLP, also for pulmonary pathosis DNN with 5-fold cross-validation and the hyperparameters tuned are the chosen techniques.

Eventually, 3 research questions are expected to be answered at the end of this research. Let us see how the research answered the indicated questions.

RQ-1: What Percentage of enhancement do Mel-spectrogram features give for respiratory sound prediction?

To answer the question, we have done experiments. Based on the observations from the experiment, using Mel-spectrogram as a feature extraction has a positive impact. This makes the model learn every important feature and hence giving a good accuracy. From our experiment of MFCC, it has 1.5% enhancement and from the closer research by Rena Liu et al, it has got a 8.98% rise. In addition to that when Mel-spectrogram is used in combination with MFCC features the performance is astonishing.

RQ-2: What is the effect of the Spec-Aug augmentation technique in respiratory sound prediction?

We have applied augmentation techniques to make the respiratory sound data balance. A relatively new augmentation technique Spec-Aug is said to have a good performance in augmentation. But in opposite to the desired outcome, Spec-Aug gives a poor result. However, the training time Spec-Aug takes shows speed. This is because the augmentation is done directly on the Mel-spectrogram itself. Whereas other augmentation techniques need further extraction after each augmentation and therefore they are slow. But considering its accuracy, the Spec-Aug augmentation technique is not desirable for difficult audio analysis tasks that have a relatively low pitch or frequency.

RQ-3: Can respiratory disease prediction be improved using patients' medical history in addition to their breathing sound?

This question has been answered by subdividing the prediction process into two. Models that work for patient's medical history as well as for respiratory sound classification. For this, a model that classifies the type of respiratory sound along with a pulmonary pathosis prediction makes the decision-making very robust. For instance, if the breathing sound of a patient is normal, our model won't classify the patient as healthy without referring to other medical records. This was one of the downsides of previous works in respiratory disease prediction. The other one is, for respiratory diseases that have the same adventitious sound type, the previous models get confused. However, the proposed solution shows a better result in such bewilder situations. Generally speaking, this research answers that respiratory disease prediction using a patient's medical history in addition to the breathing sound yields a promising outcome.

To the tip of our knowledge, no prior research has been done using both respiratory sound and patient's medical history for respiratory disease prediction. Therefore, we cannot fully compare with the previous works. However, we can compare the respiratory sound classifier result to the researches that have been done only to predict adventitious respiratory sound types. Comparison between the previous works on respiratory sound prediction to that of our proposed solution is elucidated as follows.

Table 16- Comparison of the proposed solution with previous works

	Chamber et al.	**N. Jakovljevic et al.**	**Renal Liu et al.**	**Proposed Solution**
Dataset	ICBHI	ICBHI	ICBHI + Paediatric	ICBHI
Model	SVM	HMM	CNN	CNN
Feature	MFCC	MFCC	Mel-filter bank	MFCC + Mel-Spectrogram
Augmentation	-	-	-	Time-shift + VTLP
Accuracy	49.98%	39.56%	61.02%	**72.0%**

As shown in the table above, all researchers including our research used the legend ICBHI dataset. Yet, the algorithm used and feature extraction methods are different. If we see the algorithm CNN model has outperformed the classical ML models because all of such works exsert accuracy below 60%. In feature extraction even if chamber et al and n.jakov et al uses MFCC features because of the data imbalance and the learning model they used their performance is low. In the case of Renyu Liu et al, CNN with Mel-filterbank was used but the performance was still low. From the table, we can conclude that augmentation techniques and the features employed give enhancement to our model.

CONCLUSION AND FUTURE WORK

Conclusion

Eventually, this study was conducted keeping in mind that; the work of related researchers in decreasing respiratory diagnosis uncertainty needs improvement. In this thesis respiratory disease prediction was held in two phases; first on respiratory sound then on pulmonary pathosis classification. Deep learning techniques such as CNN and DNN with Mel-spectrogram and MFCC features including time-stretch and VTLP augmentations were implemented. Even though we were limited with GPU resources; we believe that enough experiments were done to answer the research questions. Using this demonstration respiratory sound classification has shown enhancements, but the Spec-Aug augmentation technique does not give the expected result instead it shows us that this augmentation technique is not valuable for respiratory sound prediction, nevertheless the augmentation technique costs a little amount of computational time for execution, from this we can speculate that the augmentation would give a good result if the amount of data is huge. Generally, the overall respiratory disease prediction becomes less likely open to the overlap of respiratory sound anomalies and this has been evaluated with the help of an expert. Also, the work highly mimics the formal preliminary clinical diagnosis and at the same time, it shows an improvement.

Future Work

The respiratory sound makes respiratory disease prediction very difficult. Consequently, this research tries to fill some gaps, regardless of that still there are a lot of issues that have not been addressed by this research. Therefore, we have mentioned some issues to be used as future work. In feature extraction MFCC and Mel-Spectrogram have good performance however, they are not robust for noise in the sound. To address this problem, a relatively new noise-robust feature extraction GFCC (Gammaton Frequency Cepstral Coefficient) is advocated. Also in data augmentation a very powerful augmentation technique GAN is advised. Finally, to make the disease prediction more natural and suitable, using the patient medical history in the form of text or natural language is suggested.

* * *

This research work fund was granted by Adama Science and Technology University.
ASTU/SM-R/232/21

REFERENCE

Anbesse, Z. K., Mega, T. A., Tesfaye, B. T., & Negera, G. Z. (2020). Early readmission and its predictors among patients treated for acute exacerbations of chronic obstructive respiratory disease in Ethiopia: A prospective cohort study. *PLoS ONE*, *15*(10 October), 16–29. https://doi.org/10.1371/journal.pone.0239665

Badnjevic, A., Gurbeta, L., & Custovic, E. (2018). An Expert Diagnostic System to Automatically Identify Asthma and Chronic Obstructive Pulmonary Disease in Clinical Settings. *Scientific Reports*, *8*(1), 1–9. https://doi.org/10.1038/s41598-018-30116-2

Basu, V., & Rana, S. (2020). Respiratory diseases recognition through respiratory sound with the help of deep neural network. *4th International Conference on Computational Intelligence and Networks, CINE 2020*. https://doi.org/10.1109/CINE48825.2020.234388

Breebaart, J., Laboratories, D., Mckinney, M. F., & Technologies, S. H. (2014). *Features for Audio Classification*. *October*. https://doi.org/10.1007/978-94-017-0703-9

Broenlee, J. (n.d.). *How to Configure K-Fold Cross-Validation*. Retrieved August 8, 2021, from machinelearningmastery.com/how-to-configure-k-fold-cross-validation/

Caliskan, A., & Yuksel, M. E. (2017). Classification of coronary artery disease data sets by using a deep neural network. *The EuroBiotech Journal*, *1*(4), 271–277. https://doi.org/10.24190/issn2564-615x/2017/04.03

Carneiro, T., Medeiros, R. V, Nóbrega, D. A., Nepomuceno, T., Bian, G., & Albuquerque, V. H. C. D. E. (2018). *Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications*. 1–9. https://doi.org/10.1109/ACCESS.2018.2874767

Chambres, G., Hanna, P., & Desainte-Catherine, M. (2018). Automatic detection of patient with respiratory diseases using lung sound analysis. *Proceedings - International Workshop on Content-Based Multimedia Indexing*, *2018-Septe*, 0–5. https://doi.org/10.1109/CBMI.2018.8516489

Demir, F., Sengur, A., & Bajaj, V. (2020). Convolutional neural networks based efficient approach for classification of lung diseases. *Health Information Science and Systems*,

8(1), 1–8. https://doi.org/10.1007/s13755-019-0091-3

Demir, F., Turkoglu, M., Aslan, M., & Sengur, A. (2020). A new pyramidal concatenated CNN approach for environmental sound classification. *Applied Acoustics*, *170*, 107520. https://doi.org/10.1016/j.apacoust.2020.107520

Dileep, P., Das, D., & Bora, P. K. (2020). Dense layer dropout based CNN architecture for automatic modulation classification. *26th National Conference on Communications, NCC 2020*. https://doi.org/10.1109/NCC48643.2020.9055989

Donnelly, M. (2014). Personal photo. *Foundations and Trends in Signal Processing*, *i*, 1–180.

Edition, S. (2019). *The Global Impact of Respiratory Disease* (1st ed.).

Friedman, J. N., Rieder, M. J., Walton, J. M., & Society, C. P. (2014). *Bronchiolitis : Recommendations for diagnosis , monitoring and management of children one to 24 months of age*. *19*(9).

Gavriely, N., Nissan, M., Cugell, D. W., & Rubin, A. H. E. (1994). Respiratory health screening using pulmonary function tests and lung sound analysis. *European Respiratory Journal*, *7*(1), 35–42. https://doi.org/10.1183/09031936.94.07010035

Geddes, D. (2016). The history of respiratory disease management. *Medicine (United Kingdom)*, *44*(6), 393–397. https://doi.org/10.1016/j.mpmed.2016.03.006

Gibson, G. J., Loddenkemper, R., Lundba, B., & Sibille, Y. (2013). *Respiratory health and disease in Europe: the new European Lung White Book*. 559–563. https://doi.org/10.1183/09031936.00105513

Grewal, J. K., Krzywinski, M., & Altman, N. (2019). Markov models — hidden Markov models. *Nature Methods*, *16*(9), 795–796. https://doi.org/10.1038/s41592-019-0532-6

Health, E. (2015). *Chest and Lungs* (1st ed., pp. 8–35). Elsevier. http://www.us.elsevierhealth.com/product.jsp?isbn=9780323055703 CHAPTER 13 Chest and Lungs%0A31

Huzaifah, M. (2020). *Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks*. *1*(Environmental Sound

Classification), 1–5.

Hwang, Y., Cho, H., Yang, H., Won, D., Oh, I., & Lee, S. (2019). *Mel-spectrogram augmentation for sequence-to-sequence voice conversion*.

Iwana, B. K., & Uchida, S. (2020). *An Empirical Survey of Data Augmentation for Time Series Classification with Neural Networks*. 1–41. http://arxiv.org/abs/2007.15951

Jaitly, N., & Hinton, G. E. (2013). Vocal Tract Length Perturbation (VTLP) improves speech recognition. *Proceedings of the 30 Th International Conference on Machine Learning*, *90*, 42–51. http://www.iarpa.gov/

Janiesch, C., & Heinrich, K. (2021). *Machine learning and deep learning*.

Karthikeyan, K., & Mala, R. (2018). *Content Based Audio Classifier & Feature Extraction Using Ann Techniques*. *5*(04), 106–116. https://doi.org/10.26562/IJIRAE.2018.APAE10081

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *Proceedings of 2014 Science and Information Conference, SAI 2014*, *October*, 372–378. https://doi.org/10.1109/SAI.2014.6918213

Khan, A., Sohail, A., Zahoora, U., & Saeed, A. (2020). A survey of the recent architectures of deep convolutional neural networks. In *Artificial Intelligence Review* (Issue 0123456789). Springer Netherlands. https://doi.org/10.1007/s10462-020-09825-6

Khan, I., Farooqui, A. U., Ullah, R., & Emad, S. M. (2019). *Robust Feature Extraction Techniques in Speech Recognition : A Comparative Analysis*. *April*. https://www.researchgate.net/publication/332254069%0Ahttps://www.researchgate.net/profile/Rafi_Ullah13/publication/332254069_Robust_Feature_Extraction_Techniques_in_Speech_Recognition_A_Comparative_Analysis/links/5ca9b6d1a6fdcca26d045f83/Robust-Feature-Ex

Knight, E. C., Poo Hernandez, S., Bayne, E. M., Bulitko, V., & Tucker, B. V. (2020). Pre-processing spectrogram parameters improve the accuracy of bioacoustic classification using convolutional neural networks. *Bioacoustics*, *29*(3), 337–355. https://doi.org/10.1080/09524622.2019.1606734

Liu, Q., & Wu, Y. (2015). *Supervised Learning*. *April*. https://doi.org/10.1007/978-1-4419-1428-6

Liu, R., Cai, S., Zhang, K., & Hu, N. (2019). Detection of Adventitious Respiratory Sounds based on Convolutional Neural Network. *ICIIBMS 2019 - 4th International Conference on Intelligent Informatics and Biomedical Sciences*, 298–303. https://doi.org/10.1109/ICIIBMS46890.2019.8991459

Madhu, A. (2019). Data Augmentation Using Generative Adversarial Network for Environmental Sound Classification. *2019 27th European Signal Processing Conference (EUSIPCO)*, 1–5. https://doi.org/10.23919/EUSIPCO.2019.8902819

Maglaveras, N., & Paulo, I. C. (2017). *Nicos Maglaveras · Ioanna Chouvarda Volume 66 Precision Medicine Powered by pHealth and Connected Health* (Vol. 66, Issue November).

Marais, J. A. (2019). *Deep Learning for Tabular Data : An by*. *April*.

Mason, J. (2017). *STRUCTURE AND FUNCTION OF RESPIRATORY TRACT IN RELATION TO ANAESTHESIA*. *2*(Respiratory health), 15–47.

Mayorga, P., Druzgalski, C., Morelos, R. L., González, O. H., & Vidales, J. (2010). Acoustics based assessment of respiratory diseases using GMM classification. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, 6312–6316. https://doi.org/10.1109/IEMBS.2010.5628092

Mekov, E., Miravitlles, M., & Petkov, R. (2020). Artificial intelligence and machine learning in respiratory medicine. *Expert Review of Respiratory Medicine*, *14*(6), 559–564. https://doi.org/10.1080/17476348.2020.1743181

Messner, E., Member, S., Fediuk, M., Swatek, P., & Scheidl, S. (2018). *Crackle and Breathing Phase Detection in Lung Sounds with Deep Bidirectional Gated Recurrent Neural Networks*. 356–359.

Mortality, G. B. D., & Collaborators, D. (2013). Global , regional , and national age – sex specifi c all-cause and cause-specifi c mortality for 240 causes of death , 1990 – 2013 : a systematic analysis for the Global Burden of Disease Study 2013. *The Lancet*, 1990–

2013. https://doi.org/10.1016/S0140-6736(14)61682-2

Nanni, L., Maguolo, G., & Paci, M. (2020). Data augmentation approaches for improving animal audio classification. *Ecological Informatics*, *57*(February), 101084. https://doi.org/10.1016/j.ecoinf.2020.101084

Park, D. S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). *SpecAugment : A Simple Data Augmentation Method for Automatic Speech Recognition*. *1*(Speech Recognition).

Peate, I., & Studies, H. (2018). *The respiratory system*. *12*(4), 10–13.

Pham, L., Mcloughlin, I., Phan, H., Tran, M., Nguyen, T., & Palaniappan, R. (1864). *Respiratory Anomalies and Diseases*. 90–93.

Pham, L., Phan, H., Palaniappan, R., Mertins, A., & Mcloughlin, I. (2014). *anomalies and lung disease detection*. *1*(Lung health).

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal on Selected Topics in Signal Processing*, *13*(2), 206–219. https://doi.org/10.1109/JSTSP.2019.2908700

Reichert, S., Gass, R., Brandt, C., & Andrès, E. (2008). Analysis of Respiratory Sounds: State of the Art. *Clinical Medicine. Circulatory, Respiratory and Pulmonary Medicine*, *2*, CCRPM.S530. https://doi.org/10.4137/ccrpm.s530

Rocha, B., Filos, D., & Mendes, L. (2017). *Adventitious sounds*. *December*, 29–31. https://doi.org/10.1007/978-981-10-7419-6

Rocha, B. M., Filos, D., Mendes, L., Serbes, G., Ulukaya, S., Kahya, Y. P., Jakovljevic, N., Turukalo, T. L., Vogiatzis, I. M., Perantoni, E., Kaimakamis, E., Natsiavas, P., Oliveira, A., Jácome, C., Marques, A., Maglaveras, N., Pedro Paiva, R., Chouvarda, I., & De Carvalho, P. (2019). An open access database for the evaluation of respiratory sound classification algorithms. *Physiological Measurement*, *40*(3). https://doi.org/10.1088/1361-6579/ab03ea

Rodriguez, C. (2020). *COMPARATIVE ANALYSIS OF SUPERVISED MACHINE LEARNING*

ALGORITHMS FOR HEART DISEASE DETECTION. 233–247.

Salamon, J., & Bello, J. P. (2017). Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Processing Letters*, *24*(3), 279–283. https://doi.org/10.1109/LSP.2017.2657381

Series, C. (2020). *A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification*. https://doi.org/10.1088/1742-6596/1453/1/012085

Sharma, G., Umapathy, K., & Krishnan, S. (2020). Trends in audio signal feature extraction methods. *Applied Acoustics*, *158*, 107020. https://doi.org/10.1016/j.apacoust.2019.107020

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

SPHMMC. (2019). sphmmc.edu.et/about

Swaminathan, S., Qirko, K., Smith, T., Corcoran, E., Wysham, N. G., Bazaz, G., Kappel, G., & Gerber, A. N. (2017). A machine learning approach to triaging patients with chronic obstructive pulmonary disease. *PLoS ONE*, *12*(11), 1–21. https://doi.org/10.1371/journal.pone.0188532

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, *19*(1), 1–16. https://doi.org/10.1186/s12911-019-1004-8

Wehle, H. (2017). *ML – AI- COGNITIVE*. *August*.

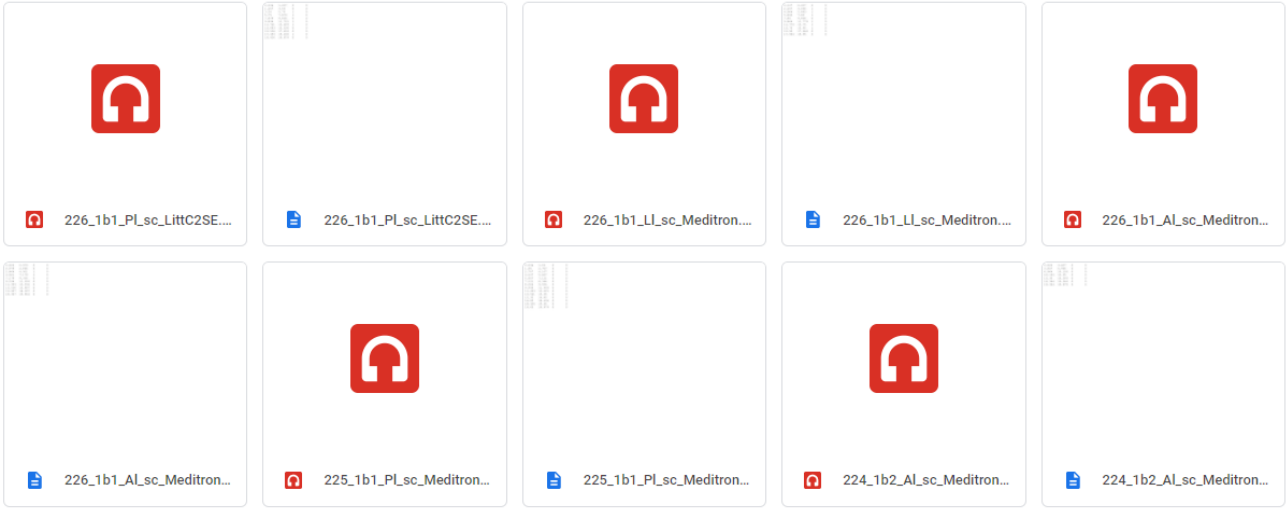
Yli-heikkilä, M. (2019). *Data Science Languages*. 2–3.

APPENDIXES

Appendix A: Sample Patient Symptom Dataset

P_Id	Age	Sex	RR	PR	Temp	Saturation	Chest	Cough	Fever	SOB	Grunting	HxAlergy	HxSmokin	AirPolluti	MalNutrit	Diagnosis
RP-1	2 yrs.	M	60	143	38.9	70	Crackle	Yes	High	Yes	Yes	No	No	Indoor	Yes	Pneumonia
RP-2	5 yrs.	M	48	120	36.1	89	Crackle	Yes	Low	Yes	No	No	No	No	No	Pneumonia
RP-3	11 months	M	80	138	38.1	74	Crackle	Yes	High	Yes	Yes	No	No	No	No	Pneumonia
RP-4	1 yr.	M	68	167	36.8	75	Wheez	Yes	Low	Yes	Yes	No	No	No	No	Asthma
RP-5	11 months	F	60	140	37.8	70	Crackle	Yes	Low	Yes	No	No	No	Indoor	No	Pneumonia
RP-6	6 yrs.	M	24	115	36	90	Wheez	Yes	Low	Yes	No	No	No	No	No	Asthma
RP-7	6 yrs.	M	33	95	36	92	Wheez	Yes	Low	Yes	No	No	No	No	No	Asthma
RP-8	6 months	M	64	152	36.4	93	Crackle	Yes	Low	Yes	No	No	No	No	No	Pneumonia
RP-9	3 yrs.	F	32	120	36.5	90	Wheez	Yes	Low	Yes	No	No	No	No	Yes	Asthma
RP-10	3 yrs.	M	26	92	37.1	92	Wheez	Yes	Low	Yes	No	Yes	No	No	No	Asthma
RP-11	14 yrs.	M	24	108	37	83	Normal	Yes	Low	Yes	Yes	Yes	No	Outdoor	No	Pneumonia
RP-12	23 yrs.	M	24	97	37.2	85	Wheez	Yes	Low	Yes	No	No	No	No	No	Asthma
RP-13	54 yrs.	M	28	100	36.1	86	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-14	52 yrs.	M	27	98	37.3	81	Wheez	Yes	Low	Yes	No	No	No	No	No	COPD
RP-15	58 yrs.	M	32	128	36.8	79	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-16	62 yrs.	M	28	112	37.4	79	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-17	1 yr.	F	35	132	37.5	84	Wheez	Yes	Low	Yes	No	No	No	No	Yes	Broncholitis
RP-18	2 yrs.	M	33	117	37	85	Wheez	Yes	Low	Yes	No	Yes	No	No	No	Broncholitis
RP-19	11 months	F	38	144	37.1	84	Wheez	Yes	Low	Yes	Yes	No	No	Indoor	No	Broncholitis
RP-20	24 yrs.	M	25	101	37.5	88	Wheez	Yes	Low	Yes	No	Yes	No	No	No	Asthma
RP-21	80 yrs.	F	40	56	36.3	57	Both	Yes	Low	Yes	Yes	No	No	Indoor	No	COPD
RP-22	65 yrs.	F	41	118	36.9	81	Both	Yes	Low	Yes	No	No	No	Indoor	No	COPD
RP-23	1 yr.	F	41	106	36.9	86	Wheez	Yes	Low	Yes	No	No	No	No	Yes	Broncholitis
RP-24	58 yrs.	M	27	99	37.5	83	Wheez	Yes	Low	Yes	No	No	Yes	No	No	COPD
RP-25	40 yrs.	M	31	110	36	80	Both	Yes	Low	Yes	No	No	Yes	No	No	COPD

Appendix B: Sample ICBHI Dataset



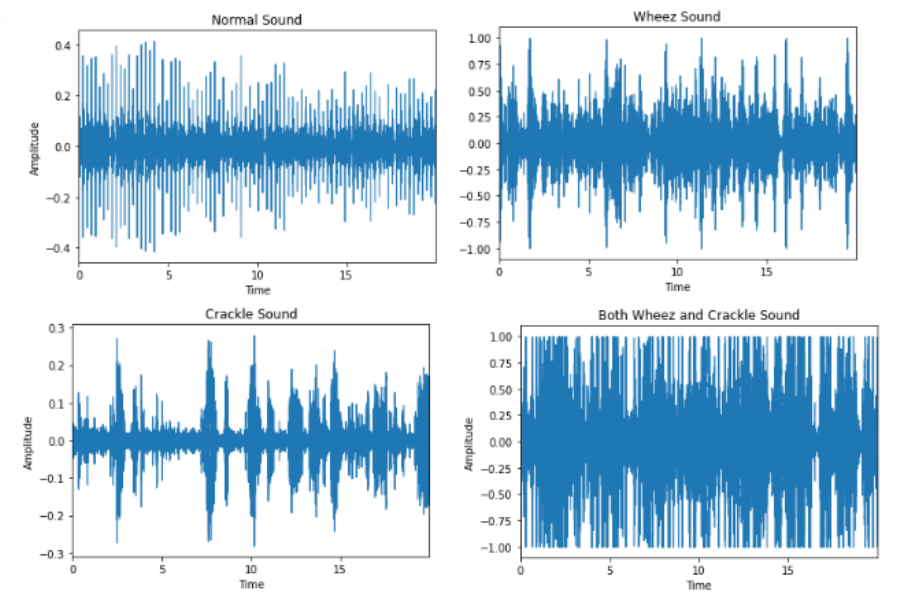
Appendix C: Data set Description

Parameter	***Symbol Used***	***Description***
Age	Age	The age of the patient
Sex	Sex	The gender of the patient
Temperature	Temp	Body temperature of a patient.
Respiratory Rate	RR	The number of breaths he/she take in a minute.
Pulse Rate	PR	The number of times the heart beats in a minute.
Saturation	So_2p	Measures the amount of oxygen to hemoglobin in the blood.
Chest	Chest	Tells the Lung sound of the patient
Cough	Cough	An act of clearing the throat by rapidly explosive air from the lung.
Fever	Fever	The patient's body temperature exceeding the normal temperature.
Shortness of Breath	SOB	Trouble in breathing is also referred as dyspnea.
Grunting	Gr	In labored breathing occurs due to low chest pressure.
History of Allergy	HxAllergy	History of abnormal reaction of the body.
History of Smoking	HxSmoking	The past story of a patient-related to tobacco smoking.
Air pollution	AirP	Is the pollution in the air due to many pollutants.

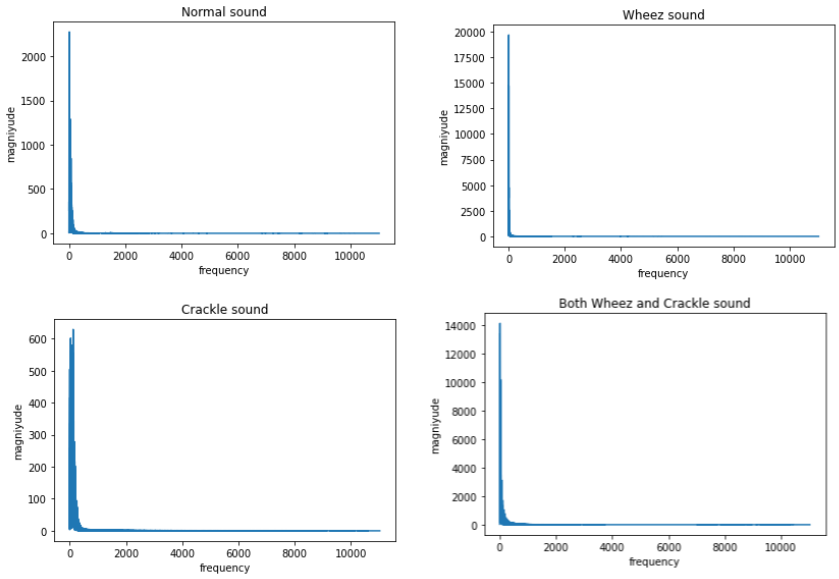
Mal Nutrition	MN	Imbalance or deficiencies in the most important nutrients
Diagnosis	Diagnosis	It is the identified or observed disease or condition.

Appendix D: Different Representations of Audio Signal

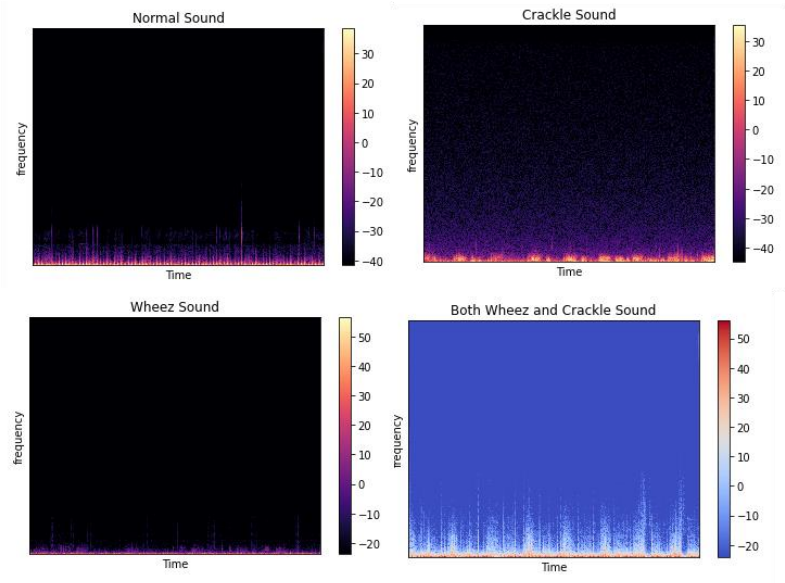
Waveform



Power Spectrum



Log Spectrum



Appendix E: Test Subjects for Evaluation

Subject 1: 7 months old female, 68 respiratory rate, 166 pulse rate, 37.1 temperature, saturation:86, low fever, have a history of allergy, no smoking history, no air pollution, no malnutrition, have grunting, and abnormal chest.

Subject 2: 2 years old male, 30 respiratory rate, 110 pulse rate, 36.7 temperature, saturation:89, low fever, have a history of allergy, no smoking history, no air pollution, have malnutrition, no grunting and abnormal chest.

Subject 3: 49 years old female, 37 respiratory rate, 119 pulse rate, 36.8 temperature, 77 saturations, low fever, have history of allergy, no smoking history, no air pollution, no mal nutrition, no grunting, abnormal chest.

Subject 4: 55 years old female, 53 respiratory rate, 129 pulse rate, 37.4 temperature, 73 saturations, low fever, have history of allergy, no smoking history, indoor air pollution, no mal nutrition, no grunting, abnormal chest.

Subject 5: 28 years old male, 29 respiratory rate, 101 pulse rate, 36.4 temperature, saturation:87, low fever, no history of allergy, have smoking history, no air pollution, no mal nutrition, no grunting, abnormal chest.

Subject 6: 7 years old male, 35 respiratory rate, 120 pulse rate, 36.9 temperature, saturation:84, low fever, have a history of allergy, no smoking history, no air pollution, no malnutrition, no grunting and abnormal chest.

Subject 7: 10 years old male, 34 respiratory rate, 118 pulse rate, 38.3 temperature, saturation:82, high fever, no history of allergy, no smoking history, indoor air pollution, no malnutrition, no grunting, normal chest.

Subject 8: 4 years old male, 43 respiratory rate, 81 pulse rate, 38 temperature, saturation:68, high fever, no history of allergy, no smoking history, no air pollution, have malnutrition, no grunting, abnormal chest.

Subject 9: 45 years old female, 35 respiratory rate, 103 pulse rate, 37.3 temperature, saturation:88, low fever, no history of allergy, no smoking history, indoor air pollution, no malnutrition, no grunting, normal chest.

Subject 10: 72 years old male, 45 respiratory rate, 101 pulse rate, 38.7 temperature, saturation:82, high fever, no history of allergy, no smoking history, no air pollution, no malnutrition, no grunting, abnormal chest.

Appendix F: Sample Prototype Code

```
private MappedByteBuffer loadmodelfile() throws IOException {
    AssetFileDescriptor fileDescriptor = this.getAssets().openFd( fileName: "rd_tf_lite.tflite");
    FileInputStream fileInputStream = new FileInputStream(fileDescriptor.getFileDescriptor());
    FileChannel fileChannel = fileInputStream.getChannel();
    long stratoffsets = fileDescriptor.getStartOffset();
    long declaredlength = fileDescriptor.getDeclaredLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY,stratoffsets,declaredlength);

}
}
```

Appendix G: Prototype of the Respiratory Disease Classifier