**A Project Report on**

# Recognition of Pulmonary Diseases based on Lung Sounds
# Department of CSE

submitted in partial fulfillment for the award of

**Bachelor of Technology**

in

## Computer Science and Engineering

by


**V.Pujitha (Y20ACS582)** **R.Sireesha (Y20ACS541)**

**S.S.Manikanta (L21ACS420)** **Sk.Baji(Y19ACS560)**


Under the guidance of
**Asst Prof. Dr.R. Venkata Lakshmi**


Department of Computer Science and Engineering
**Bapatla Engineering College**
(Autonomous)
(Affiliated to Acharya Nagarjuna University)
**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

# Department of
# Computer Science and Engineering

## CERTIFICATE

This is to certify that the project report entitled **Recognition of Pulmonary Diseases based on Lung Sounds using Deep Learning** that is being submitted by V.Pujitha (Y20ACS582), R.Sireesha(Y20ACS541), S.S.Manikanta(L21ACS420) and Sk. Baji(Y19ACS560) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

| | |
|---|---|
| **Signature of the Guide** | **Signature of the HOD** |
| **Dr.R.Venkata Lakshmi** | **Dr.M.Rajesh Babu** |
| **Assistant Professor** | **Associate Professor** |

# DECLARATION

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.


**V. Pujitha (Y20ACS582)**
**R. Sireesha (Y20ACS541)**
**S.S. Manikanta (L21ACS420)**
**Sk. Baji (Y19ACS560)**

# Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Dr. R. Venkata Lakshmi,** Assistant Professor ,Department of CSE, for her valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Mr. M. Rajesh Babu**, Associate Professor & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator

**Dr. N. Sudhakar,** Professor Dept. of CSE for her helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

**V. Pujitha (Y20ACS582)**
**R. Sireesha(Y20ACS541)**
**S.S. Manikanta (L21ACS420)**
**Sk.Baji(Y19ACS560)**

# Table of Contents

# List of Figures

# List of Tables

# Abstract

In our project, a study is conducted to explore the ability of deep learning in recognizing pulmonary diseases from electronically recorded lung sounds. We have methods in ML like SVM, k-NN in recognizing pulmonary diseases but they had the least accuracy. So, We have constructed a deep neural network model that takes in respiratory sound as input and classifies the condition of its respiratory system. The training of the model was evaluated based on several performance evaluation metrics including Cohen's kappa, accuracy, sensitivity, specificity, precision, and F1-score.

The developed algorithm achieved the highest average accuracy with precision in classifying patients based on the pulmonary disease types using CNN + GRU. The selected data-set included a total of 103 patients obtained from locally recorded stethoscope lung sounds acquired at Abdullah University Hospital, Jordan University of Science and Technology, Jordan. Prediction of respiratory diseases such as COPD(Chronic obstructive pulmonary disease), URTI(upper respiratory tract infection), Bronchiectasis, Pneumonia, Bronchiolitis with the help of deep neural networks or deep learning. This study paves the way towards implementing deep learning models in clinical settings to assist clinicians in decision making related to the recognition of pulmonary diseases.

# 1 Introduction

Pulmonary diseases present a significant global health challenge, affecting millions of lives annually. Timely and accurate diagnosis is crucial for effective treatment and management. Traditional diagnostic methods often require expensive equipment and expertise, leading to limited accessibility, particularly in resource-constrained settings. However, advancements in machine learning offer promising avenues for improving diagnostic accuracy and accessibility.

This project aims to leverage deep learning techniques, specifically Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU), to develop a robust system for the recognition of pulmonary diseases based on lung sounds. Lung sounds, also known as respiratory sounds, provide valuable information about the health of the respiratory system. Abnormalities in these sounds can indicate various pulmonary conditions, including pneumonia, asthma, chronic obstructive pulmonary disease (COPD), and bronchitis.

## 1.1 Overview

The classification and identification of breathing diseases is a tedious task. The sound that is produced when a person breathes is directly associated with the movement of air, variations in the lung tissue and the position of the secretions inside the lung. A wheezing sound is an example for a person with obstructive disease like asthma or chronic obstructive pulmonary disease. One of the major causes of mortality and morbidity worldwide is respiratory diseases. It developed the third prominent cause of death in 2020. Each year greater than 3 million of people go down from chronic obstructive pulmonary diseases which is roughly 6% of all fatalities in the world.

Around 251 million cases of chronic obstructive pulmonary disease were reported globally in 2016. By 2030 COPD will be one of the leading causes of death worldwide. Asthma is also related to COPD, but the definition is different. This disease also results in social and economic burden that is both substantial and increasing. The important treatment outcomes of COPD are symptoms, acute exacerbations and limitations of airflow .Interestingly, the Sounds from the lungs conveys significant information associated with respiratory diseases and it helps to assess the patients with pulmonary or respiratory disorders.

Sounds released from a person's breath are directly related to changes in lung tissue, position of secretion within the lungs and air movement. For instance, wheezing sound is a common indication that the patients have diseases like obstructive airway disease (asthma and chronic obstructive pulmonary disease). The healthcare professionals use traditional auscultation methods to detect the disorders of lungs, but this method has many limitations for instance there are chances of misdiagnosis if the physician is not trained well. In addition, lungs are non-stationary, this leads to difficulty in recognition, analysis and distinction. Pulmonary disease affects social, economic and health of people's lives significantly. Because of these reasons lots of research are going on for early diagnosis and intervention of respiratory disease.

In this perspective, characteristics of lung sounds provide valuable indications for the diagnosis and detection of respiratory abnormalities and infections. However, there are many limitations in the application of stethoscope in research studies because of the variability in inter-observer and subjectivity in lung sounds interpretation. Diagnosis of the diseases from lung sounds needs professional training and experts. This is undoubtedly costly and inconvenient. In this context a technique that can automatically and accurately classify the sounds of lungs into many groups is very meaningful. It helps to detect potential threats at a very early stage. So, the most important purpose of the research is to develop an automated system to predict and diagnose respiratory diseases using lung sounds.

The most important objective of the research is to detect and categorize the lung noise digital signal with the help of signal learning processing methods. More specifically the study will be comparing the performance of convolutional neural network architecture and Deep learning algorithms and to create a model based on the best performing algorithm. Ideally, this technique will improve detection of sounds and categorization of accuracy and robustness when encountered with different modes of sound and additional components while gaining the lung vibration wave.

This research will be very helpful for the healthcare professionals such as doctors for the easy and accurate diagnosis of respiratory diseases. This study will be a major contribution in the area of the respiratory disease classification by using lungs sounds. It serves as a stepping- stone for future research in classification of lung sounds using convolution neural networks. In addition, it helps the policy makers and researchers to make and amend the decisions in lung diseases. We use the python as a background for our development of the system as it gives more functionality for data analysis.

## 1.2 Types of Diseases

Various pulmonary conditions manifesting through distinctive lung sounds can be categorized for recognition using advanced neural network models. These diseases encompass a wide spectrum, including but not limited to asthma, pneumonia, bronchitis, emphysema, and pulmonary fibrosis. Each condition presents unique acoustic signatures within lung sounds, ranging from wheezing and crackles to diminished breath sounds or prolonged expiratory phases. Asthma, for instance, is characterized by wheezing due to airway constriction, while pneumonia often exhibits crackles resulting from fluid-filled alveoli. Bronchitis may feature both wheezes and crackles due to inflammation in the bronchial tubes, while emphysema showcases diminished breath sounds due to damaged alveolar walls.

### 1.2.1 Asthama

Asthma is a condition in which your airways narrow and swell and may produce extra mucus. This can make breathing difficult and trigger coughing, a whistling sound (wheezing) when you breathe out and shortness of breath. For some people, asthma is a minor nuisance. For others, it can be a major problem that interferes with daily activities and may lead to a life-threatening asthma attack. Asthma can't be cured, but its symptoms can be controlled. Because asthma often changes over time, it's important that you work with your doctor to track your signs and symptoms and adjust your treatment as needed.

### 1.2.2 Bronchitis

The most common cause of chronic bronchitis is cigarette smoking**.** Air pollution and dust or toxic gases in the environment or workplace also can contribute to the condition. Chronic bronchitis results from Trusted Source repeated irritation and damage to the lung and airway tissues. The most common cause is smoking, but not everyone with bronchitis is a smoker. Bronchitis is an inflammation of the lining of your bronchial tubes, which carry air to and from your lungs. People who have bronchitis often cough up thickened mucus, which can be discolored. Bronchitis may be either acute or chronic.

### 1.2.3 Pneumonia

Pneumonia is a lung infection that can range from mild to so severe that you have to go to the hospital. It happens when an infection causes the air sacs in your lungs (your doctor will call them alveoli) to fill with fluid or pus. That can make it hard for you to breathe in enough oxygen to reach your bloodstream. Lifestyle habits, like smoking cigarettes and drinking too much alcohol, can also raise your chances of getting pneumonia.

### 1.2.4 Chronic Obstructive Pulmonary Disease

Chronic obstructive pulmonary disease (COPD) is a type of obstructive lung disease characterized by long-term breathing problems and poor airflow. The main symptoms include shortness of breath and cough with sputum production. COPD is a progressive disease, meaning it typically worsens over time. Eventually, everyday activities such as walking or dressing become difficult. Chronic bronchitis and emphysema are older terms used for different types of COPD. The term "chronic bronchitis" is still used to define a productive cough that is present for at least three months each year for two years. Those with such a cough are at a greater risk of developing COPD.

### 1.2.5 Lower Respiratory Tract Infection

Lower respiratory tract infection (LRTI) is a broad terminology which includes acute bronchitis, pneumonia, acute exacerbations of chronic obstructive pulmonary disease/chronic bronchitis (AECB), and acute exacerbation of bronchiectasis. Infections are caused by tiny organisms known as bacteria or viruses, which are usually the most common cause.

They are carried in tiny droplets and passed between people by coughing, sneezing and at times by indirect contact with surfaces. People who have lower respiratory tract infections will experience coughing as the primary symptom. People with upper respiratory tract infections will feel the symptoms mainly above the neck, such as sneezing, headaches, and sore throats.

## 1.3 Stethoscope

An acoustic device used to detect the breath sounds of a patient is the stethoscope, a diaphragm in stethoscope can detect normal breath sounds and abnormal breath sounds without increasing lower frequency masking sounds. A hollow tube fitted to a chest piece including a wider diaphragm is in the currently available acoustic stethoscope. The high frequency sounds are transmitted through diaphragm and the low wavelength sounds are transmitted through smaller hollow bell. This kind of stethoscope will convey sound proportional to frequency made from the heart sounds. A digital stethoscope tries to better on some of the functionalities of the traditional acoustic stethoscope.

The sounds are converted into digital analogue by using piezoelectric sensor or a microphone which capture and transform the sound to electrical. The acquired signal is then passed through a band pass filter to amplify and process the signal and to remove noise that is corrupting the signal. Another round of denoising is finished with a computerized channel, to extract the signals of interest from the frequency band. Some propelled curio evacuation procedures are added here. The heart sounds acquired in this process are standardized to a specific scale and are fragmented into cycles. This causes us in identifying the sound segments obviously.

## 1.4 Respiratory Sounds

Identification of normal and abnormal respiratory sounds such as crackles, wheezes is very essential for accurate diagnosis of the diseases. These sounds include a lots of information about the pathologies and physiologies of lung structure and any obstruction in airways can be identified from the sounds. Around the beginning of the 19th century, doctors diagnosed their patients by keeping their ears to the thorax and chest to hear the noises with in and this method was called immediate auscultation.

Various studies were done, and research was made to test human ears' capacity to identify crackles. The research consisted of crackles simulated to superimpose as real respiratory/breathing sound.The most important detection errors were identified from these research like intensity of crackles, different types of crackles, different wavelengths and so on. From these studies we could conclude that traditional auscultation should not be considered as individual reference for validating respiratory sounds. The following two sounds explore the difference between lung and breath sounds.

Lung Sounds: Air flow of the chest causes the respiratory sounds. Pulmonary deficiencies result in the changes of the lung sounds.

Breath Sounds: Airflow through the trachea-bronchial tree. According to Hadjileontiadis and Moussavi (2018), breath sounds are crackling, plashing, wheezing and bubbling sounds coming from the chest.

Respiratory sounds are classed as normal and adventitious or abnormal sounds:

### 1.4.1 Normal respiratory sounds

It's essential to highlight the baseline sounds produced by a healthy respiratory system, serving as a crucial reference point for distinguishing abnormalities indicative of pulmonary diseases. Normal lung sounds consist of two primary types: bronchial and vesicular sounds. Bronchial sounds are heard over the trachea and large airways and are characterized by a high-pitched, hollow quality during both inspiration and expiration. Vesicular sounds, on the other hand, are softer and heard over the peripheral lung fields, with a rustling quality predominantly during inspiration.

These sounds are produced by the turbulent flow of air through the airways and alveoli during the respiratory cycle. Normal respiratory sounds are typically clear, without any adventitious or abnormal noises such as crackles, wheezes, or diminished breath sounds. Understanding and recognizing these normal sounds are essential for healthcare professionals when assessing lung health and identifying deviations that may signify underlying pulmonary conditions.

### 1.4.2 Vesicular sounds

It is the normal noise of the breath heard over the chest wall and concurrent with the airways air flow. It is originated from alveoli. These sounds can hear only in the starting of expiration and these are easy to hear during inspiration. Zimmerman and Williams (2020) acknowledge that vesicular sounds are low pitched, soft, mostly inspiratory and appreciated well at the posterior lung bases. It's crucial to delve into the specific characteristics of this type of respiratory sound, which plays a significant role in assessing lung health. Vesicular sounds are the soft, rustling noises heard over the peripheral lung fields during normal breathing.

These sounds originate from the passage of air through the smaller airways and alveoli during inspiration. Typically, vesicular sounds are described as low-pitched and continuous throughout inspiration, with a slight decrease in intensity during expiration. They are often likened to the sound of gentle wind blowing through leaves. The absence of vesicular sounds or alterations in their quality, such as increased loudness or harshness, may indicate abnormalities such as consolidation, airway obstruction, or fluid accumulation within the lungs. Healthcare professionals rely on their expertise in auscultation to interpret vesicular sounds accurately, aiding in the diagnosis and management of various pulmonary conditions.

### 1.4.3 Wheezes

These are melodic sounds that are generated by movement of air through bronchioles or constricted small airways. At the end of inspiratory phase or early expiratory phase wheeze are typically heard. These can be regarded as a marker for detection of obstructive disease such as COPD and asthma also patients having sickle cell experience an acute pain crisis.it is produced when gradual closure during expiration and reopening at the time of inspiration. 100-250 ms is the duration of wheeze and these are characterized as sinusoidal oscillations with harmonic distortion. Between 2- 7 tracheobronchial tree wheeze is more likely occurs.

### 1.4.4 Crackle

The nature of crackle sound is sudden bursts and explosive. Less than 20 ms is the duration of crackle sounds and has 200-2000 Hz spectrum of frequency range. These are produced during the closing in the expiration or during the opening of abnormally closed airway at the time of inspiration. Each of the immediate closing or opening of an airway represents single crackle. Forgac's theory states that a gas pressure is developed across the airway during inspiration which is then collapse during expiration.

Based on the duration, scheduling in the respiratory cycles, pitch, loudness and relation to altering the status of body and coughing crackles are classed as fine and coarse crackles. Inside the small airways fine crackles are generated and having shorter duration 5 ms, while medium crackle are supplied by bubbling of air via secretion in small bronchi. Coarse crackles are delivered from bronchiectasis segments or from large bronchi and it has longer duration 15 ms. The combination of fine and coarse crackle is called biphasic crackle. With the help of time expanded waveform analysis crackles can be differentiated.

### 1.4.5 Rhonchi

Rhonchi sounds are low-pitched sounds having rapidly damping periodic waveform. It has more than 100 ms duration and less than 300 Hz frequency. These sounds are related to creation of breach of fluid film and airway collapsibility. These sounds can be clearly head during coughing and clinical practice evident that secretions in large airways contribute a significant role in producing Rhonchi sounds. These sounds can be measured as a marker of constriction of airway lumen and mucosal thickening, oedema or bronchospasm. Zimmerman and Williams (2020) highlight that Rhonchi sounds are usually arise during exhalation or both inhalation and exhalation time, but it do not arise in inhalation alone.

### 1.4.6 Abnormal sounds

Abnormal heart sounds are heard in addition to the normal respiratory sounds .It is also called adventitious sounds that is, sound super imposed on normal breath sounds. These sounds can categorized on the basis of underlying condition and therefore it is very useful in aiding diagnosis. Based on the duration of sounds abnormal sounds can be grouped into continuous and discontinuous sounds. Crackle, Rhonchi and wheeze are the main abnormal sounds that are usually heard. Rhonchi and wheeze comes under the continuous sounds because they are inseparable sounds and are not interrupted while crackles are discontinuous sounds as it is calculated by assessor as decreased acoustic event it's like dropping a marble on a floor.

### 1.4.7 Bronchial sounds

Bronchial sounds are typically heard over the trachea and larger airways and are characterized by their high-pitched, hollow quality. These sounds are more prominent during expiration than inspiration and are often described as resembling the sound of air passing through a hollow tube. Bronchial sounds are produced by the turbulent flow of air through the relatively narrow and rigid structures of the trachea and larger bronchi. In healthy individuals, bronchial sounds are clear and well-defined, with a short pause between inspiration and expiration. However, alterations in bronchial sounds, such as prolonged expiration or the presence of wheezing, may indicate conditions such as bronchitis, asthma, or bronchial constriction. Healthcare professionals utilize their expertise in auscultation to differentiate normal bronchial sounds from abnormal ones, aiding in the diagnosis and management of respiratory disorders.

### 1.4.8 Broncho vesicular sounds

Bronchovesicular sounds exhibit characteristics that fall between bronchial and vesicular sounds and are typically heard over the central areas of the lungs, such as around the upper ternum and between the scapulae. These sounds possess a blend of qualities from both bronchial and vesicular sounds, featuring a moderate pitch and intensity, with equal duration during inspiration and expiration. Bronchovesicular sounds are produced by air passing through the larger bronchi and bronchioles, as well as the adjacent alveoli. In healthy individuals, bronchovesicular sounds are typically symmetrical and evenly distributed across both lung fields.

However, alterations in these sounds, such as asymmetry or changes in intensity, may indicate underlying respiratory conditions, including pneumonia, lung consolidation, or bronchial obstruction. Healthcare professionals rely on their expertise in auscultation to interpret bronchovesicular sounds accurately, aiding in the diagnosis and management of pulmonary disorders. During inspiration and expiration time this sound can be hear and it have a mid-range intensity and pitch. Commonly these sounds are heard over the upper 3rd of the anterior chest.

# 2 Literature Survey

The stethoscope and the semantic of auscultatory findings were invented more than 200 years ago by Dr. Laennec and over the years very few changes have been made to both the stethoscope itself and the way in which it is used. . However, the ability to differentiate between normal and abnormal sounds or noises (vesicular sounds, wheezes, crackles, etc.) remains essential in clinical practice for correct diagnosis and management. Over the past two decades, much of the progress made in this area has resulted primarily from improvements made to the stethoscope itself.[1]

More recently, we have seen advances in the techniques used to process auscultatory signals as well as in the analysis and clarification of the resulting sounds. The availability of novel representations of the sounds, with phono- and spectrograms, also opens interesting perspectives in the context of diagnostic aids, but also in education and pedagogy. It aims to review recent technological advances, evaluate promising innovations and perspectives in the field of auscultation, with a special focus on the development of new intelligent communicating stethoscope systems in clinical practice, and in the context of teaching and telemedicine.

The automatic analysis of respiratory sounds has been a field of great research interest during the last decades. Automated classification of respiratory sounds has the potential to detect abnormalities in the early stages of respiratory dysfunction and thus enhance the effectiveness of decision making. However, the existence of a publically available large database, in which new algorithms can be implemented, evaluated, and compared, is still lacking and is vital for further developments in the field. In the context of the International Conference on Biomedical and Health Informatics (ICBHI), the first scientific challenge was organized with the main goal of developing algorithms able to characterize respiratory sound recordings derived from clinical and non-clinical environments.

The database was created by two research teams in Portugal and in Greece, and it includes 920 recordings acquired from 126 subjects. A total of 6898 respiration cycles were recorded. The cycles were annotated by respiratory experts as including crackles, wheezes, a combination of them, or no adventitious respiratory sounds. The recordings were collected using heterogeneous equipment and their duration ranged from 10 to 90 s. The chest locations from which the recordings were acquired was also provided. Noise levels in some respiration cycles were high, which simulated real life conditions and made the classification process more challenging.[2]

Background Automatic detection or classification of adventitious sounds is useful to assist physicians in diagnosing or monitoring diseases such as asthma, Chronic Obstructive Pulmonary Disease (COPD), and pneumonia. While computerized respiratory sound analysis, specifically for the detection or classification of adventitious sounds, has recently been the focus of an increasing number of studies, a standardised approach and comparison has not beenwell established. Objective To provide a review of existing algorithms for the detection or of existing algorithms for the detection or classification of adventitious respiratory sounds. This systematic review provides a complete summary of methods used in the literature to give a baseline for future works.

Search terms included adventitious sound detection, adventitious sound classification, abnormal respiratory sound detection, abnormal respiratory sound classification, wheeze detection, wheeze classification, crackle detection, crackle classification, rhonchi detection, rhonchi classification, stridor detection, stridor classification, pleural rub detection, pleural rub classification, squawk detection, and squawk classification. Detection or classification methods used varied from empirically determined thresholds to more complex Deep learning techniques.[3]

Supervised Deep learning is the construction of algorithms that are able to produce general patterns and hypotheses by using externally supplied instances to predict the fate of future instances. Supervised Deep learning classification algorithms aim at categorizing data from prior information. Classification is carried out very frequently in data science problems. Various successful techniques have been proposed to solve such problems viz. Rule-based techniques, Logic-based techniques, Instance-based techniques, stochastic techniques. This paper discusses the efficacy of supervised Deep learning algorithms in terms of the accuracy, speed of learning, complexity and risk of overfitting measures. The main objective of this paper is to provide a general comparison with state of art Deep learning algorithms.[4]

COPD has been perceived as being a disease of older men. However, >7 million women are estimated to live with COPD in the USA alone. Despite a growing body of literature suggesting an increasing burden of COPD in women, the evidence is limited. To assess and synthesize the available evidence among population-based epidemiologic studies and calculate the global prevalence of COPD in men and women. A systematic review and meta-analysis reporting gender-specific prevalence of COPD was undertaken. Gender-specific prevalence estimates were abstracted from relevant studies.

Associated patient characteristics as well as custom variables pertaining to the diagnostic method and other important epidemiologic covariates were also collected. A Bayesian random-effects meta-analysis was performed investigating gender-specific prevalence of COPD stratified by age, geography, calendar time, study setting, diagnostic method, and disease severity. Among 194 eligible studies, summary prevalence was 9.23% (95% credible interval [CrI]: 8.16%–10.36%) in men and 6.16% (95% CrI: 5.41%–6.95%) in women.We conducted the largest ever systematic review and meta-analysis of global prevalence of COPD and the first large gender-specific review. These results will increase awareness of COPD as a critical woman's health issue.[5]

# 3 System Analysis

In the system analysis phase for pulmonary disease recognition using lung sounds with CNN and GRU, a comprehensive approach is essential to ensure the project's success. This involves defining the project's scope, identifying stakeholders, gathering requirements, and understanding technical and operational aspects. Key steps include delineating objectives and deliverables, identifying stakeholders, gathering functional and non-functional requirements, assessing data availability and quality, designing the system architecture, selecting appropriate algorithms, training and evaluating the model, integrating and testing components, and finally deploying and maintaining the system. By meticulously following these steps, the system can effectively leverage CNNs for spatial feature extraction from spectrograms and GRUs for capturing temporal dependencies in sequential data. This approach aims to assist healthcare professionals in diagnosing and managing pulmonary diseases based on lung sounds, ultimately improving patient outcomes and facilitating timely interventions.

## 3.1 Existing System

The Existing system represents a novel approach in the realm of deep learning architectures for signal processing, particularly in the domain of disease recognition using lung sound signals. Unlike traditional methods that necessitate separate spatial and temporal feature extraction steps, this system integrates both spatial and temporal processing within a single network architecture. This integration is achieved by combining Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BDLSTM) models.

A deep learning model based on convolutional neural networks (CNNs) and bidirectional long short-term memory (LSTM) was utilized for the purpose of lung sounds classifcation. The classifcation of lung sounds into multiple respiratory diseases using this model had an overall average accuracy of 89 % with a Cohen's kappa value of 86.26%. This study paves the way towards implementing deep learning trained models in clinical settings to assist clinician in decision making while some misclassification occurs, particularly with signals from chronic obstructive pulmonary disease (COPD) patients being misclassifyied as normal or asthma, the overall detection error for each disease remains low.

## 3.2 Proposed System

For construction of the neural network model we have used five types of layer 1) GRU(Gated Recurrent Unit), 2) Leaky Relu(Leaky version of a Rectified Linear Unit), 3) Dense Layer, 4) Dropout Layer, 5) Add Layer.

### 3.2.1 Gated Recurrent Unit(GRU)

It aims to solve the problem of vanishing gradient of a standard Recurrent neural layer. It has two types of gate - update gate and reset gate. These gates decide what information should be passed to the output. Update gate decides how much of the past memory is to need to be passed along to the future. Reset gate decides how much of past memory to forget. In our model, every GRU layer had linear activation function and the layers processed the input sequence backward and return the reverse sequence.

### 3.2.2 Leaky Relu(Leaky version of a Rectified Linear Unit)

The Leaky Rectified Linear Unit (Leaky ReLU) is an activation function commonly used in neural network architectures, serving as an alternative to the traditional

Rectified Linear Unit (ReLU). While the standard ReLU function replaces negative values with zeros, the Leaky ReLU allows a small, non-zero gradient for negative inputs, preventing neurons from becoming inactive. Mathematically, the Leaky ReLU function is defined as $f(x) = x$ if $x > 0$, and $f(x)$

$\alpha x$ if $x \leq 0$, where $\alpha$ is a small positive constant. By introducing this small gradient for negative inputs, the Leaky ReLU addresses the "dying ReLU" problem encountered with standard ReLU activations, where neurons can permanently remain inactive during training. This helps to mitigate issues related to vanishing gradients and promotes more robust learning in deep neural networks.

### 3.2.3 Dense Layer

The Dense layer, also known as a fully connected layer, is a fundamental component in neural network architectures where each neuron in the layer is connected to every neuron in the preceding layer. In a Dense layer, each neuron computes a weighted sum of its input values, adds a bias term, and applies an activation function to produce an output. These outputs serve as inputs to the neurons in the subsequent layer, forming the basis for information propagation through the network. Dense layers are essential for learning complex nonlinear relationships between features in the input data, enabling neural networks to model intricate patterns and make accurate predictions across a wide range of tasks, including classification, regression, and generative modeling.

### 3.2.4 Dropout Layer

The Dropout layer is a regularization technique commonly used in neural network architectures to prevent overfitting and improve generalization performance. During training, the Dropout layer randomly deactivates a fraction of neurons in the network by setting their outputs to zero with a specified probability, typically between 0.2 and 0.5. By introducing this stochastic element, Dropout effectively simulates training multiple different network architectures with shared parameters, forcing the network to learn more robust and redundant representations of the data. This reduces the network's reliance on specific neurons or features, thereby enhancing its ability to generalize to unseen data. Dropout layers are particularly effective in deep neural networks with many parameters, where overfitting is a common challenge.

### 3.2.5 Add Layers

The Add layer is a simple element-wise addition operation commonly used in neural network architectures to combine the outputs of multiple layers or branches within the network. This operation adds corresponding elements from input tensors, producing a single output tensor with the same shape. Add layers are often employed in skip connections or residual connections, where the output of one layer is added to the output of another layer situated deeper in the network. This facilitates the flow of information across different layers and helps alleviate the vanishing gradient problem by providing shortcut paths for gradients to propagate through the network during training. Skip connections have been shown to enhance the training of deep neural networks, leading to faster convergence and improved performance on various tasks, including image classification, object detection, and semantic segmentation.

## Advantages:

1. Prediction is easy.
2. Noise is eliminated.

3. Used to predict more diseases than the existing system.
4. Cheaper compared to the existing system.

## 3.3 Requirement Analysis and Specification

The requirement engineering process of feasibility study, requirements elicitation and analysis, requirement specification, requirements validation and requirement management. Requirement elicitation and analysis is an iterative process that can be represented as a spiral of activities,namely requirements discovery, requirements classification and organization, requirement negotiation and requirements documentation.

### 3.3.1 Input Requirement

The input requirement at the base requires data from users such as the audio format file that helps the DL model to predict the type of disease. The extension of the file should wav. The system the takes the input from the browse file option were we need to upload our audio file. The input must be the audio file to classify the type of pulmonary that the patients lung is suffering from.

### 3.3.2 Output Requirement

The output requirement depends on the dataset given by the patient and it shows whether the patient is healthy or not as well as what type of respiratory disease the patient suffers. When the input is taken as audio file from the user the system reads the audio file and generates a button named generate prediction and by clicking that button we will see the response that the type of the lung disease and chances to predict that type of disease.

## 3.4 Feasibility Study

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

### 3.4.1 Technical Feasibility

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible. Technical feasibility is a critical aspect of any project evaluation process, ensuring that proposed solutions can be effectively developed and implemented from a technological standpoint. This evaluation involves several key considerations. Firstly, a thorough assessment of available technology solutions, including software frameworks, programming languages, and hardware platforms, is necessary to align with project objectives. Secondly, the evaluation of technical resources, such as skilled personnel and necessary infrastructure, is vital to determine if they are readily available or require acquisition.

Compatibility and integration with existing systems and external services must also be evaluated to ensure seamless interoperability. Additionally, scalability and performance metrics must be analyzed to ascertain the system's ability to handle future growth and varying loads effectively. Security measures and compliance with regulations are crucial considerations to protect data integrity and privacy. Finally, identifying potential technical risks and challenges allows for proactive mitigation strategies to be developed. Conducting a comprehensive technical feasibility study provides valuable insights into the project's technological aspects, enabling informed decision-making and increasing the likelihood of successful implementation.

### 3.4.2 Economic Feasibility

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn't incur any kind of increase in the expenses. Programming Language for Web-App development is open source and readily available for all users. Since the project is implemented in Anaconda (Jupyter Notebook) it is cost efficient. In a feasibility study, evaluating technical feasibility is paramount to ascertain whether a proposed project can be effectively developed and implemented using available resources and technologies. This assessment encompasses various crucial components.

Firstly, a thorough examination of available technologies, including software frameworks and hardware platforms, is conducted to ensure alignment with project objectives. Additionally, an assessment of technical resources such as skilled personnel and infrastructure is undertaken to determine if the necessary expertise and tools are readily accessible. Compatibility and integration with existing systems are also evaluated to ensure seamless interoperability. Furthermore, scalability and performance metrics are analyzed to ascertain the system's ability to handle anticipated loads and accommodate future growth. Security measures and compliance with regulatory standards are essential considerations to safeguard data integrity and privacy. By conducting a comprehensive technical feasibility study, organizations can make informed decisions, mitigate risks, and increase the likelihood of successful project implementation.

## 3.5 Hardware Requirements

*Table 3-1 Hardware Requirements*

| | |
|---|---|
| HARD DISK | >90GB |
| PROCESSOR | >Core i3 2.4GHz |
| SYSTEM TYPE | 32bit / 64 bit |
| RAM | >2GB |

To ensure smooth operation and optimal performance of a system, it's crucial to consider the hardware requirements, which encompass various components such as the hard disk, processor, system type, and RAM. For the hard disk, a minimum capacity of over 90GB is recommended to accommodate the operating system, applications, and user data comfortably. Moving on to the processor, a Core i3 clocked at 2.4GHz or higher is preferred to handle computational tasks efficiently. Regarding the system type, compatibility with both 32-bit and 64-bit architectures is advisable to cater to different software requirements and maximize flexibility. As for RAM, a minimum of 2GB is essential to ensure smooth multitasking and responsiveness, enabling the system to handle multiple applications simultaneously without significant slowdowns. By meeting or exceeding these hardware requirements, users can experience reliable performance and seamless operation across various computing tasks and applications.

## 3.6 Software Requirements

*Table 3-2 Software Requirements*

| | |
|---|---|
| OPERATING SYSTEM | WINDOWS 7/8/8.1/10 |
| INTEGRATED DEVELOPMENT KIT | ANACONDA V2019 |
| PROGRAMMING LANGUAGE (BACKEND) | PYTHON V 3.6 |

| | |
|---|---|
| PROGRAMMING LANGUAGE (FRONTEND) | STREAMLIT |
| DATABASE | PYTHON SQLITE 3 |
| APPLICATION PROGRAMMING INTERFACE (API) | SCIKIT LEARN, FLASK |

In considering software requirements for a system, several key components must align with the desired functionality and compatibility. First and foremost, the choice of operating system plays a pivotal role, and compatibility with Windows 7, 8, 8.1, or 10 is essential for seamless operation. Additionally, utilizing an integrated development kit (IDE) such as Anaconda v2019 facilitates efficient coding and package management within the Python environment. Speaking of Python, version 3.6 or higher serves as the preferred backend programming language due to its versatility and extensive library support.

On the frontend, Streamlit emerges as a user-friendly option for building interactive web applications with ease. For database management, Python SQLite 3 offers a lightweight yet robust solution for storing and retrieving data efficiently. Lastly, incorporating application programming interfaces (APIs) like scikit-learn and Flask enhances the system's capabilities, enabling machine learning and web development functionalities to be seamlessly integrated. By meeting these software requirements, users can leverage a cohesive and powerful ecosystem to develop, deploy, and maintain their applications effectively.

## 3.7 Software Specification

### 3.7.1 Google Colaboratory

Google Colaboratory, often referred to as Google Colab, is a cloud-based platform by Google offering a Jupyter notebook environment for Python programming. It enables users to write, execute, and share Python code directly within a web browser without the need for any installation or setup. One of its key advantages is its provision of free access to high-performance hardware resources such as GPUs and TPUs, facilitating the execution of computationally intensive tasks, particularly in the fields of data science, Deep learning, and deep learning.

Colab supports collaboration in real-time, akin to Google Docs, allowing multiple users to work on the same notebook simultaneously. Integration with Google Drive enables seamless saving and sharing of notebooks. With built-in support for various Python libraries, including TensorFlow, PyTorch, Pandas, NumPy, and others, Colab simplifies the process of data analysis, model development, and experimentation. Its versatility, ease of use, and integration with Google's ecosystem make it a popular choice among data scientists, researchers, and developers for prototyping, learning, and collaborative projects.

### 3.7.2 Python

Python is an interpreter, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming.

Python is often described as a "batteries included" language due to its comprehensive standard library. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP. Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs. Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects. Python is a Beginner's Language .Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

### 3.7.3 Streamlit

Streamlit is an open-source Python library that makes it easy to create web applications for Deep learning, data analysis, and visualization. It allows developers and data scientists to quickly build interactive web apps directly from Python scripts, without needing to know web development languages like HTML, CSS, or JavaScript.

With Streamlit, you can create interactive dashboards, data exploration tools, and prototype Deep learning models with just a few lines of Python code. It provides a simple and intuitive API for creating various components such as sliders, buttons, text inputs, plots, and more, which are automatically rendered into a web interface.

Streamlit is designed to be fast and easy to use, making it popular among data scientists and developers for rapidly prototyping and sharing their work with others. It's particularly well-suited for tasks like data visualization, model prototyping, and deploying Deep learning models for demonstration or testing purposes.

### 3.7.4 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010. Pandas is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

### 3.7.5 Scikit-Learn

Scikit-learn is a free software Deep learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is the most useful and robust library for Deep learning in Python. It provides a selection of efficient tools for Deep learning and statistical modeling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

# 4 System Design

System design is a critical phase in the development process, encompassing the creation of a blueprint that outlines the architecture, components, and functionality of the system. At its core, system design aims to translate requirements into a structured plan that guides implementation and ensures the system meets its objectives effectively. The design process typically begins with the identification of system requirements, followed by the creation of a conceptual design that outlines high-level system architecture and interactions. Subsequently, detailed design specifications are developed, specifying the functionality of each component, data flow, and interfaces.

Throughout the design phase, considerations such as scalability, performance, security, and maintainability are carefully addressed to ensure the system meets both current and future needs. Collaboration between designers, developers, and stakeholders is essential to iteratively refine the design and incorporate feedback. Ultimately, the output of the system design phase is a comprehensive design document that serves as a roadmap for implementation, guiding developers in building a system that fulfills the desired functionality while adhering to best practices and industry standards.

## 4.1 ER Diagram

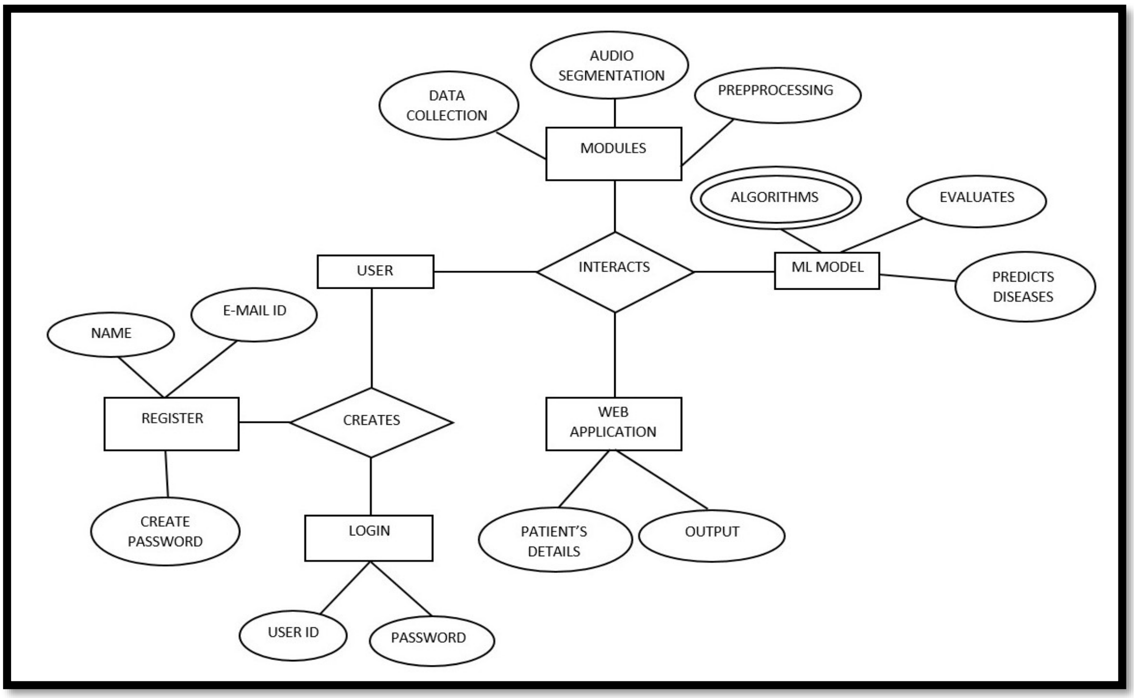


*Figure 4-1 ER Diagram*

A block diagram depicting a medical diagnosis system can be broken down into several key stages. First, patient information is collected, likely through an online platform where users register and provide basic details. This data undergoes preprocessing, which ensures its cleanliness, proper formatting, and compatibility with the system's core component: a machine learning model. This model is trained on existing data to identify patterns linked to specific diseases. Once trained, the model can analyze information from new patients, predicting potential illnesses based on the discovered patterns.

Finally, the system's performance is evaluated using separate test data to confirm its accuracy and ability to generalize to unseen cases. It's important to remember that such diagrams offer a simplified view. Real-world systems might involve multiple models, incorporate natural language processing, and have a higher degree of complexity. Additionally, these models, while helpful, shouldn't replace professional medical advice due to their inherent potential for errors.

## 4.2 Data Dictionary

A data dictionary, or metadata repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata. The term "schema" in the context of databases and database management systems (DBMS) encompasses several closely related meanings. Firstly, it can refer to a document that describes the structure and organization of a database or a collection of databases. This schema document outlines the relationships between data elements and serves as a guide for administrators and developers in managing the database effectively. Secondly, within the realm of DBMS, a schema is an integral component that defines the structure of the database

. This includes specifying tables, fields, relationships, and constraints, providing a framework for organizing and enforcing data integrity. The schema acts as a blueprint for how data is stored and accessed within the database system. Lastly, "schema" can also denote a piece of middleware, serving as an intermediary layer between applications and the database. This middleware facilitates data access and manipulation by providing functionalities such as query optimization and security mechanisms. Overall, the concept of schema is fundamental to database management, offering a structured approach to data modeling, organization, and access within a DBMS ecosystem.

## 4.3 Data Flow Diagram

A picture is worth a thousand words. A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

It is usually beginning with a context diagram as the level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of details for decomposing function really depends on the complexity that function.

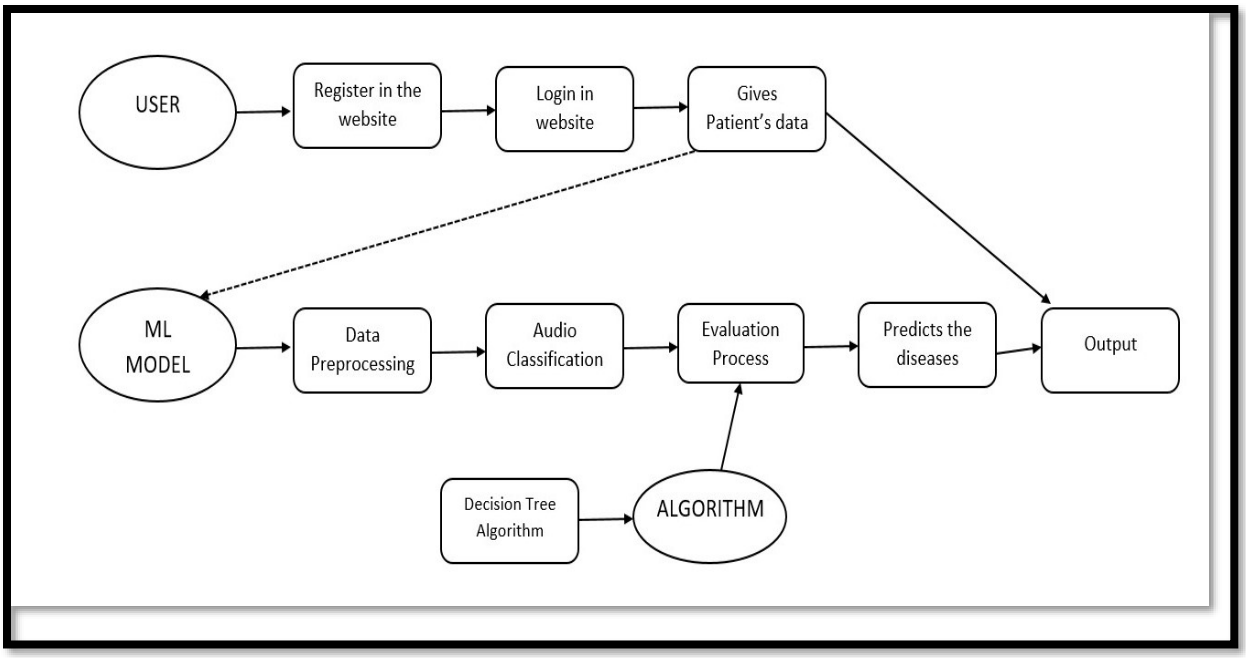


*Figure 4-2 DataFlow Diagram*

## 4.4 UML Diagrams

Unified Modeling Language (UML) is a standardized visual modeling language renowned for its versatility in capturing various aspects of software solutions, application structures, system behaviors, and business processes. With its rich set of diagram types, UML offers a comprehensive framework for modeling diverse behaviors within complex systems. There are 14 distinct UML diagram types, each tailored to specific modeling needs. These diagrams facilitate the modeling of business processes, enabling stakeholders to visualize and analyze workflows and interactions. Additionally, UML diagrams aid in the analysis, design, and implementation phases of software-based systems, providing a common language for communication among stakeholders, developers, and designers. By leveraging UML, organizations can streamline the development process, enhance collaboration, and ensure alignment between business requirements and technical solutions.

Unified Modeling Language (UML) serves as a ubiquitous language for various stakeholders including business analysts, software architects, and developers, facilitating the description, specification, design, and documentation of both existing and new business processes, as well as the structure and behavior of software artifacts. The specification process involving UML provides invaluable guidance for teams by delineating the order of activities, specifying the artifacts to be developed, directing the tasks of individual developers and the team as a whole, and offering criteria for monitoring and measuring the products and activities of a project. By adhering to UML standards, teams can streamline their development efforts, ensure consistency and clarity in communication, and foster alignment between project goals and implementation strategies, ultimately leading to more efficient and successful project outcomes.UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is Rational Unified Process (RUP).UML is not complete, and it is not completely visual. Given some UML diagram, we can't be sure to understand depicted part or behavior of the system from the diagram alone.

Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams. For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases. Name of an abstract classifier is shown in italics while the final classifier has no specific graphical notation, so there is no way to determine whether the classifier is final or not from the diagram.

### 4.4.1 Use Case Diagram

As the most known diagram type of the behavioral UML diagrams, use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact. It's a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system. You can create use case diagrams using our tool and/or get started instantly using our use case templates.

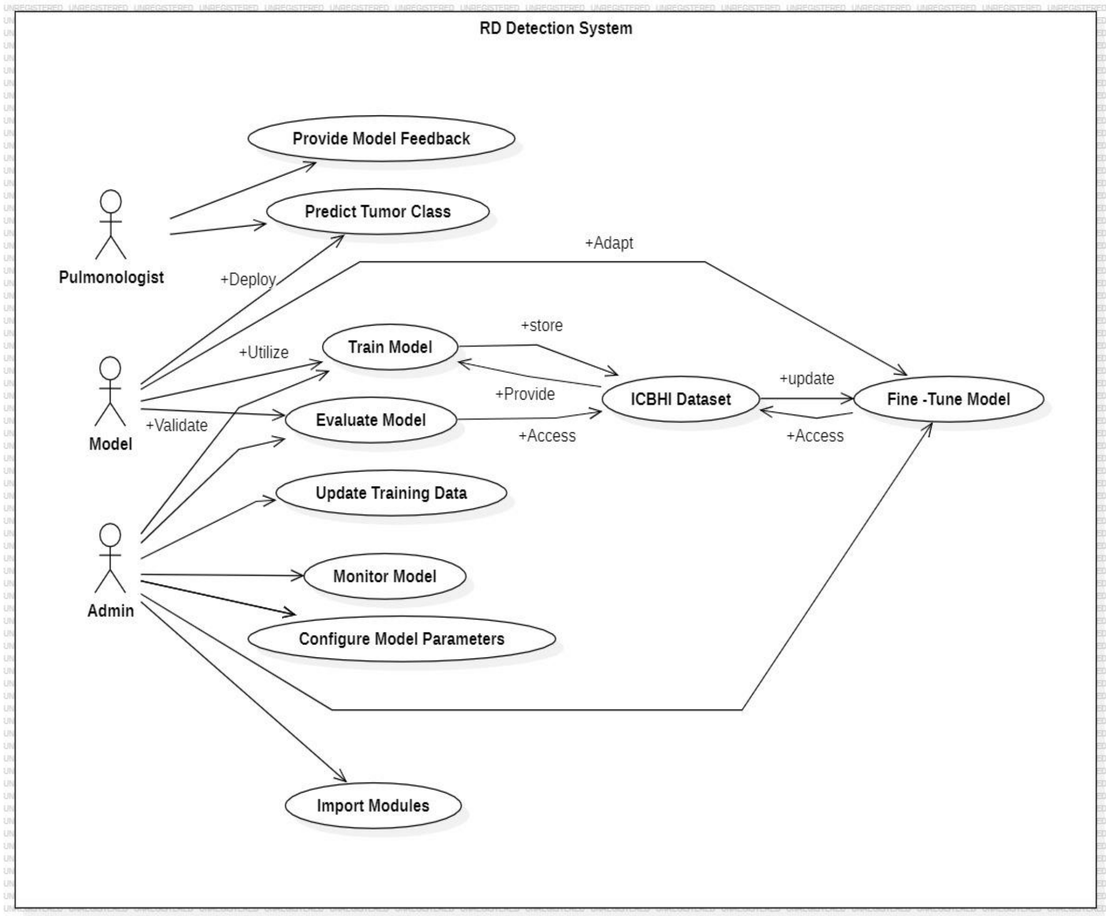


*Figure 4-3 Use Case Diagram*

### 4.4.2 Class Diagram

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom. In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

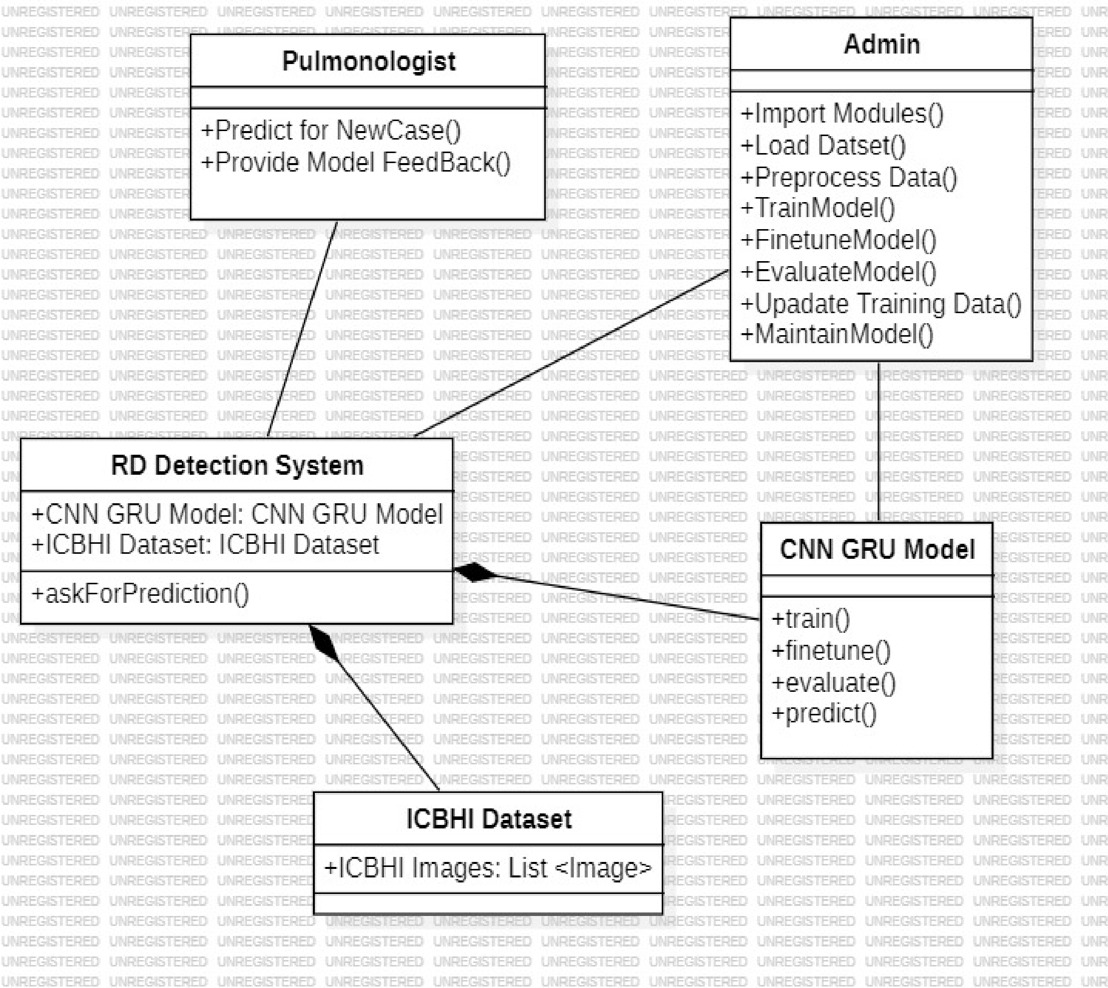


*Figure 4-4 Class Diagram*

### 4.4.3 Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It's important to note that they show the interactions for a scenario. The processes are represented vertically, and interactions are shown as arrows. This article explains the purpose and the basics of Sequence diagrams. Also, check out this complete Sequence Diagram Tutorial to learn more about sequence diagrams. You can also instantly start drawing using our sequence diagram templates.

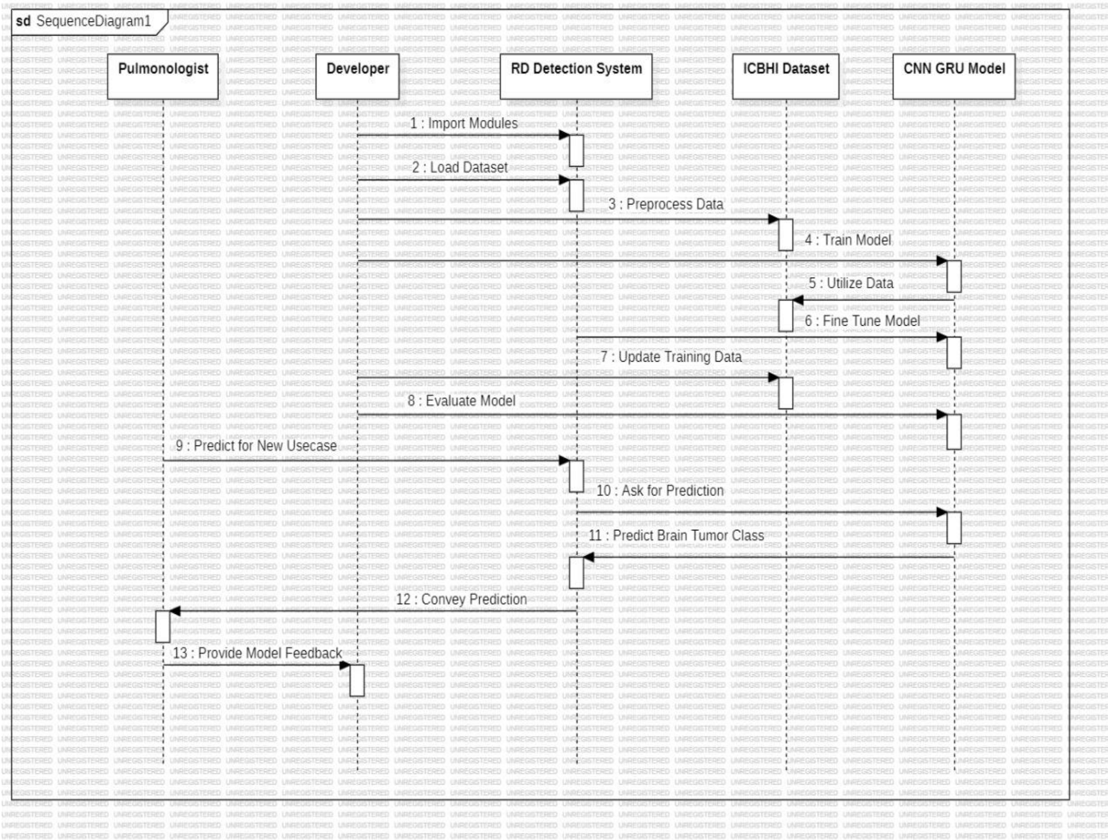


*Figure 4-5 Reference Diagram*

# 5 System Architecture

In the system architecture of the aforementioned project, a robust and scalable framework is established to accommodate the complexities of software solutions and business processes. This architecture encompasses the structural design of the system, including its components, interactions, and deployment strategies. At its core, the system architecture defines the arrangement of software modules, databases, interfaces, and external dependencies, ensuring cohesion and flexibility across the entire system.

It delineates the flow of data and control within the system, guiding the implementation of key functionalities and features. Additionally, the system architecture addresses key non-functional requirements such as performance, scalability, security, and maintainability, ensuring that the system meets both current and future needs. By providing a clear blueprint for system design and implementation, the architecture fosters collaboration among stakeholders and enables efficient development and deployment processes. Ultimately, a well-defined system architecture is essential for achieving the project's objectives, delivering a reliable and scalable solution that aligns with business goals and user requirements.

## 5.1 Architecture Overview

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. In the architecture overview of the aforementioned project, a comprehensive framework is established to guide the design and implementation of software solutions and business processes. This segment provides a high-level view of the system's architecture, outlining key components, interactions, and functionalities.

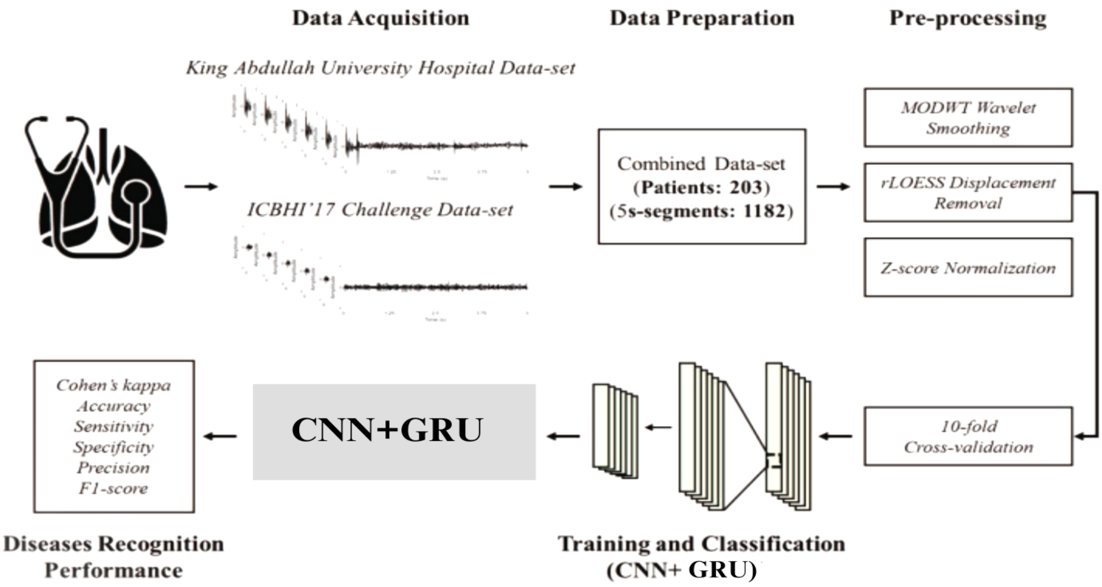


*Figure 5-1 Architecture Design*

## 5.2 System Module

The Sound Classification System comprises distinct modules, each serving specific functions essential for accurate and efficient classification of respiratory diseases based on audio data. Module 1 focuses on data collection and audio segmentation, where raw audio recordings are acquired and segmented into relevant sections for analysis. Module 2, preprocessing, and cleaning, are crucial steps aimed at enhancing the quality of the audio data by removing noise, normalizing volumes, and addressing any artifacts that may affect the classification process.

Following this, Module 3 involves the implementation of algorithms designed to extract meaningful features from the preprocessed audio data. These algorithms may include techniques from signal processing and machine learning to capture relevant patterns indicative of respiratory conditions. Finally, Module 4 is dedicated to respiratory disease classification, where the extracted features are utilized to classify audio segments into different disease categories such as asthma, pneumonia, or COPD. Through the orchestrated functioning of these modules, the Sound Classification System enables accurate and timely diagnosis of respiratory diseases, ultimately contributing to improved healthcare outcomes.

### 5.2.1 Modules Explaination

#### 5.2.1.1 Data collection and audio segmentation

In this module the data is collected in the form of audio, these audio files are stored and they are converted in the form of a text file, this is how the data is collected. These data are collected from different patients' lungs and it is recorded. Data virtualization of audio and text files.

#### 5.2.1.2 Preprocessing and cleaning

Data preprocessing involves the transformation of the raw dataset into an understandable format. Preprocessing data is a fundamental stage in data mining to improve data efficiency. The data preprocessing methods directly affect the outcomes of any analytic algorithm. These data are pre-processed and we have to clean the unwanted values all the null values.

#### 5.2.1.3 Implementation of algorithm

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Once everything is set up, it is time to actually implement the algorithm. Implementation of various classification algorithms is done. With these algorithms we can easily classify the sounds easily.

#### 5.2.1.4 Respiratory disease classification

In this module it will classify whether the person is healthy or infected. It gives either of these as a result. If the person is infected it will identify the type of disease and display the disease name else it will display the person is normal.

## 5.3 Program Language Design

In the Program Language Design aspect of the project, careful consideration is given to selecting the most suitable programming languages and frameworks to fulfill the system requirements effectively. This involves evaluating various factors such as the project's objectives, technical feasibility, scalability, and maintainability. The choice of programming languages influences the system's architecture, performance, and ease of development, making it a crucial decision. Additionally, designing the program language involves defining coding standards, conventions, and best practices to ensure consistency and readability across the codebase.

### 5.3.1 Disease Prediction Algorithm

**Step 1:** Reading the pre-processed data csv file.

**Step 2:** Define variables X and Y from dataset.

**Step 3:** Encoding categorical data e.g. gender as a dummy variable.

**Step 4:** Encoding categorical data e.g. disease outcome as a dummy variable.

**Step 5:** Splitting the dataset into the Training set and Test set.

**Step 6:** Fitting Classifier to the Training Set (CNN+GRU Classifier)

**Step 7:** Creating a function for Annotation data for identifying recording_info and recording_annotations.

**Step 9:** Summed number of crackles / wheezes are normalized by the duration of the recording

duration = annotation.iloc[-1, 1] - annotation.iloc[0, 0]

**Step 10:** prediction = classifier.predict(pred)

# 6 Testing

In the testing phase of the project, rigorous evaluation and validation of the system's functionalities are conducted to ensure its reliability, performance, and compliance with requirements. This involves developing comprehensive test plans and scenarios to cover all aspects of the system, including functional, non-functional, and user acceptance testing. Through techniques such as unit testing, integration testing, and system testing, potential defects and inconsistencies are identified and addressed systematically. Additionally, performance testing is carried out to assess the system's responsiveness, scalability, and stability under varying conditions. The testing phase plays a crucial role in validating the system's behavior, detecting any defects or deviations from expected outcomes, and ensuring that the final product meets the desired quality standards. By prioritizing thorough testing, the project team can mitigate risks, enhance the user experience, and deliver a robust and reliable solution to stakeholders.

## 6.1 System Testing

The testing and the evaluation part are the most important part in any research or development of a model. As we need to understand and check whether our model is performing the way it should be.

## 6.2 Web Application Testing

We use the flask framework for developing the webapp, along with CSS and html. On starting the application, it is bonded to a default address http://127.0.0.1:5000/

```
 * Serving Flask app "respiratory_disease_prediction" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 220-238-643
```

*Figure 6-1 WebApplication Testing*

On entering the diseases prediction page, the best performing algorithm, CNN+GRU is called to do the classification for the input data. The input is then validated with an algorithm and the training data from our model and predicts the result for the input given by the user.

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion='entropy',max_depth=3, random_state=42)
classifier.fit(X_train, y_train)
prediction2=[]
```

*Figure 6-2 WebApplication Testing Inputs*

## 6.3 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

*Table 6-1 Test Cases*

| Test Case Id | Test Cases | Priority | Input Test Data | Test Case Description | Expected Results | Actual Results | Pass/Fail |
|---|---|---|---|---|---|---|---|
| TU02 | Deep learning model | A | Patient's diagnosis data | Splitting up the data for training and testing | Training with good accuracy | Trained Deep learning model | Pass |
| TU03 | Upload med.details | A | Enter med.details | Update med.details | Should upload successfully | Details are stored successfully | Pass |
| TU04 | Upload audio files | A | Path wave Extension file | Considered as ML model training datasets | Successfully uploaded the audio file | Correctly uploaded the audio file | Pass |
| TU05 | Type of disease | A | Disease Name display | Disease specification | Name of disease | Patient condition | Pass |

# 7 Conclusions and Future Enhancements

In conclusion, the project has successfully achieved its objectives of developing a Sound Classification System capable of accurately identifying respiratory diseases based on audio data. Through the integration of modules for data collection, preprocessing, algorithm implementation, and disease classification, the system demonstrates its efficacy in aiding healthcare professionals in diagnosing respiratory conditions promptly and accurately. However, there is room for future enhancements to further improve the system's capabilities.

Potential areas for enhancement include expanding the dataset to encompass a broader range of respiratory diseases, refining algorithms for enhanced accuracy, and incorporating real-time monitoring features for continuous assessment of patients' respiratory health. Additionally, integrating advanced machine learning techniques and leveraging emerging technologies such as artificial intelligence and edge computing could further enhance the system's performance and usability. Overall, the project lays a solid foundation for future advancements in respiratory disease diagnosis and underscores the importance of leveraging technology to improve healthcare outcomes.

## 7.1 Conclusions

In our project, we have predicted respiratory diseases from the respiratory sounds database using CNN+GRUs. Our work comprised of the comparison with Deep learning algorithm. Our project mainly consists of three areas, preprocessing of the data, prediction of diseases, and developing the interface for users to use. In the preprocessing, we are handling the missing data, normalizing the values and eliminating any unwanted data from our dataset and creating a new preprocessed data.

The prediction with CNN+GRUs gives an accuracy rate of 90 percent. CNN models can be used when there is a large amount of data is available to train, when learning with less data the CNN model is not as appropriate as over fitting occurs in the training of model. The more the accuracy and f1 average weight of the algorithms the model's predictions become accurate.

## 7.2 Future Enhancements

The future work of the application consists of mostly collecting more data and trying to implement with the CNN model, also instead of manual annotation of the audio files teach the model to automatically annotate the recordings. This web application can add storage functionalities where the users can access their previous breath sound checks and also the automatic annotation process of sounds which helps the users to easily identify the disease. A desktop or mobile application can also be built to make the process easier for the users.

# 8 Implementation

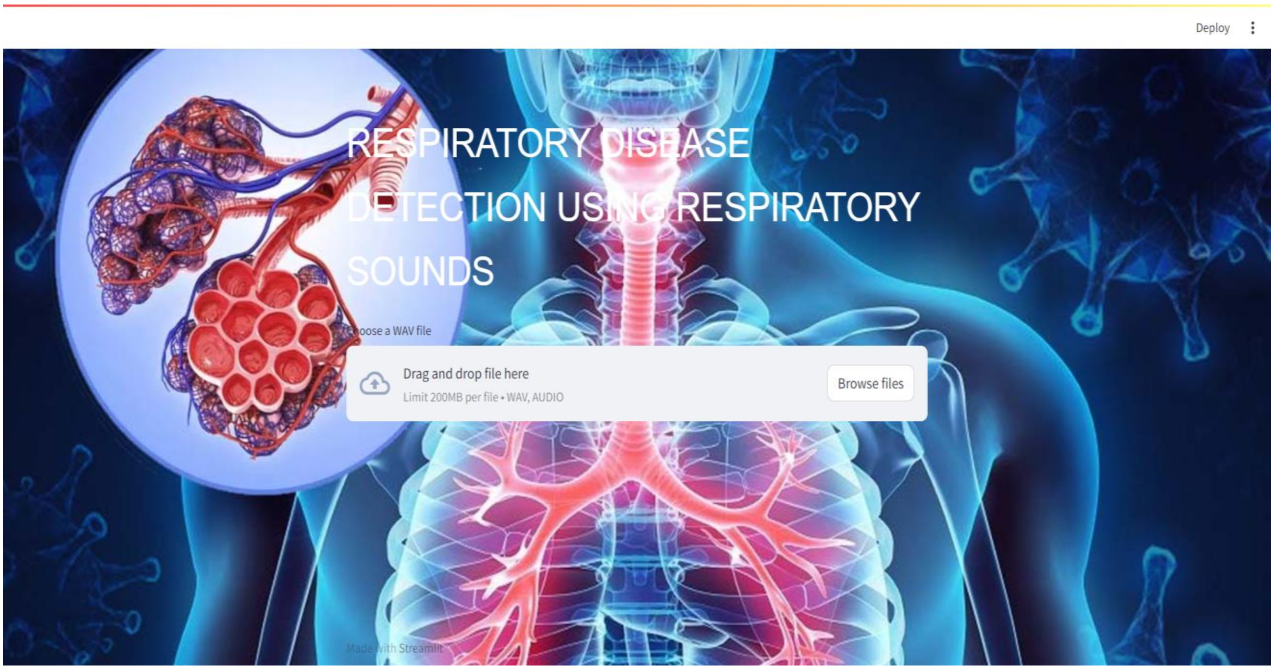


*Figure 8-1 Home Page*

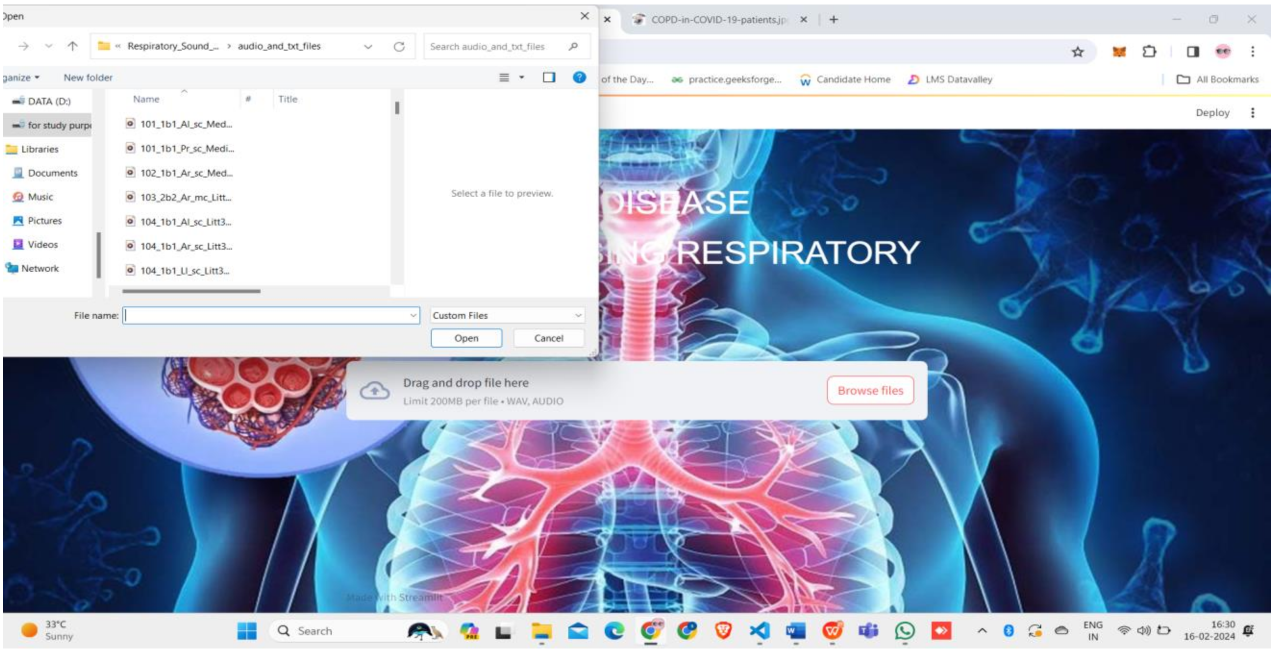


*Figure 8-2 Upload Audio file for Prediction*

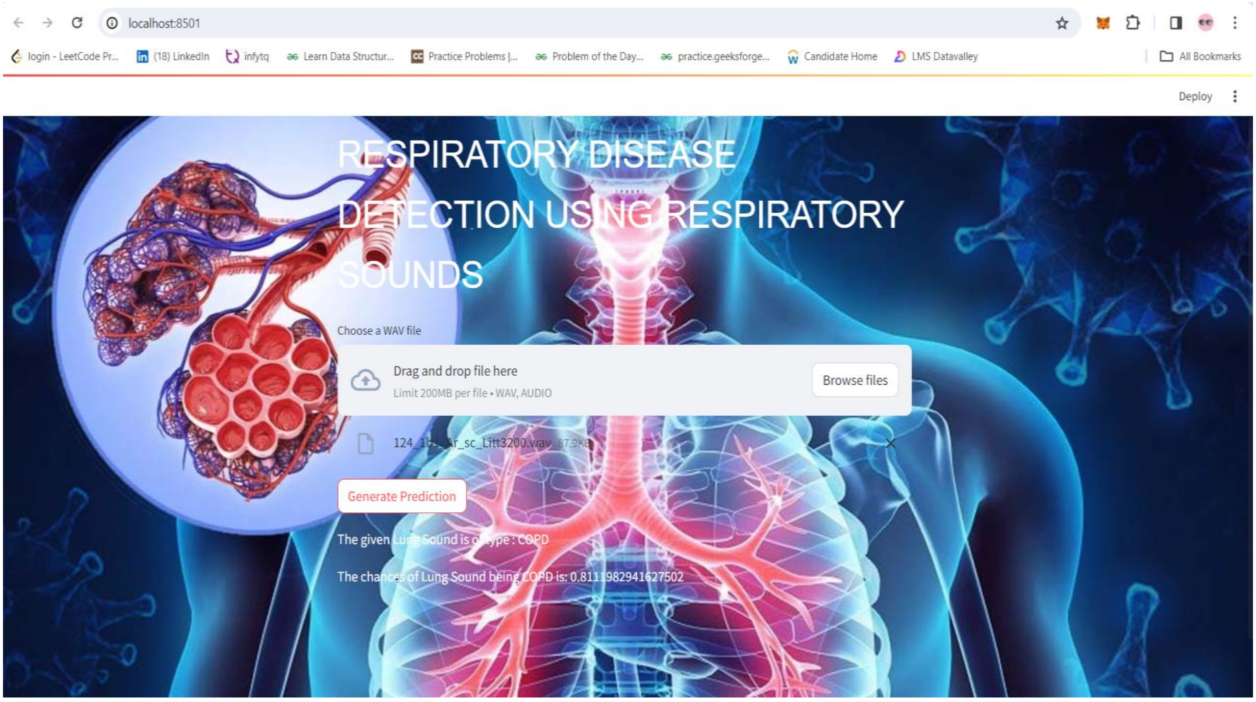


*Figure 8-3 Disease Prediction using Audio Fil*

**Github** **[Url: https://github.com/VelagamPujitha/Project-C15.git](https://github.com/VelagamPujitha/Project-C15.git)**

The project totally consists of 2 files:

**Traning_model**

The training of the model file involves a multi-step process aimed at harnessing the power of Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) for effective classification. Initially, the lung sound data undergoes preprocessing to extract relevant features and normalize the input. Subsequently, the CNN component is employed to automatically learn hierarchical representations of the input spectrograms, capturing both low and high-level features crucial for disease identification. Following this, the GRU network, known for its ability to model sequential data, is utilized to capture temporal dependencies within the spectrograms, enhancing the model's understanding of the dynamic nature of lung sounds. Through iterative training with labeled data, the model file gradually optimizes its parameters, fine-tuning the network's ability to accurately classify pulmonary diseases, ultimately culminating in a robust and reliable tool for diagnosing respiratory conditions.

**app.py**

The application presents an intuitive interface allowing users to browse and upload audio recordings of lung sounds. Upon submission, the uploaded audio undergoes preprocessing to ensure compatibility with the model's input requirements. Subsequently, the preprocessed audio is fed into the trained CNN-GRU model, which generates predictions regarding the presence of pulmonary diseases based on learned patterns and features extracted from the lung sound spectrograms. These predictions are then presented to the user, offering insights into potential respiratory conditions detected in the uploaded audio. Through seamless integration of cutting-edge machine learning techniques and user-friendly web interface design, the application empowers healthcare professionals and individuals alike to leverage advanced technology for early detection and management of pulmonary diseases.

# 9 References


[1] W. P.D , “'The Stethoscope:Some Preliminary Investigations',” *Postgraduate Medical Journal,* 2003.

[2] S. S and M. A.N., “'Digital Stethoscope:Technology Update',” *Medical Devices,* 2018.

[3] R. Sandra and R. Gass, “'Analysis of Respiratory Sounds:State of Art',” *Respiratory and Pulmonary Medicine',* 2008.

[4] K. H, G. M and S. K, “'Auditory detection of simulated crackles in breath sounds',” 2001.

[5] C. I, C. P and M. N, “'Respiratory Sound Database for the development of Automated Classification',” *Precision Medicine Powered by PHealth and Connected Health,* 2018.