

# PULMONARY AILMENT CLASSIFICATION

# USING PHONOPNEUMOGRAPHY

**A PROJECT REPORT**

***Submitted by***

# ARSHIYA SULTHANA T DIVYA SHREE S PRUTHVIKA M VASUMATHI R

**AC20UIT006 AC20UIT018 AC20UIT035 AC20UTI049**

*In partial fulfilment for the award of the degree of*

# BACHELOR OF TECHNOLOGY

*in*

# INFORMATION TECHNOLOGY

**ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)**

# Dr. M.G.R. NAGAR, HOSUR 635109

**ANNA UNIVERSITY: CHENNAI 600 025**

# MAY 2024

ANNA UNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

Certified that this project report “**RESIDENTIAL PROPERTY VALUATION”** is the bonafide work of **ARSHIYA SULTHANA T (AC20UIT006), DIVYA SHREE S (AC20UIT018), PRUTHVIKA M**

**(AC20UIT035), VASUMATHI R (AC20UIT049)”** who carried out the project work under my supervision.

# Dr. D. THILAGAVATHY M.E., PhD HEAD OF THE DEPARTMENT PROFESSOR,

Department of IT,

Adhiyamaan College of Engineering, (Autonomous),

Hosur – 635109.

# Mrs. R. SUJATHA M.E., (Ph.D). SUPERVISOR

ASSOCIATE PROFESSOR,

Department of IT,

Adhiyamaan College of Engineering, (Autonomous),

Hosur – 635109.

Submitted for the VIVA-VOCE held on at

Adhiyamaan College of Engineering (Autonomous), Hosur.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

It is one of the most efficient tasks in Our life to choose the appropriate words to express one’s gratitude to the beneficiaries.

We are very much grateful to God who helped us all the way through the project and how molded us into what we are today.

We show Our sincere thanks to **Dr. G. RANGANATH M.E., Ph.D.,** Principal, Adhiyamaan College of Engineering Hosur, for this Valuable opportunity and environment given to us.

We extend our hearty gratitude to **Dr. D. THILAGAVATHY M.E., Ph.D.,** Professor and Head of the Department, Department of Information Technology, Adhiyamaan College of Engineering (Autonomous), Hosur, for her guidance and valuable suggestions and encouragement throughout this project.

We are highly indebted to **Mrs. R. SUJATHA M.E., (Ph.D).** Associate Professor, Department of Information Technology, Adhiyamaan College of Engineering, Hosur, whose immense support encouragement and valuable guidance were responsible to complete this project successfully.

We extend our hearty thankfulness to our Project Coordinators and all our Department faculty for their support in helping us complete project successfully.

Finally, we would like to thank to our parents, without their motivational and support would not have been possible for us to complete this project successfully.

**ABSTRACT**

Lung auscultation is one of the most popular diagnostic modalities used by the pulmonary experts to analyze the condition of the respiratory system. When auscultating various areas on the anterior and posterior sides of the chest , lung sounds can be detected. Lung sounds are indicative of different anatomical flaws in the lungs and provide accurate prognoses regarding respiratory health, resulting in more trustworthy medical tool for identifying respiratory disorders.

According to a recent study conducted by the world health organization (WHO), approximately ten million (M) people die each year as a result of respiratory diseases. In order to analyze respiratory sounds on a computer, we developed a cost-effective and easy-to-use Algorithm that can be used with any device. Employed two types of machine learning algorithms; Gammatone Cepstrum Coefficients Features in a Convolutional Neural Network and Since using GTCC and STFC features with a CNN-LSTM algorithm. We prepared four data sets for CNN-LSTM algorithm to classify respiratory audio: (1) healthy versus pathological classification; (2) rale, rhonchus, and normal sound classification; (3) singular respiratory sound type classification; and (4) audio type classification with all sound types.

# KEYTERMS:

Respiratory Disease classification, Pulmonary disease classification, Lung disease classification, Lung disease classification based on lung sound.

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| **4.1** | ARCHITECTURAL DIAGRAM | **8** |
| **4.2** | USECASE DIAGRAM | **9** |
| **4.3** | CLASS DIAGRAM | **10** |
| **4.4** | ACTIVITY DIAGRAM | **11** |
| **4.5** | SEQUENCE DIAGRAM | **12** |
| **4.6** | DATA FLOW DIAGRAM | **13** |
| **4.7** | DEPLOYMENT DIAGRAM | **14** |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE NAME** | **PAGE NO** |
| **5.2** | HARDWARE | **20** |
|  | REQUIREMENTS |  |

**RESIDENTIAL PROPERTY VALUATION**

# TABLE OF CONTENTS

CHAPTER NO TITLE PAGE NO ABSTRACT iv

LIST OF FIGURES v

[LIST OF TABLES vi](#_TOC_250011)

1. INTRODUCTION 1
   1. [OVERVIEW 1](#_TOC_250010)
   2. [PURPOSE 2](#_TOC_250009)
2. LITERATURE SURVEY 3
3. SYSTEM ANALYSIS 4
   1. [EXISTING SYSTEM 5](#_TOC_250008)
      1. DRAWBACKS 5
   2. [PROPOSED SYSTEM 6](#_TOC_250007)
      1. [ADVANTAGES 7](#_TOC_250006)
4. SYSTEM DESIGN 8
   1. ARCHITECTURAL DIAGRAM 8
   2. [USECASE DIAGRAM 9](#_TOC_250005)
   3. [CLASS DIAGRAM 10](#_TOC_250004)
   4. [ACTIVITY DIAGRAM 11](#_TOC_250003)
   5. [SEQUENCE DIAGRAM 12](#_TOC_250002)
   6. [DATA FLOW DIAGRAM 13](#_TOC_250001)
   7. [DEPLOYMENT DIAGRAM 14](#_TOC_250000)
5. SYSTEM SPECIFICATION 15
   1. SOFTWARE REQUIREMENS 15
   2. HARDWARE REQUIREMENTS 20
6. SYSTEM MODULE 22
   1. [BLUEPRINT 22](#_bookmark0)
   2. SPECIFICATION 23
7. CONCLUSION 24
   1. [CONCLUSION 24](#_bookmark1)
   2. [FUTURE WORK 25](#_bookmark2)
8. DATASET DESCRIPTION 26

[APPENDICES 27](#_bookmark3)

[SOURCE CODE 27](#_bookmark4)

[SCREENSHOTS 40](#_bookmark5)

REFERENCE 44

**CHAPTER 1 INTRODUCTION**

# OVERVIEW

# Respiratory diseases are the leading causes of death and disability worldwide, with the poorest regions having the greatest disease burden. Ageing and risk factors such as smoking, environmental pollution, and body weight also play a key role in this issue. Chronic respiratory diseases, including asthma, account for 7% of all deaths worldwide and are the third leading cause of death. Between 1990 and 2017, the number of deaths due to chronic respiratory diseases increased by 18%.

# Pneumonia kills millions annually, particularly among children under 5 years old. Over 10 million people develop tuberculosis (TB), which is the most common lethal infectious disease. Lung cancer kills 1.6 million people each year and is the deadliest cancer globally. Respiratory diseases make up five of the 30 most common causes of death: COPD, lower respiratory tract infection, tracheal, bronchial, and lung cancer, TB, and asthma. Over 1 billion people suffer from acute or chronic respiratory conditions.

# Lung diseases significantly affect people's social, economic, and health lives. Social deprivation is the most important factor affecting death and disability rates, with the highest rates seen in the poorest regions. Lower mortality rates are seen in more affluent countries due to better access to health services and improved treatments.

# The treatment of lung diseases is of great importance in the medical field, and research is ongoing for early diagnosis and intervention in respiratory diseases. However, 45% of WHO Member States report having less than one physician per 1000 population, which can lead to mistakes. Finding new ways to help doctors save time is a priority, and automatic and reliable tools can help diagnose more people and reduce errors due to work overload.

# PURPOSE

As rapid growth of respiratory diseases is witnessed around the world, medical research ﬁeld has gained interest in integrating potential audio signal analysis-based technique. From the past few decades, computer science constantly improving the ability to analyze media data automatically and with the help of diagnosis tools we are able to process image and/or audio information. Hence, Computer science could help nursing staff or doctors for diagnosis by proposing faster and reliable tools and by giving customizable tools for medical monitoring to the patient. Like in other application domains, audio signal analysis tools can potentially help in analyzing respiratory sounds to detect problems in the respiratory tract. Audio analysis aids in timely diagnosis of respiratory ailments more effortlessly in the early stages of a respiratory dysfunction. Apart from respiratory check-ups, every cardiac assessment also includes an audio auscultation in which one the medical specialist listens to sounds from the patient body with different tools like stethoscope or sonography. This shows how important sound analysis is for heart and lungs disease detection.

Respiratory sounds may be acquired by the easy and non-invasive auscultation procedure. Auscultation is an effective technique in which physicians evaluate and diagnose the disease after using a stethoscope for lung disease. This method is inexpensive and easy as it does not require internal intervention into the human body. However, traditional stethoscopes may be exposed to external noise sounds and cannot ﬁlter the audio frequencies of the body in auscultation and cannot create permanent recordings in monitoring of the disease course. Also, there is a possibility of untrained physicians incorrectly recognizing abnormalities, which can be due to not calibrating the instrument and/or due to noisy environment, is very high using this method.

As lung and heart diseases remains the leading cause of death globally, there are many studies about lung and heart sound identiﬁcation. Since then, there are lots of improvements, for processing records taken in noisy environments. Furthermore, new kinds of methods drastically improve the domain, as machine learning and deep learning. These approaches contribute a lot to computer vision, or audio analysis. This gives more relevant information from respiratory sounds extracted and contribute to reducing the time for diagnosis, consequently increasing treatment efﬁciency. Thus, an automated algorithm developed to recognize abnormalities in respiratory sounds may be of great relevance to clinical diagnosis. Also researchers are looking in to combining speech and signal processing tools techniques with image analysis-based tools techniques [4, 5, 6] can also help doctors predict or guess about the presence of respiratory diseases based on verbal communication before they even start with the X-ray screening or other procedures.

Machine learning has proven to be an effective technique in recent years and machine learning algorithms have been successfully used in a large number of applications. The development of computerized lung sound analysis has attracted many researchers in re- cent years, which has led to the implementation of machine learning algorithms for the diagnosis of lung sound. In our research we have used machine learning techniques in computer-based lung sound analysis. A brief description of the types of lung sounds and their characteristics is provided. We examined speciﬁc lung sounds/disorders, the number of subjects, the signal processing and classiﬁcation methods and the outcome of the analyses of lung sounds using machine learning methods that have been performed by previous researchers. Before diagnosing disease based on their types, it is important to ﬁrst ensure that whether a person has any lung infection. True positive case can then be pushed for further processing, such as diagnosis.

# CHAPTER 2 LITERATURE SURVEY

1. Yi Ma, Xinzi Xu, and Yongfu Li. Lungrnl: An improved adventitious lung sound classiﬁcation using non-local block resnet neural network with mixup data augmen- tation. In Interspeech, pages 2902–2906, 2020.

Their method employs a non-local block ResNet neural network along with mixup data augmentation to improve classification accuracy. By leveraging advanced neural network architectures and data augmentation techniques, their research aims to facilitate more precise diagnosis of respiratory conditions, offering potential benefits for medical applications. This innovation could significantly enhance automated diagnostic systems by providing more reliable assessments based on lung sound analysis, thus potentially improving patient care and medical outcomes in respiratory health.

1. Jyotibdha Acharya and Arindam Basu. Deep neural network for respiratory sound classiﬁcation in wearable devices enabled by patient speciﬁc model tuning. IEEE transactions on biomedical circuits and systems, 14(3):535–544, 2020.

The study focuses on utilizing deep neural networks for classifying respiratory sounds captured by wearable devices. They emphasize patient-specific model tuning to improve classification accuracy, which holds promise for personalized healthcare monitoring.

1. Himadri Mukherjee, Ankita Dhar, Sk Md Obaidullah, KC Santosh, Santanu Phadikar, and Kaushik Roy. Linear predictive coefﬁcients-based feature to identify top-seven spoken languages. International Journal of Pattern Recognition and Artiﬁcial Intelligence, 34(06):2058006, 2020.

The methodology involves extracting features using linear predictive coefficients (LPC) from spoken language signals. Advantages include LPC's ability to capture spectral characteristics efficiently, aiding in accurate language identification. LPC-based features offer a compact representation of speech, enhancing computational efficiency. However, potential disadvantages may include sensitivity to noise and variability in speech signals, which could affect classification accuracy. Despite this, the proposed approach contributes significantly to language identification systems, particularly for the top seven spoken languages, offering potential benefits for automated speech recognition, translation, and other language-based applications.

1. Bruno M Rocha, Dimitris Filos, Lus Mendes, Gorkem Serbes, Sezer Ulukaya, Yasemin P Kahya, Niksa Jakovljevic, Tatjana L Turu kalo, Ioannis M Vogiatzis, Eleni Perantoni, et al. An open access database for the evaluation of respiratory sound classiﬁcation algorithms. Physiological measurement, 40(3):035001, 2019.

The paper titled "An Open Access Database for the Evaluation of Respiratory Sound Classification Algorithms" by Bruno M Rocha et al. was published in Physiological Measurement in 2019. The study introduces a freely accessible database for assessing respiratory sound classification algorithms. This database aids researchers in evaluating and comparing the performance of algorithms designed for analyzing respiratory sounds. By providing a standardized dataset, the paper facilitates advancements in respiratory sound classification technology, potentially leading to improved diagnostic tools for respiratory conditions. This initiative promotes collaboration and innovation in the field of respiratory health monitoring and diagnosis.

1. Fatih Demir, Aras Masood Ismael, and Abdulkadir Sengur. Classiﬁcation of lung sounds with cnn model using parallel pooling structure. IEEE Access, 8:105376– 105383, 2020

Their research focuses on utilizing a Convolutional Neural Network (CNN) model with a parallel pooling structure for classifying lung sounds. This innovative approach aims to enhance the accuracy and efficiency of lung sound classification algorithms, potentially aiding in the diagnosis of respiratory conditions. By leveraging CNNs and parallel pooling, the study contributes to the advancement of automated diagnostic systems for respiratory health monitoring, offering promising implications for medical applications and patient care.

1. Hai Chen, Xiaochen Yuan, Zhiyuan Pei, Mianjie Li, and Jianqing Li. Triple- classiﬁcation of respiratory sounds using optimized s-transform and deep residual networks. *IEEE Access*, 7:32845–32852, 2019.

The study introduces a novel approach for classifying respiratory sounds into three categories using an optimized S-transform and deep residual networks. By combining signal processing techniques with deep learning methods, their research aims to improve the accuracy and reliability of respiratory sound classification, potentially enhancing diagnostic capabilities for respiratory conditions. This innovative methodology contributes to the advancement of automated diagnostic systems for respiratory health monitoring, offering promising implications for medical applications and patient care.

1. Kun-Hsi Tsai, Wei-Chien Wang, Chui-Hsuan Cheng, Chan-Yen Tsai, Jou-Kou Wang, Tzu-Hao Lin, Shih-Hau Fang, Li-Chin Chen, and Yu Tsao. Blind monaural source separation on heart and lung sounds based on periodic-coded deep autoencoder. IEEE Journal of Biomedical and Health Informatics, 24(11):3203–3214, 2020.

Their research presents a method for separating heart and lung sounds from mixed audio recordings using a periodic-coded deep autoencoder. This innovative approach combines deep learning techniques with signal processing to enable blind source separation, potentially improving diagnostic accuracy and facilitating better analysis of cardiac and respiratory health conditions.

1. Mukherjee Himadri, Hanan Salam, and KC Santosh. Lung health analysis: Ad- ventitious respiratory sound classiﬁcation using ﬁlterbank energies. *IEEE Journal of Biomedical and Health Informatics*, 2021.

The study focuses on analyzing lung health by classifying adventitious respiratory sounds using filterbank energies. Their research contributes to improving respiratory health assessment by employing signal processing techniques to accurately classify abnormal lung sounds, potentially aiding in the diagnosis and monitoring of respiratory conditions.

# CHAPTER 3 SYSTEM ANALYSIS

# EXISTING SYSTEM

Traditional diagnostic standards, radiographic imaging, spirometry and pulmonary function testing, artificial intelligence (AI) and machine learning-based systems, and telemedicine platforms are examples of respiratory illness categorization systems. In radiological imaging, illnesses including pneumonia, TB, and lung cancer are detected and classified using chest X-rays and CT scans. Traditional diagnostic criteria comprise symptoms, physical examination, and medical history. Asthma, restrictive lung disorders, COPD, and other lung function characteristics are measured by pulmonary function tests, which also provide information on forced vital capacity and FEV1.

To accurately categorize illnesses, machine learning and artificial intelligence (AI)-based systems examine data sources such as electronic health records, medical imaging, and patient demographics. Using algorithms to provide suggestions, CDSSs combine patient data with medical expertise to help healthcare professionals diagnose and treat respiratory disorders.

# DISADVANTAGES

Existing respiratory disease classification systems have made significant progress in diagnosis and management, but they still have several disadvantages. These include subjectivity and variability, which can lead to misdiagnosis or delayed diagnosis, and inaccuracy, which can result from radiological imaging not accurately distinguishing between respiratory conditions. Cost and accessibility are also significant issues, as radiological imaging and pulmonary function tests can be expensive and not easily accessible in all healthcare settings, especially in underserved areas. Time-consuming procedures like spirometry and interpretation of radiological images can delay diagnosis and treatment, leading to prolonged patient suffering and increased healthcare costs.

Limited sensitivity and specificity are also issues, as machine learning and AI-based systems may have limitations in sensitivity and specificity, particularly when trained on imbalanced or insufficient data. Privacy and data security concerns arise from AI-based systems relying on patient data, such as electronic health records (EHRs), which could compromise patient confidentiality and trust in healthcare systems.

Technological barriers, such as poor internet connectivity, limited access to digital devices, and lack of familiarity with technology among patients and healthcare providers, can also hinder the adoption and effectiveness of telemedicine for respiratory disease diagnosis and management. Addressing these disadvantages requires ongoing research and development efforts to improve the accuracy, accessibility, and cost-effectiveness of existing respiratory disease classification systems.

# PROPOSED SYSTEM

Automated algorithmic approach that can categorize lung sounds in a variety of diseased states. Another objective of this present research work is to propose a lightweight deep learning architecture that can classify lung sounds accurately while keeping parameter size and computational complexity less.

The majority of these respiratory diseases have almost similar kind of symptoms; therefore, it becomes difficult for the doctor to predict the actual disease just by hearing the lung sound only and requires additional tests, such as spirometry test. The novel contributions of the proposed framework are itemized as follows.

1. Designing a novel lightweight LSTM, namely, CNN\_LSTM, to classify the lung sound efficiently, while keeping the architecture lightweight in terms of total trainable parameters and model storage size.
2. Classification of seven respiratory diseases for the first time, utilizing three publicly available lung sound databases: ICBHI 2017 challenge database and chest wall lung sound database. Engaging all the databases also ensures the robustness of the classification mechanism, as the DLM is trained with a wide variety of lung sounds.
3. Computing the ablation study and classification report containing the statistics of layers, parameters, accuracy, precision, recall, F1 score, and so on, in order to have a thorough performance/classification accuracy analysis of the proposed lightweight CNN\_LSTM.
4. The patterns in these are then summarized as a feature Fused STFT and GTCC set. The features are used to train a model used to classify sounds as either normal or crackles. There are several methods for feature extraction, and several classifiers with associated learning methods.

# ADVANTAGES

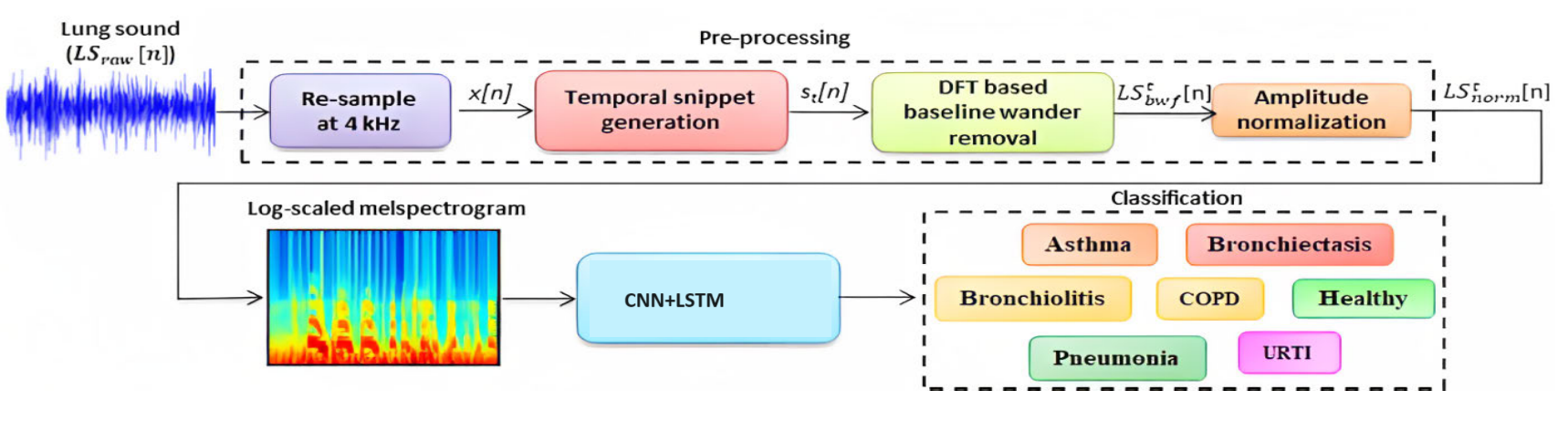
The main advantage of the proposed system are mentioned as followed,

* The proposed system uses less resources.
* The deep learning architecture which is computationally less complex than the conventional feature extraction and classification methods.
* The best results in terms of accuracy from both the datasets were obtained with CNN classification.
* Used in various successful applications in both computer vision and Lung Signal recognition.
* Applications of this system can be in context-aware and intelligent wearable Medical devices, and audio archive management systems.

# CHAPTER 4 SYSTEM DESIGN

* 1. **ARCHITECTURE DIAGRAM**

An architecture diagram is a visual representation of all the elements that make up part, of a system. Diagrams provide a comprehensive overview of a process or system, aiding understanding and facilitating effective collaboration. They show how each component interacts with others and the larger system, enhancing collaboration. Understanding how processes and features interact helps identify weak points and bottlenecks, fostering a joint effort. They are useful in meetings, presentations, and technical documents, making them a valuable tool for understanding and addressing non-static product issues.

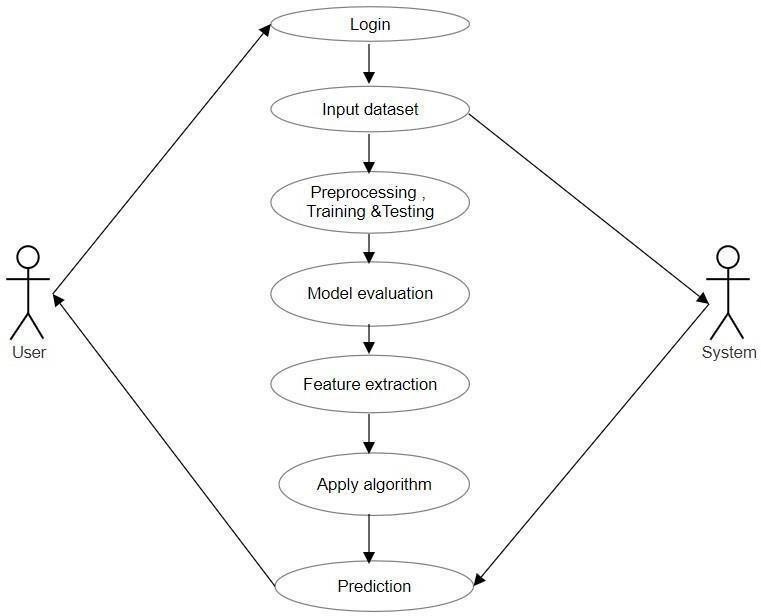


# Fig no 4.1 Architectural diagram

# USECASE DIAGRAM

A use case diagram is a behavioral diagram. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

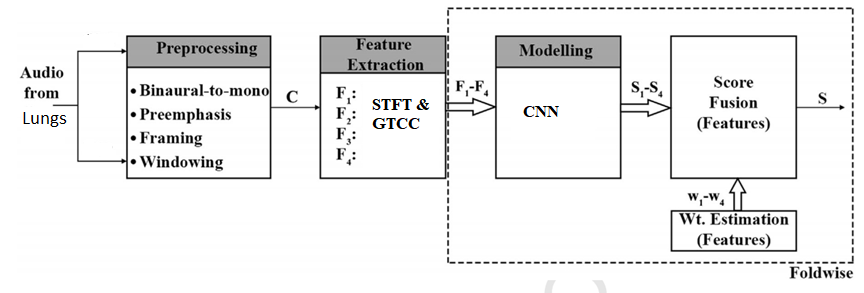
It shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Also represented in terms of actors and their specific goals.



# Fig no 4.2 Use case diagram

# CLASS DIAGRAM

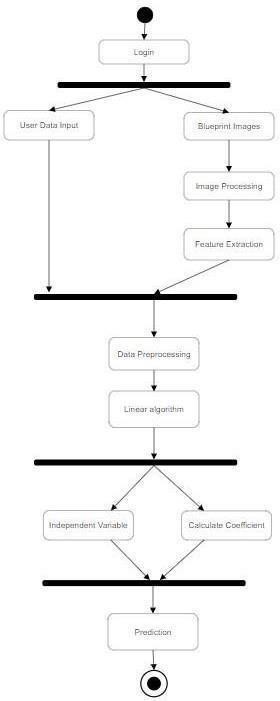
A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class.



# Fig no 4.3 Class diagram

# ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

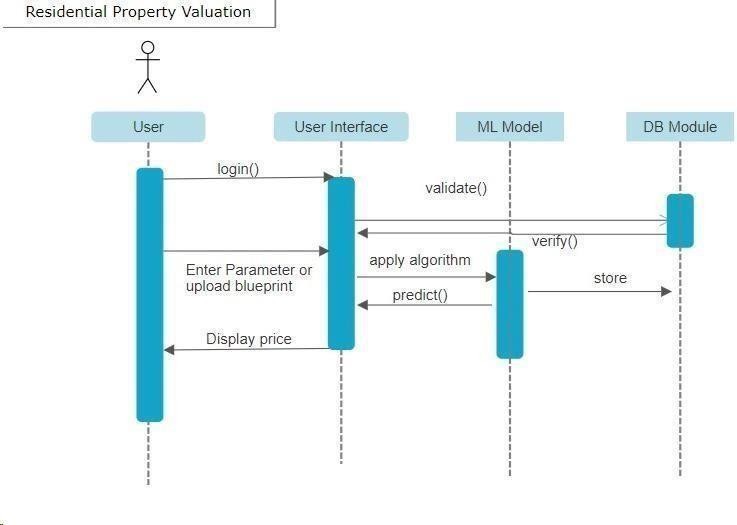


# Fig no 4.4 Activity diagram

# SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is an interaction diagram that shows how processes operate with one another and in what order. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

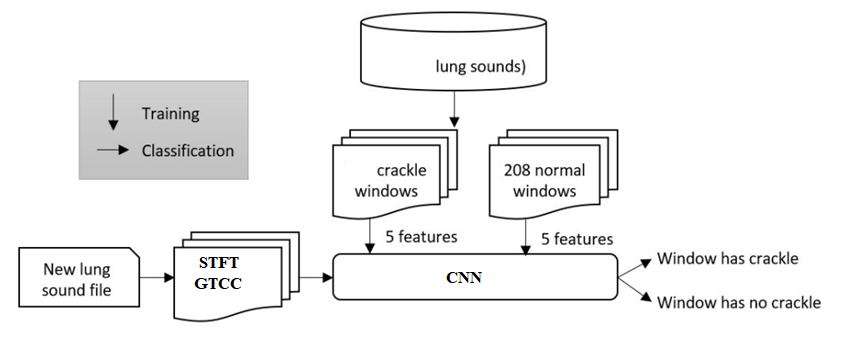
It simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.



# Fig no 4.5 Sequence diagram

# DATA FLOW DIAGRAM

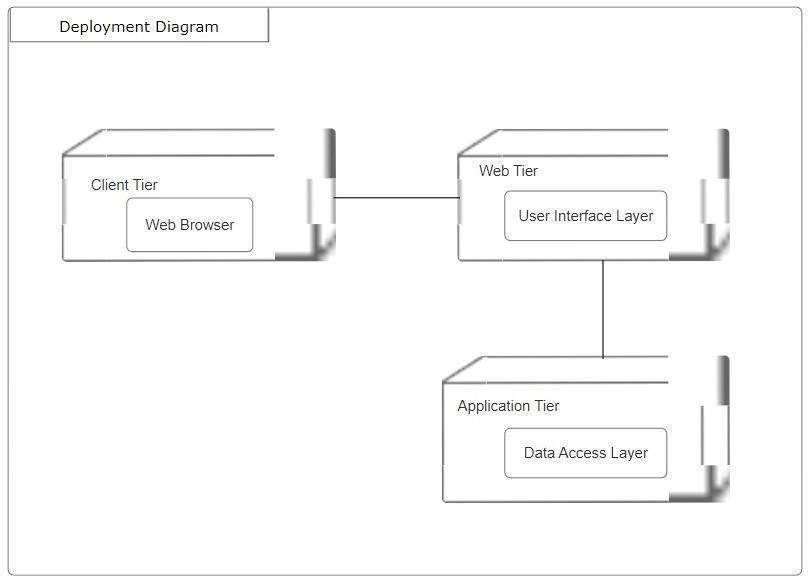
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand- drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.



# Fig no 4.6 Data flow diagram

# DEPLOYMENT DIAGRAM

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components exist, what software components run on each node, and how the different pieces are connected. They capture the hardware that will be used to implement the system and the links between different items of hardware. They model physical hardware elements and the communication paths between them. They can be used to plan the architecture of a system. They are also useful for Document the deployment of software components or nodes



# Fig no 4.7 Deployment diagram

**CHAPTER 5 SYSTEM SPECIFICATIONS**

# SOFTWARE REQUIREMENT

* + - HTML
    - CSS
    - Machine Learning Algorithms
    - Python Language
    - Anaconda IDE

# HTML

HTML, which stands for HyperText Markup Language, is the standard markup language used to create web pages and define the structure and content of documents on the World Wide Web. HTML is not a programming language; instead, it is a markup language that consists of a set of tags and elements used to structure and present content on web pages. Here are some key points to understand about HTML

1. Document Structure: <head><title><body>.
2. Tags and Elements:<p>
3. Attributes: Elements can have attributes that provide additional information about the element. Attributes are usually placed within the opening tag and are used to specify things like the source of an image, the target URL of a link, or the styling of an element.
4. Text Content: HTML is used to structure text content, such as headings, paragraphs, lists (both ordered and unordered), and blockquotes.
5. Hyperlinks:<a>
6. Images and Multimedia: <img>, <video>and <audio>.
7. Semantic Elements: <header>, <nav>, <article><section> and <footer>.

# CSS

CSS, which stands for Cascading Style Sheet, is a stylesheet language

used to describe the presenting and visual design of HTML and XML documents, including webpages. CSS enables web developers to control the layout, formatting, colors, fonts and other visual aspects of a webpage, separating the content (HTML) from its presentation. Here are key aspects and concepts of CSS.

# FLASK

Flask is a micro web framework for python designed to make it easy

to build web applications. It is known as for its simplicity, flexibility and minimalism, providing the essential tools for web development without imposing a specific structure are set of libraries on the developers.here are some key aspects and features of flask .

Flask is a popular choice for web development in python, offering a clean straight forward way to build web applications. Its simplicity and flexibility make it excellent framework for developers who want to have more control over their project structure and component.

# Machine Learning Algorithms

Machine Learning (ML) is a technology which is a subset of artificial intelligence in which the computer machine is trained based on the labeled data set and works according to the trained set of data and the predictions will occur or the algorithm of the machine is trained in the way in which it will learn from the from the examples of the environment in which it is been deployed, makes experience and provides the predictions of that environment, without explicitly programming computers. The two main processes of machine learning algorithms are classification and regression.

# Types of Machine Learning Algorithms

* 1. **Supervised Learning**

This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these sets of variables, we generate a function that maps inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

# Unsupervised Learning

In this algorithm, we do not have any target or outcome variable to predict estimate. It is used for clustering populations in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means.

# Reinforcement Learning

Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process

# Common Machine Learning Algorithms

The list of commonly used machine learning algorithms. These algorithms can be applied to almost any data problem are Linear Regression, Logistic Regression, Decision Tree, SVM, Naive Bayes, KNN, K-Means etc

# Python IDE

An IDE (Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include as

An editor designed to handle code (with, for example, syntax highlighting and auto- completion)

Most IDEs support many different programming languages and contain many more features. They can, therefore, be large and take time to download and install. You may also need advanced knowledge to use them properly.

In contrast, a dedicated code editor can be as simple as a text editor with syntax highlighting and code formatting capabilities. Most good code editors can execute code and control a debugger. The very best ones interact with source control systems as well. Compared to an IDE, a good dedicated code editor is usually smaller and quicker, but often less feature rich.

# HADWARE REQUIREMENTS

|  |  |
| --- | --- |
| **Component** | **Minimum Requirement** |
| Processor | 64-bit, four-core, 2.5 GHz minimum per core |
| RAM | 4 GB |
| Hard disk | 120 GB |

**Table 5.2 Hardware Requirements Processor**

A processor, also known as a central processing unit (CPU), is a crucial component of a computer that executes instructions and performs calculations necessary for a computer to function. It acts as the brain of the system, carrying out tasks like running programs, handling data, and managing input/output operations.

Processors are designed to handle a wide range of tasks, from basic operations like arithmetic and logic calculations to more complex tasks like running software applications and managing system resources. They come in various architectures, speeds, and capabilities, and different types of processors are optimized for different types of tasks.

# RAM

RAM stands for Random Access Memory. It is a type of computer

memory that is used to store data and machine code currently being used and processed by a computer. Unlike storage devices like hard drives or SSDs, which provide long-term storage, RAM provides temporary storage that is quickly accessible to the CPU.

# Hard Disk

A hard disk, often abbreviated as HDD, stands for Hard Disk Drive. It is a non-volatile data storage device that is used for long-term storage of digital data. Unlike RAM (Random Access Memory), which is volatile and loses data when the computer is powered down, the data stored on a hard disk remains intact even when the computer is turned off.

Storage devices like hard disks are needed to install operating systems, programs and additional storage devices, and to save documents. Without devices like HDDs that can retain data after they have been turned off, computer users would not be able to store programs or save files or documents to their computers. This is why every computer needs at least one storage device to permanently hold data as long as it is needed.

Some of the most common storage drive capacities include the following:

* + - 16 GB, 32 GB and 64 GB. This range is among the lowest for HDD storage space and is typically found in older and smaller devices.
    - 120 GB and 256 GB. This range is generally considered an entry point for HDD devices such as laptops or computers.
    - 500 GB, 1 TB and 2 TB. Around 500 GB and above of HDD storage is typically considered decent for an average user. Users can most likely store all their music, photos, videos and other files with this much space. Individuals with games that take up a lot of space should find 1 TB to 2 TB of HDD space suitable.
    - More than 2 TB. Anything over 2 TB of HDD space is suitable for users who work with high-resolution files, who need to store or house a large amount of data, or who want to use that space for backup and redundancy.

**CHAPTER 6**

**SYSTEM MODULE**

# DATA PROCESSING

Typically, to evaluate robustness of algorithms, health professionals detect adventitious respiratory sounds by annotating sounds with the help of Respiratory Sound Annotation Software (SAS). As audio clip contains high deviations across its entire length, its analysis is not trivial. Therefore, each audio clip is broken down into smaller segments called frames to facilitate analysis. In our research, we divided the clips into frames consisting of 256 sample points with a 100-point overlap in between them. The parameters were chosen based on [22]. The same 200 audio clips (as in Figure 3.3) are shown in Figure 4.1 after framing. The number of Sz sized overlapping frames Of with O overlapping points for a signal having S points is presented below:

*Of* = l*S* − *S zO* + 1m. (4.1)

After framing the signal into shorter segments, it was observed that in various in- stances the starting and ending points were not aligned in a frame. These discontinuities/ jitters lead to smearing of power across the frequency spectrum. This posed a problem in the form of spectral leakage during frequency domain analysis which produced ad- ditional frequency components. To tackle this, the frames were subjected to a window function. Hamming window was chosen for this purpose due to its efﬁcacy as demon- strated in [22]. Post framing, jitters might be observed in them which interfere with the Fourier Transformation of the same in the form of spectral leakage. In order to minimize such problems, the frames are usually multiplied with a windowing function which ap- proaches 0 towards its ends and reaches its peak in the middle. Amidst various such windowing functions, Hamming Window function is one of the popularly used window- ing functions. The same frames (Figure 4.1) are presented in Figure 4.2 after windowing. The hamming window is mathematically illustrated below:

*A*(*z*) = 0.54 − 0.46 cos 2π*z*!*S z* − 1

,

(4.2)

where *A*(*z*) is the hamming window function and *z* is a point within a frame.

# Feature extraction

After frame extraction, we performed Linear Predictive Coefﬁcient(LPC) analysis [23] on each of them. A present sample is represented in terms of previous samples. The previous P samples are used to present the rth sample in a signal s() as presented below:

*s*(*r*) ≈ *p*1 *s*(*r* − 1) + *p*2 *s*(*r* − 2) + *p*3 *s*(*r* − 3)+, . . . , +*pPs*(*r* − *P*), (4.3)

where p1, p2,. . . , p*P* are the LPCs or predictors. The error of this prediction *E*(*r*) bounded by the actual and predicted samples: (*s*(*r*) and *s*ˆ(r)) can be explained as

*P*

*E*(*r*) = *s*(*r*) − *s*ˆ(*r*) = *s*(*r*) − X *pk s*(*r* − *k*). (4.4)

*k*=1

The error of sum of squared differences (as shown below) is minimized to generate the unique predictors for a *x* sized frame, which can be expressed as

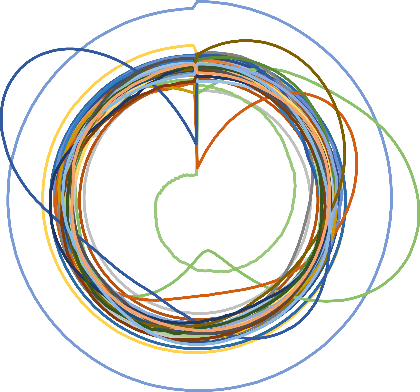
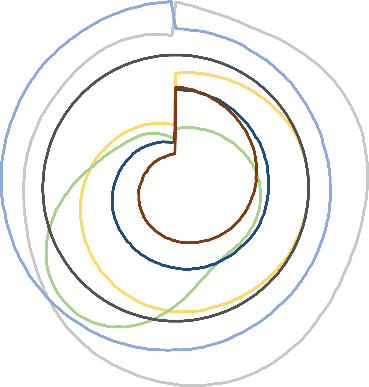
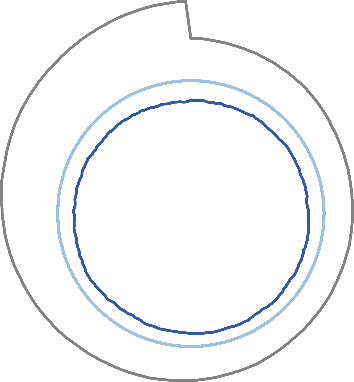
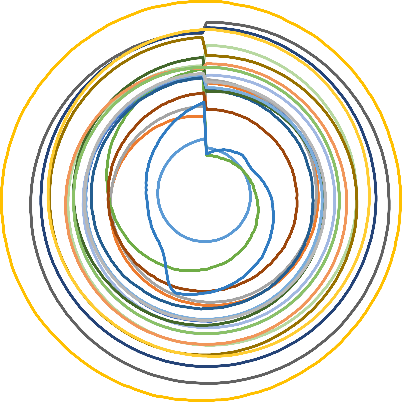
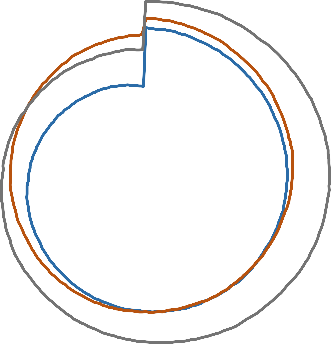
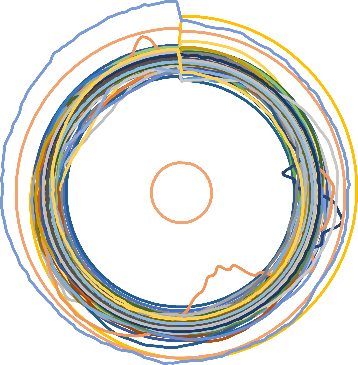
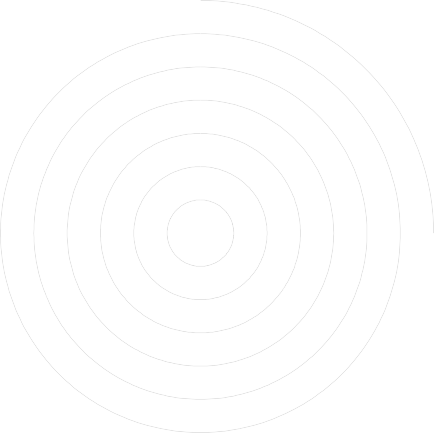
*P*

2

*Er* = Xh*sr*(*x*) − X *pk sr*(*x* − *k*)i . (4.5)

*x k*=1

1



0.8

0.6

0.4

0.2

0

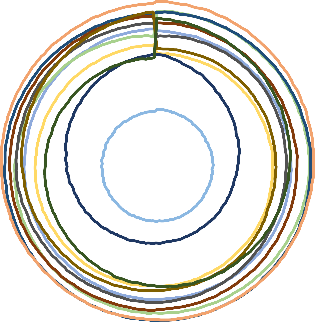
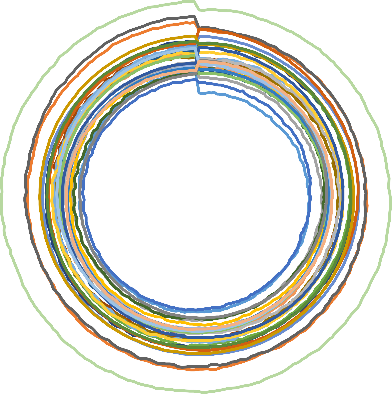
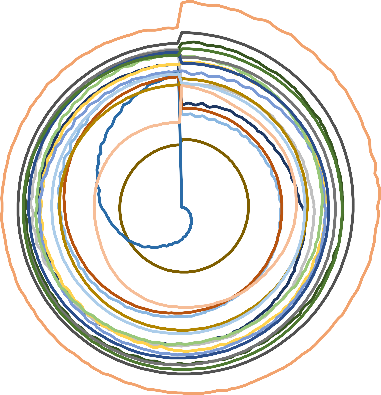
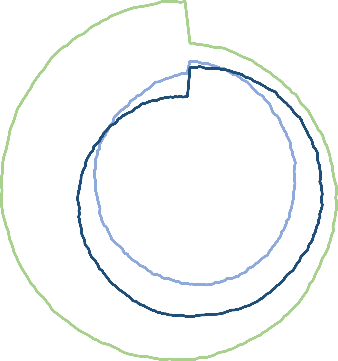
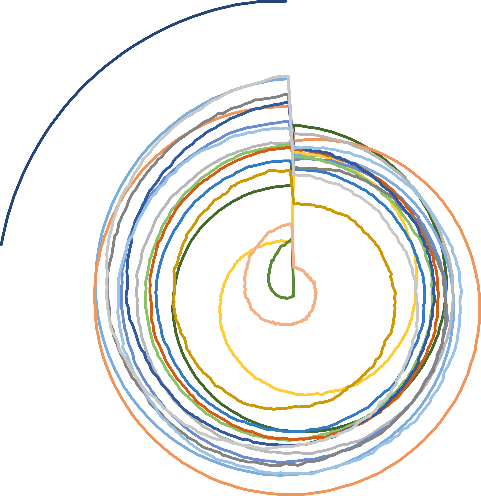
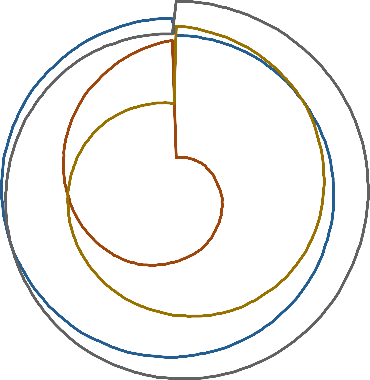
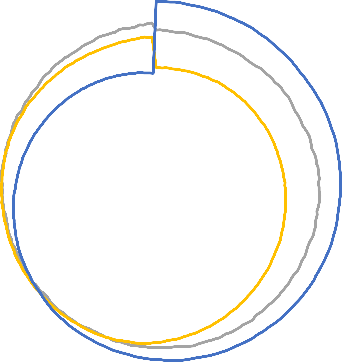
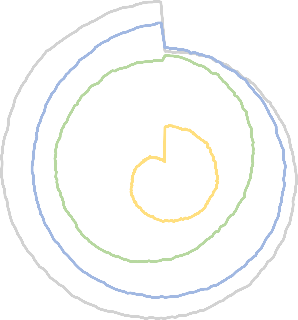
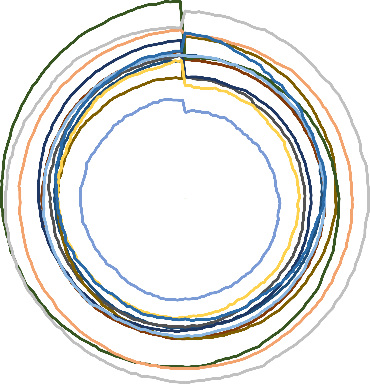
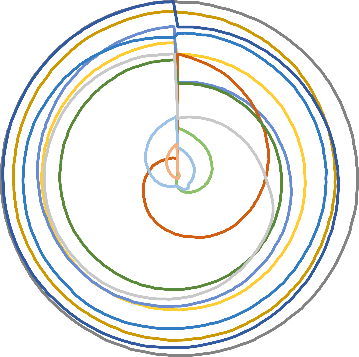
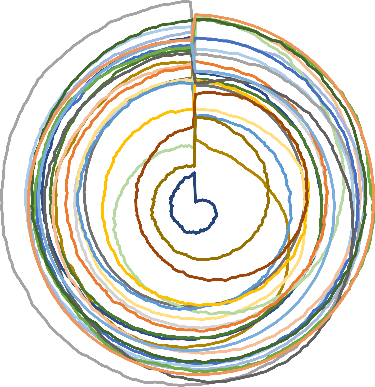
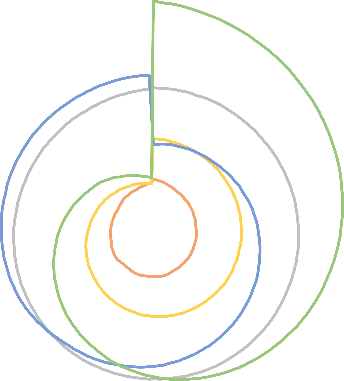
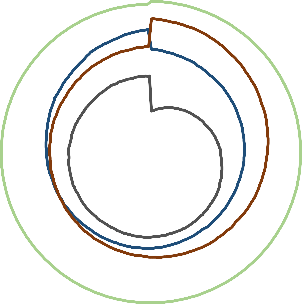
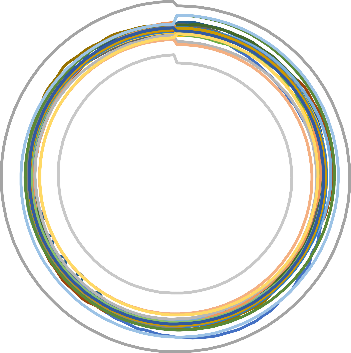
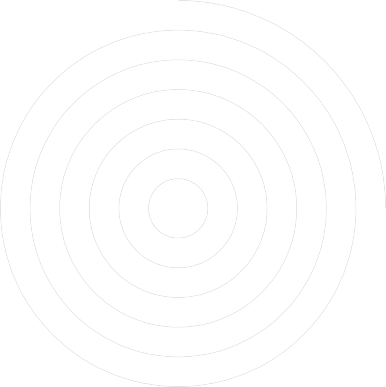
-0.2

-0.4

-0.6

-0.8

1



0.8

0.6

0.4

0.2

0

-0.2

-0.4

-0.6

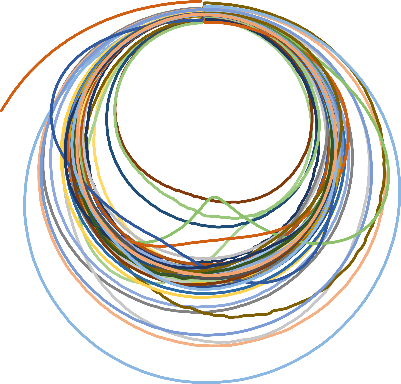
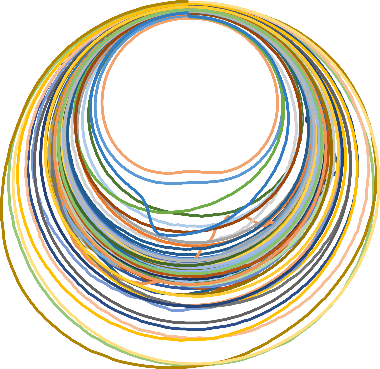
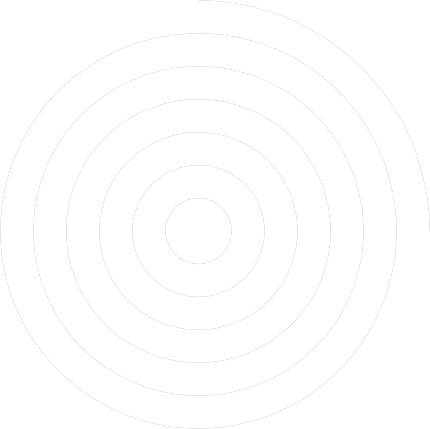
-0.8

-1

Figure 4.1: The same 200 audio clips (as in Fig. 3.3) after framing: healthy class (left) and non-healthy class (right).

1

1



0.8

0.6

0.4

0.2

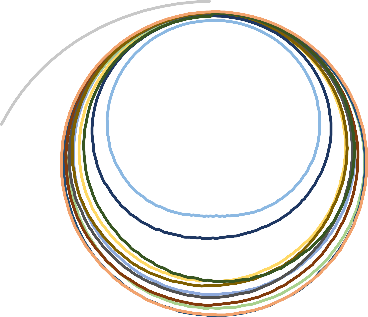
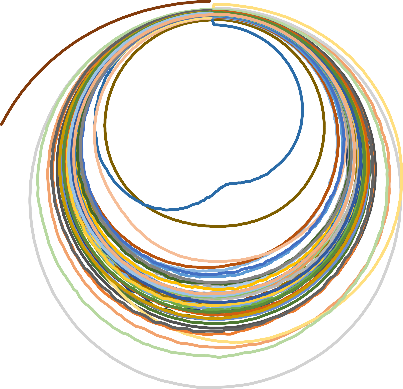
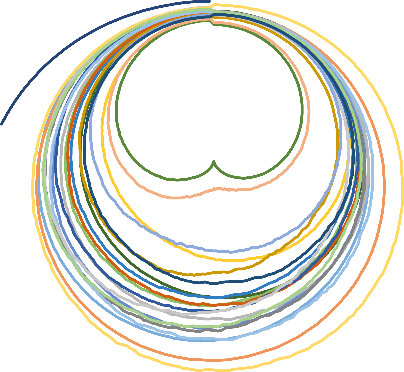
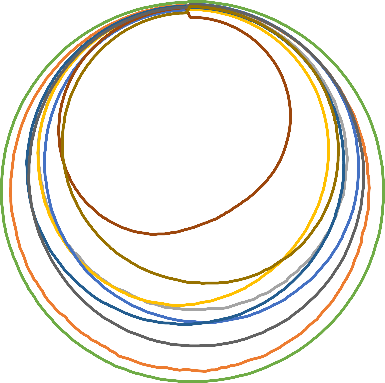
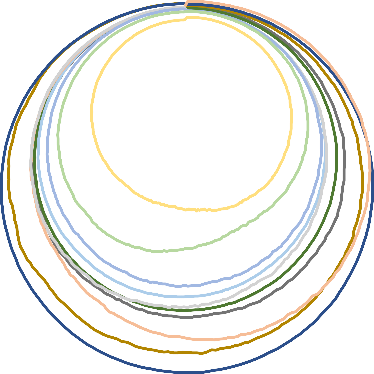
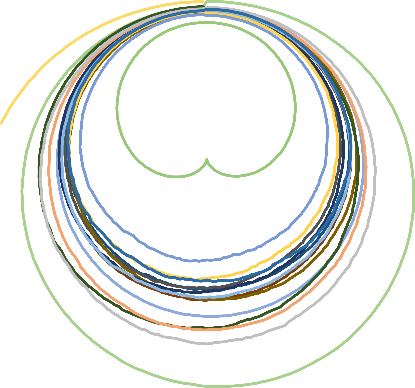
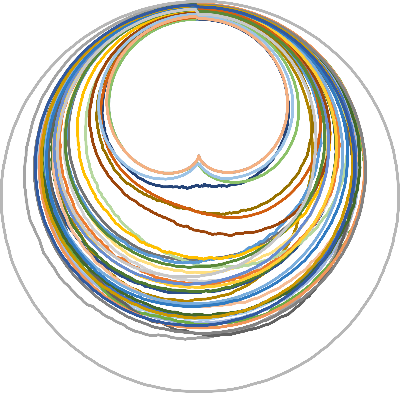
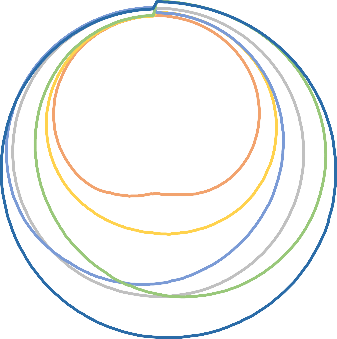
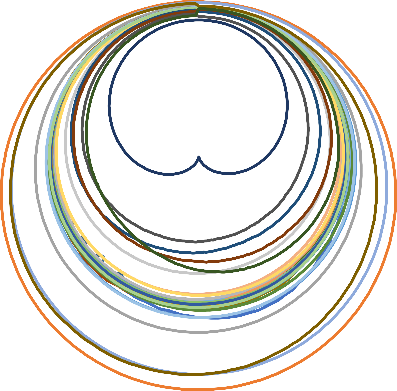
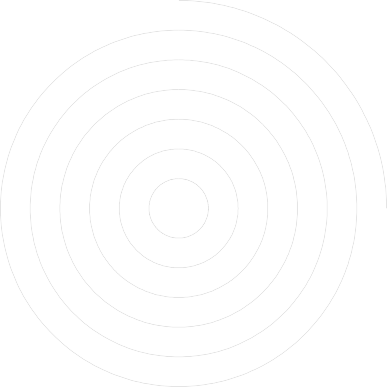
0

-0.2

-0.4

-0.6

-0.8



0.8

0.6

0.4

0.2

0

-0.2

-0.4

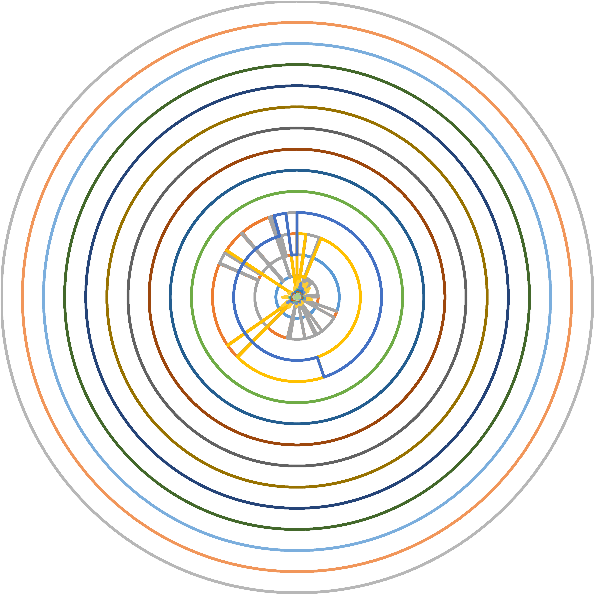
-0.6

-0.8

-1

Figure 4.2: Representation of the same 200 audio clips (as in Fig. 3.3) after windowing: healthy class (left) and non-healthy class (right).

14



12

10

8

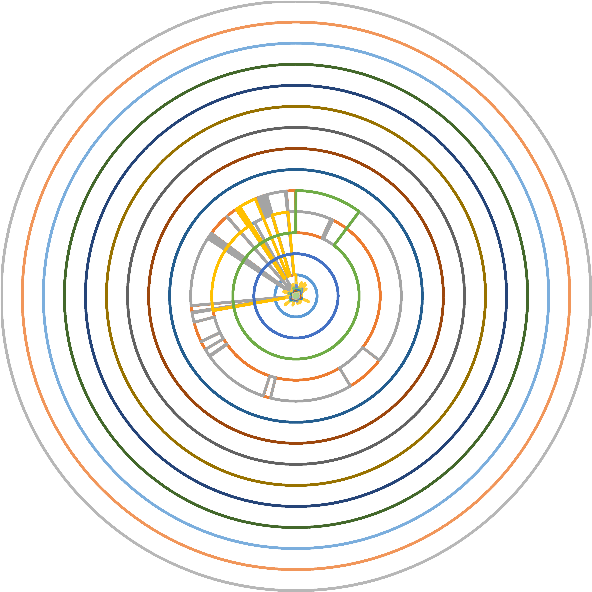
6

4

2

0

14



12

10

8

6

4

2

0

Figure 4.3: Representation of 30 dimensional features for the audio clips: healthy class (left); non-healthy class (right).

Thereafter, a recursive technique is used to compute the Cepstral coefﬁcients (*C*), which is expressed as

*C*0 = log*e P*

C*r* = *pr* + P*r*−1 *qrCq pr*−*q*, *f or* 1 < *r* ≤ *P and*

*q*=1

*q*=*r*−*P qrCq pr*−*q*, *f or r* > *P* (4.6)

C*r* = P*r*−1

Since clips in the dataset were of unequal lengths and number of frames obtained varied. When features were extracted in frame level, it produced different dimensions. To handle this, we performed two operations: a) grading and b) standard deviation measurement.

1. Firstly, the sum of GTCC coefﬁcients in each of the frequency ranges (bands) across all the frames was computed. Based on the sum of these energy values, bands were graded in an ascending order. This sequence of band numbers was used as features that helped in identifying dominance of different bands for the clips from various categories.
2. Secondly, standard deviation was computed for every band. These two metrics were stacked to form the feature, which is independent of the clip length. 10, 20, 30, 40 and 50 dimensional features were extracted for the 2 classes. The trend of the 30 dimensional feature values (best result) for the 2 classes is shown in Fig. 4.3

# Classiﬁcation

# 6.3.1 Multi-layer perceptron (MLP)

Multilayer perceptron’s (MLPs) otherwise called as Feedforward neural networks (FNNs) are the archetypes of deep learning models. These networks were inspired by neuro- science and how we believe neurons work in the brain.

The purpose of these networks is to approximate some function f by mapping an input domain to an output domain, which can be applied to solving complex problems such as prediction or classiﬁcation from high dimensional data to a set of labels.

These networks consist of multiple layers, where the ﬁrst layer is the input layer and the last is the output layer. The intermediate layers in the network are called the hidden layers and their number can vary. The use of multiple layers is what originated the term “Deep Learning”, with each additional layer creating an additional level of abstraction or representation.

Each layer is comprised of a number of neurons that represent activation values, and it determines the width of that layer. Each neuron has a number of input weights that connect to each of the neurons of the previous layer, with the exception of the neurons in the input layer.

The activation values of the input layer are propagated forward in the direction of the output layer with no feedback connections where the outputs of the neurons are fed to previously activated neurons, hence the designation of “feedforward”.

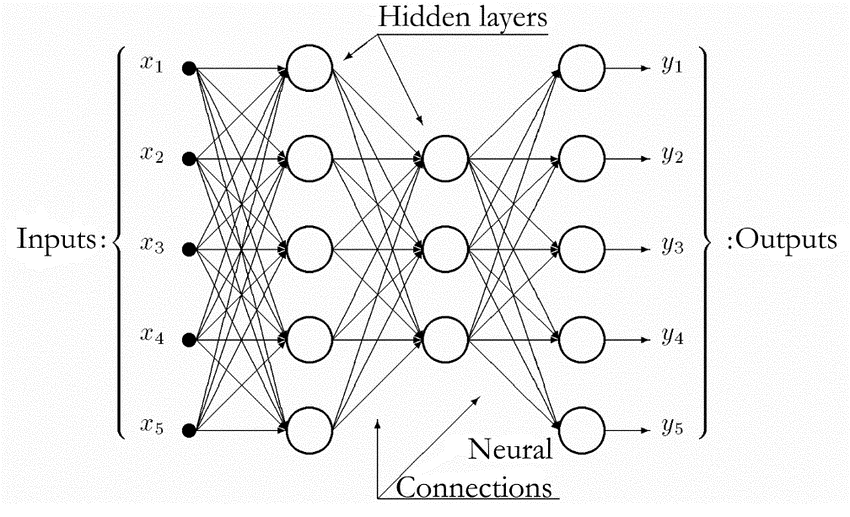


Figure 4.4: Structure of a feed-forward ANN with two hidden layers

The network is associated with a directed acyclic weighted graph describing how the

functions are composed together. The network’s parameters consist of the weights and biases between layers. The output activation values of a layer are represented as a vector, with each entry of the vector representing the activation value of a single neuron. The size of the vector corresponds to the number of neurons in that layer.

The weights between layers are represented as a 2D matrix, with each entry of the matrix at coordinates i,j representing the weight connecting the neuron i from layer l - 1 to the neuron j in the layer l. The biases between layers are represented as a vector with the same size as the number of neurons in the next layer.

The mathematical equation for the calculation of the output of each layer of the feed- forward model is deﬁned as:

* hl = gl (Wlhl - 1 + bl), the activation values of a layer. With Wlhl being the dot product operation between the weight matrix of the current layer and the output values of the previous layer.
* y = hL - 1, the activation values of the ﬁnal output layer of the network

**6.3.2 Layers**

We employed MLP classiﬁer, feed-forward artiﬁcial neural network – for classiﬁcation purpose [57]. Feedforward neural networks are made up of the input layer, output layer and hidden layer. It is a supervised learning algorithm trained on a dataset using a func- tion f() : Zn → Zo, where n and o represent the dimensions for input and output. For a given set of features P = p1, p2, ..., pn and aim x, a non- linear function is learned for classiﬁcation. The difference between MLP and logistic regression lies in the existence of one or more non-linear layers (hidden layers) between the input and the out- put layer. MLP consists of three or more layers (input layer, output layer and one or more hidden layers) of non-linear activating neurons. The number of hidden layers can be increased according to the requirement of developing a model to accomplish certain task. The ini- tial layer is the input layer which comprises of a set of neurons p*i*|*p*1, *p*2, ..., *pn* denoting the features. Each neuron of the hidden layer modiﬁes the values from the previous layer using sum of weights as: w1 *p*1 + *w*2 *p*2+, ..., +*w*2 *pn*.

The activation function that represents the relationship between input and output layer in of non-linear nature. It makes the model ﬂexible in deﬁning unpredictable relationships. The activation function can be expressed as:

*yi* = tanh(*wi*) and *yi* = (1 + *ewi* )−1, (4.7)

where yi and wi denotes the outcome of the ith neuron and weighted sum of the input features. The values from the ultimate hidden layer are provided to the output layer as output values. Each layer of MLP contains several fully connected layers as each neuron in a layer is attached to all the neurons of the previous layer. The parameters of each neuron are independent of the remaining neurons of the layer ensuring possession of unique set of weights. The initial momentum and learning rate were set to 0.2 and 0.3 respectively, function is learned for classiﬁcation.

The difference between MLP and logistic regression lies in the existence of one or more non-linear layers (hidden layers) between the input and the out- put layer. MLP consists of three or more layers (input layer, output layer and one or more hidden layers) of non-linear activating neurons. The number of hidden layers can be increased according to the requirement of developing a model to accomplish certain task. The ini-tial layer is the input layer which comprises of a set of neurons p*i*|*p*1, *p*2, ..., *pn* denoting the features. Each neuron of the hidden layer modiﬁes the values from the previous layer using sum of weights as: w1 *p*1 + *w*2 *p*2+, ..., +*w*2 *pn*.

The activation function that represents the relationship between input and output layer in of non-linear nature. It makes the model ﬂexible in deﬁning unpredictable relationships. The activation function can be expressed as:

*yi* = tanh(*wi*) and *yi* = (1 + *ewi* )−1, (4.7)

where yi and wi denotes the outcome of the ith neuron and weighted sum of the input features. The values from the ultimate hidden layer are provided to the output layer as output values. Each layer of MLP contains several fully connected layers as each neuron in a layer is attached to all the neurons of the previous layer. The parameters of each neuron are independent of the remaining neurons of the layer ensuring possession of unique set of weights. The initial momentum and learning rate were set to 0.2 and 0.3 respectively.

# CHAPTER 7 CONCLUSION AND FUTURE WORK

# CONCLUSION

Looking at the audio content, it is difﬁcult to classify respiratory sounds. In our research, a system is presented for distinction of healthy and non-healthy lung sounds which is very important prior to further diagnosis of the type and severity of infection. We have performed our experiments using a publicly available dataset and evaluations indicate that the highest accuracy of 99.22% with an AUC value of 0.9993 is obtained.We have presented a machine learning based approach for detecting crackles in sounds recorded using a stethoscope as part of a large health survey. We evaluated several feature extraction methods, and classifiers using many sound recordings. A simple 5-dimenstional layer and a CNN-LSTM Kernel performed best. The low-dimensional feature vector makes the CNN-LSTM very fast and it can classify lung sounds in real-time.

# FUTURE WORK

Automated adventitious sounds detection or classiﬁcation provides a promising solution to overcome the limitations of conventional auscultation. In future the subject area for future investigation will be:

1. To use larger dataset and test further on robustness in presence of higher percentages of noise.
2. Attempts will also be made towards isolation of breath sounds from the ambient noises and heart- beat sounds [58] for better analysis.
3. Other acoustic techniques [59] will be applied for even better modelling of the lung sounds along with deep learning-based approaches.
4. To have clinical use in pulmonary health screening and as a tool in differential diagnosis of pulmonary diseases.
5. Finally, we will be trying to identify the nature and severity of infection from the breath sounds.

# CHAPTER 8

# DATASET DESCRIPTION

The lung sounds that are heard over the chest wall are caused by the airﬂow in the lungs during the inspiration and expiration phases. These sounds are non-stationary and non-linear signals, which make it difﬁcult for physicians to recognize any abnormalities [13]. The types and characteristics of lung sounds are listed in Fig. 3.1 [49, 50, 51, 52, 53, 54, 55, 56]. Abnormal breath sounds include the absence or reduced intensity of sounds where they should be heard or, by contrast, the presence of sounds where there should be none, as well as the presence of adventitious sounds. As opposed to those classiﬁed as “normal”, abnormal sounds are those which may indicate a lung problem, such as inﬂammation or an obstruction. Each lung disorder is associated with one or more lung sounds [13]. The disorders that are associated with each sound are also detailed in Fig. 3.1. The dominant frequency of heart sounds is typically below 150Hz, whereas the dominant frequency of lung sounds ranges between 150 and 2000Hz. This difference in the frequencies makes it easier to ﬁlter the heart sounds from the lung sounds. The durations of the different types of lung sounds are also mentioned in Fig. 3.1.



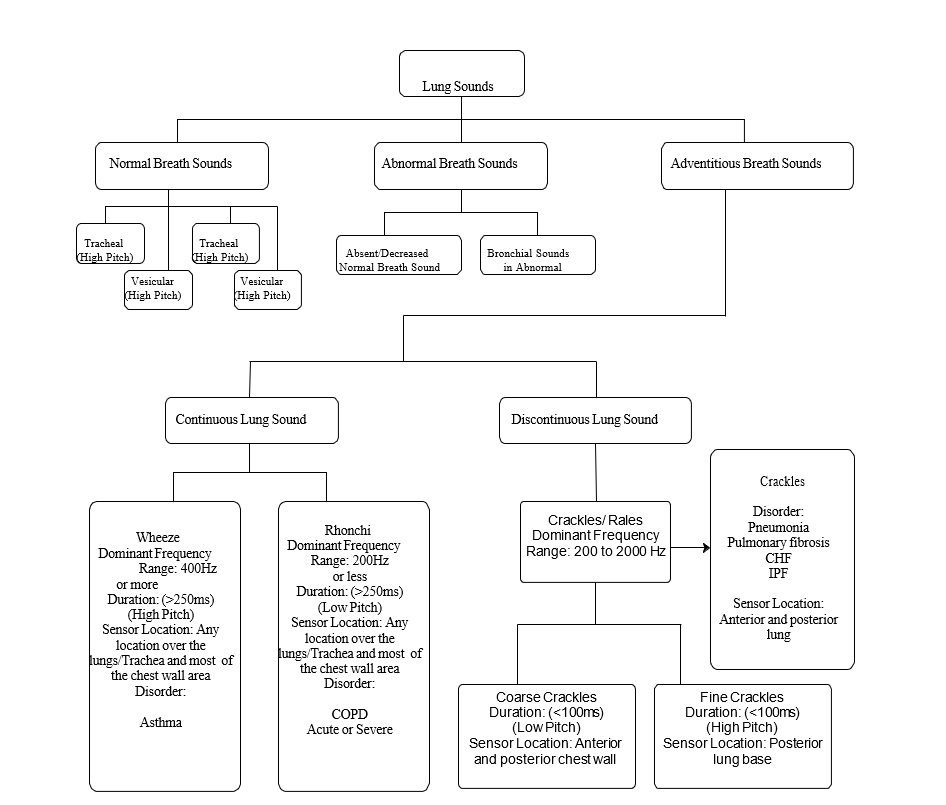


Figure: 3.1 : Lung sound classification

The ICBHI (International Conference on Biomedical and Health Informatics) dataset [24] was originally compiled to support the scientiﬁc challenge on respiratory data analysis organized in conjunction with the 2017 International Conference on Biomedical Health Informatics (ICBHI). The current version of this database is made freely available for research which contains both the public and the private dataset of the ICBHI challenge. The Respiratory Sound Database contains audio samples that were collected independently by two research teams in two different countries (Greece and Portugal) over several years.

The data collection required several years, and the ﬁnal dataset consists of 920 labeled audio tracks from 126 distinct participants. It is currently the largest annotated, publicly available dataset.

The two independent research groups are

1. Respiratory Research and Rehabilitation Laboratory (Lab3R), School of Health Sciences, University of Aveiro, Aveiro, Portugal and
2. Papanikolaou General Hospital and the General Hospital of Imathia, Aristotle University of Thessaloniki and the University of Coimbra, Thessaloniki, Greece.

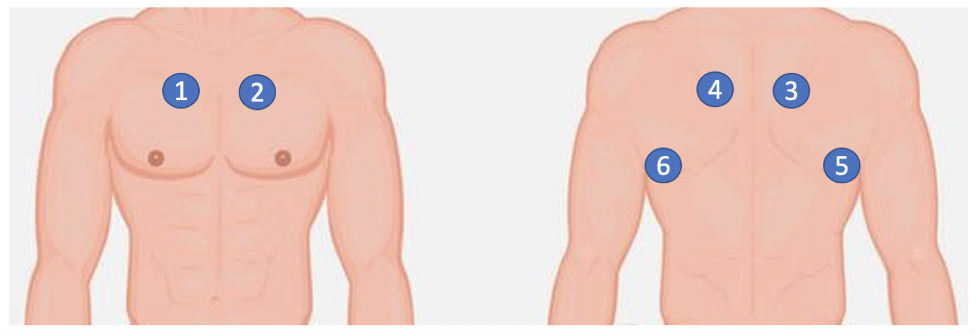


Figure 3.2: Locations from which respiratory sounds were collected: right anterior (1), left anterior (2), right posterior (3), left posterior (4), right lateral (5) and left lateral (6).

These audio signals were recorded using one of the following stethoscope systems:

1. Electronic Stethoscope 3200, 3M Littmann,
2. Classic II SE Stethoscope, 3M Littmann
3. C417 L Professional Lavalier Microphone, AKG HARMAN, and
4. Meditron Master Elite Electronic Stethoscope, Welch Allyn. The sounds were collected from six different positions (left/right anterior, posterior and lateral) as illustrated in Figure 3.2.

The audios were collected in both clinical and non-clinical settings from adult participants of disparate ages. Participants encompassed patients with lower and upper respiratory tract infections, pneumonia, bronchiolitis, COPD, asthma, bronchiectasis, and cystic ﬁbrosis..

# APPENDICES

# SOURCE CODE

# Login.html

<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>Login</title>

<link rel="stylesheet" type="text/css" href="./static/css/login.css">

</head>

<body>

<div id="bg"></div>

<form>

<div class="form-field">

<input type="email" placeholder="Email / Username" required/>

</div>

<div class="form-field">

<input type="password"placeholder="Password"required/>

</div>

<div class="form-field">

<button class="btn" type="submit" onClick="location.href='register.html'">Log in</button>

</div>

</form>

</body>

</html>

# Register.html

<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>Reigster</title>

<link rel="stylesheet" type="text/css" href="./static/css/register.css">

</head>

<body>

<div id="bg"></div>

<form>

<div class="form-field">

<input type="text" placeholder="Name" required/>

</div>

<div class="form-field">

<input type="email" placeholder="Email / Username" required/>

</div>

<div class="form-field">

<input type="password" placeholder="Password" required/>

</div>

<div class="form-field">

<input type="password" placeholder="retype-Password" required/>

</div>

<div class="form-field">

<button class="btn"type="submit"onClick="location.href='home.html'"> submit</button>

</div>

</form>

</body>

</html>

# Home.html

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<link rel="stylesheet" type="text/css" href="./static/css/home.css">

<head>

<body>

<form>

<div id="bg"></div>

<div class="form-field">

<p >Select the option</p>

<button class="btn" >

<a href='blueprint\_upload.html'>BLUEPRINT</a></button>

<button class="btn">

<a href='specification.html'>SPECIFICATION</a></button>

</div>

</form>

</body>

<html>

# Blueprint\_upload.html

<!DOCTYPE html>

<html>

<head>

<title>Image Upload</title>

<link rel="stylesheet" type="text/css" href="./static/css/home.css">

</head>

<body>

<form method="POST" action="/upload" enctype="multipart/form-data">

<div id="bg"></div>

<div class="form-field">

<p>upload the Blueprint</p>

<input type="file" name="file">

<a href='uploads'><button class="btn" value="Upload">Upload</button></a>

</div>

</form>

</body>

</html>

**Blueprint\_specification.html**

<!DOCTYPE html>

<html>

<head>

<title>RPV</title>

<link rel="stylesheet" type="text/css" href="./static/css/specification.css">

</head>

<body>

<div id="bg"></div>

<form method="POST" action="/predict-by-blueprint">

<div class="form-field">

<input type="text" name="no\_of\_bathrooms" placeholder="no\_of\_bathrooms" required><br>

</div>

<div class="form-field">

<input type="text" name="no\_of\_floors" placeholder="no\_of\_floors" required><br>

</div>

<div class="form-field">

<input type="text" name="ATM" placeholder="ATM yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="school" placeholder="school yes=1/no=0 " required><br>

</div>

<div class="form-field">

<input type="text" name="security" placeholder="security yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="carparking" placeholder="carparking yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="hospital" placeholder="hospital yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="location" placeholder="location" required>

</div>

<button class="btn" type="submit" value="Predict">Predict</button>

</form>

</body>

</html>

# Specification.html

<!DOCTYPE html>

<html>

<head>

<title>RPV</title>

<link rel="stylesheet" type="text/css" href="./static/css/specification.css">

</head>

<body>

<div id="bg"></div>

<form method="POST" action="/predict">

<div class="form-field">

<input type="text" name="area" placeholder="square feet" required>

</div>

<div class="form-field">

<input type="text" name="no\_of\_bedrooms" placeholder="no\_of\_bedrooms" required><br>

</div>

<div class="form-field">

<input type="text" name="no\_of\_bathrooms" placeholder="no\_of\_bathrooms" required><br>

</div>

<div class="form-field">

<input type="text" name="no\_of\_floors" placeholder="no\_of\_floors"required><br>

</div>

<div class="form-field">

<input type="text" name="ATM" placeholder="ATM yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="school" placeholder="school yes=1/no=0 " required><br>

</div>

<div class="form-field">

<input type="text" name="security" placeholder="security yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="carparking" placeholder="carparking yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="hospital" placeholder="hospital yes=1/no=0" required><br>

</div>

<div class="form-field">

<input type="text" name="location" placeholder="location" required>

</div>

<button class="btn" type="submit" value="Predict">Predict</button>

</form>

</body>

</html>

# Result.html

<!DOCTYPE html>

<html>

<head>

<title>RPV</title>

<link rel="stylesheet" type="text/css" href="./static/css/home.css">

</head>

<body>

<form>

<div id="bg"></div>

<div class="form-field">

<p >Residential Property Valuation</p>

<p>Predicted Price: Rs.{{ prediction }}</p>

</div>

</form>

</body>

</html>

# Image\_processing.py

import re

from PIL import Image import pytesseract

def image\_word():

# Path to your Tesseract executable (you should have Tesseract installed) pytesseract.pytesseract.tesseract\_cmd = 'C:/Program Files/Tesseract- OCR/tesseract.exe'

# Open the image using PIL (Pillow)

image = Image.open('./uploads/blueprint1.jpg') # Perform OCR to extract text from the image text = pytesseract.image\_to\_string(image)

# Define a regular expression pattern to extract numeric parts of keywords (e.g., "3 BHK" or "4 BHK")

numeric\_pattern = r'(\d+)\s\*BHK'

# Use a regular expression to find and extract numeric parts of keywords numeric\_parts = re.findall(numeric\_pattern, text, re.IGNORECASE)

# Print the numeric parts detected in the text print(f"Numeric parts detected: {', '.join(numeric\_parts)}")

# Regular expression pattern to match area values in the "40'-0"X 40'-0"" format area\_pattern = r'(\d+\'-\d{1,2}"\s\*X\s\*\d+\'-\d{1,2}")'

# Find and extract area values from the text

areas = re.findall(area\_pattern, text, re.IGNORECASE) # Function to convert area values to square feet

def convert\_area\_to\_square\_feet(area\_value):

# Extract the width and height using regex

match = re.match(r'(\d+)\'-(\d{1,2})"\s\*X\s\*(\d+)\'-(\d{1,2})"', area\_value) if match:

width\_feet = int(match.group(1)) width\_inches = int(match.group(2)) height\_feet = int(match.group(3)) height\_inches = int(match.group(4)) total\_area = (width\_feet ) \* (height\_feet ) return total\_area

else:

return 0

# Process each area value in the list and convert to square feet converted\_areas = [convert\_area\_to\_square\_feet(area\_value) for area\_value in areas]

return numeric\_parts, converted\_areas if name == " main ":

numeric\_parts, converted\_areas = image\_word() print("Numeric Parts:", numeric\_parts) print("Converted Areas:", converted\_areas)

# Routes.py

from flask import Flask, request, redirect, url\_for, render\_template, flash import os

from geopy.geocoders import Nominatim from werkzeug.utils import secure\_filename from image\_processing import image\_word import joblib

app = Flask( name )

app.secret\_key = 'your\_secret\_key' # Change this to a secret key for secure session management

app.config['UPLOAD\_FOLDER'] = 'uploads' app.config['ALLOWED\_EXTENSIONS'] = {'png', 'jpg', 'jpeg', 'gif'} def allowed\_file(filename):

return '.' in filename and filename.rsplit('.', 1)[1].lower() in app.config['ALLOWED\_EXTENSIONS']

# Load the pre-trained machine learning model model = joblib.load('pred3model (4).pkl') @app.route('/')

def login():

return render\_template('login.html') @app.route('/register.html')

def register():

return render\_template('register.html') @app.route('/home.html')

def home1():

return render\_template('home.html') @app.route('/blueprint\_upload.html') def upload\_html():

return render\_template('blueprint\_upload.html') @app.route('/uploads')

def render():

return render\_template('blueprint\_specification.html') @app.route('/upload', methods=['POST'])

def upload():

if 'file' not in request.files:

flash('No file part')

return redirect(request.url) file = request.files['file']

if file.filename == '':

flash('No selected file') return redirect(request.url)

if file and allowed\_file(file.filename):

filename = secure\_filename("blueprint1.jpg") # Rename the uploaded file to "blueprint1.jpg"

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename)) flash('File successfully uploaded as blueprint1.jpg')

return render\_template('blueprint\_specification.html')else: flash('Invalid file format. Allowed formats are: png, jpg, jpeg, gif') return redirect(request.url)

@app.route('/specification.html') def home():

return render\_template('specification.html') @app.route('/predict', methods=['POST']) def predict():

if request.method == 'POST': area = float(request.form['area'])

bedrooms = int(request.form['no\_of\_bedrooms']) bathrooms = int(request.form['no\_of\_bathrooms']) floors = int(request.form['no\_of\_floors'])

ATM = int(request.form['ATM']) school = int(request.form['school']) security = int(request.form['security'])

carparking = int(request.form['carparking']) hospital = int(request.form['hospital']) location = request.form['location']

geolocator = Nominatim(user\_agent="geocoder") location = geolocator.geocode(location)

if location is not None:

latitude = location.latitude longitude = location.longitude

feature\_vector = [area, bedrooms, bathrooms, floors, ATM, hospital, school,carparking, security, latitude, longitude]

prediction = model.predict([feature\_vector])

return render\_template('result.html', prediction=int(prediction[0])) @app.route('/predict-by-blueprint', methods=['POST'])

def predict\_by\_blueprint():

if request.method == 'POST':

original\_list = image\_word() print(original\_list)

area\_bedrooms = [[int(item) for item in sub\_list] for sub\_list in original\_list] area = area\_bedrooms[0][0]

bedrooms = area\_bedrooms[1][0]

bathrooms = int(request.form['no\_of\_bathrooms']) floors = int(request.form['no\_of\_floors'])

ATM = int(request.form['ATM'])

school = int(request.form['school']) security = int(request.form['security']) carparking = int(request.form['carparking']) hospital = int(request.form['hospital']) location = request.form['location']

geolocator = Nominatim(user\_agent="geocoder") location = geolocator.geocode(location)

if location is not None:

latitude = location.latitude longitude = location.longitude

feature\_vector = [area, bedrooms, bathrooms, floors, ATM, hospital, school,carparking, security, latitude, longitude]

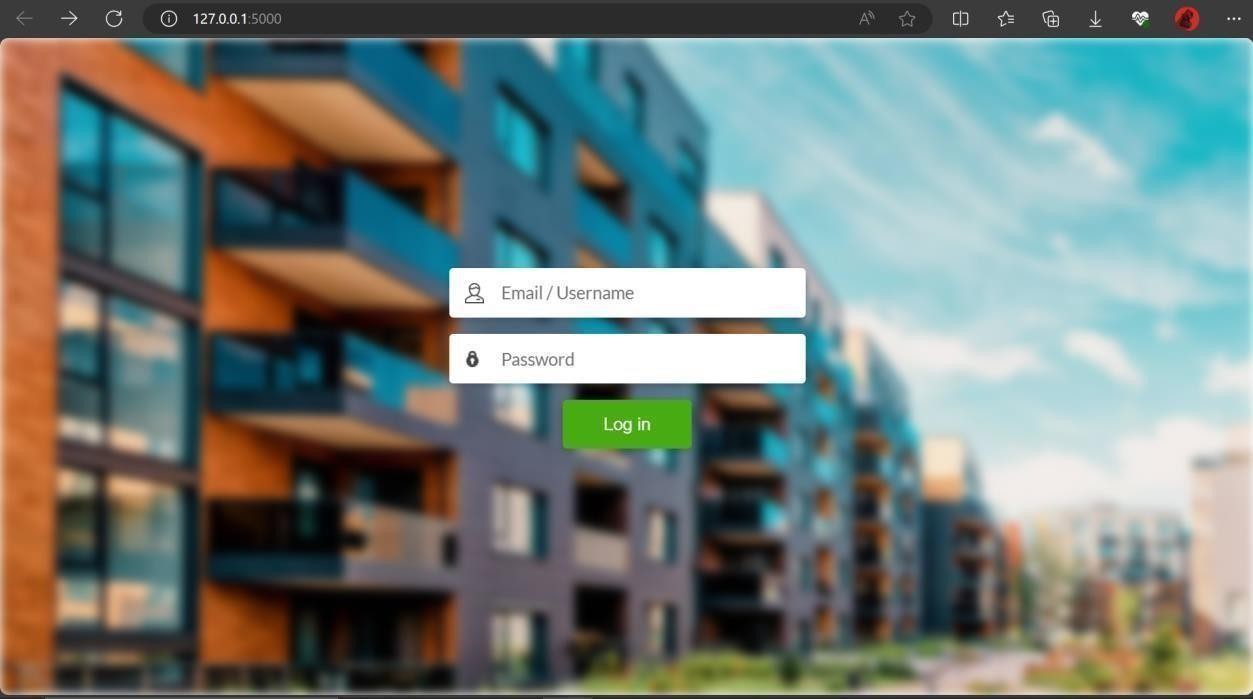
prediction = model.predict([feature\_vector])

return render\_template('result.html', prediction=int(prediction[0]/100)) if name == ' main ':

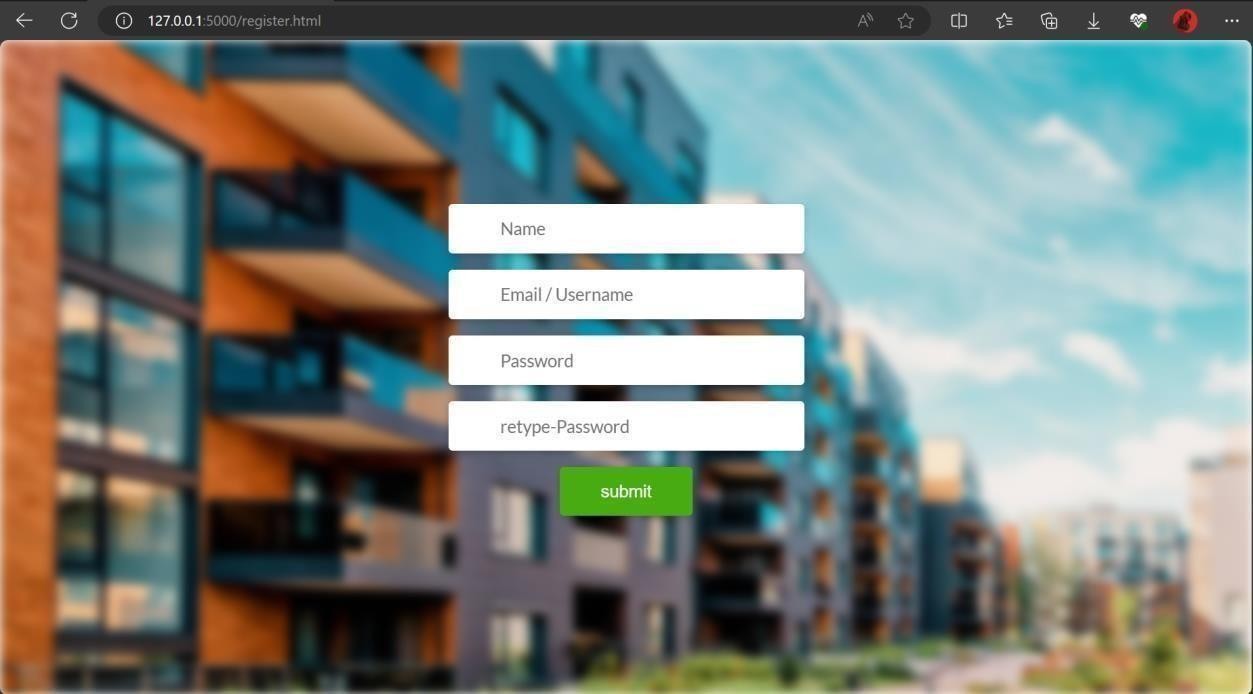
app.run(debug=True)

# SCREENSHOTS

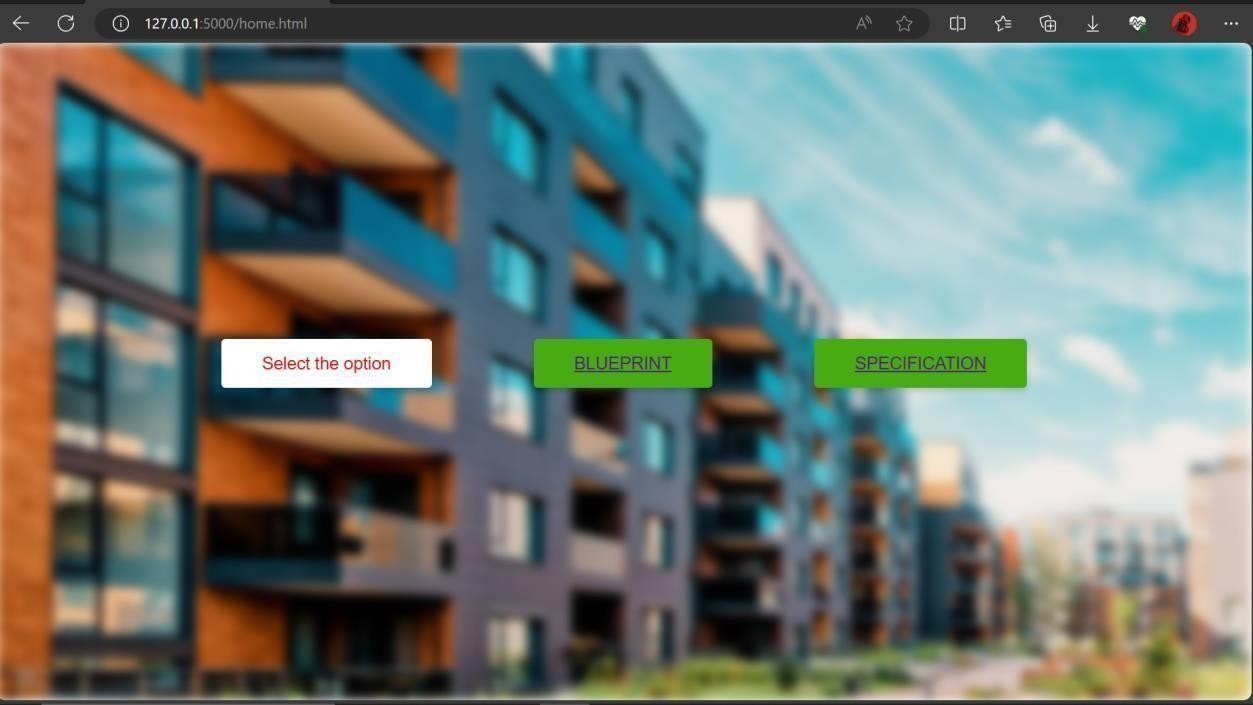
**Login page**



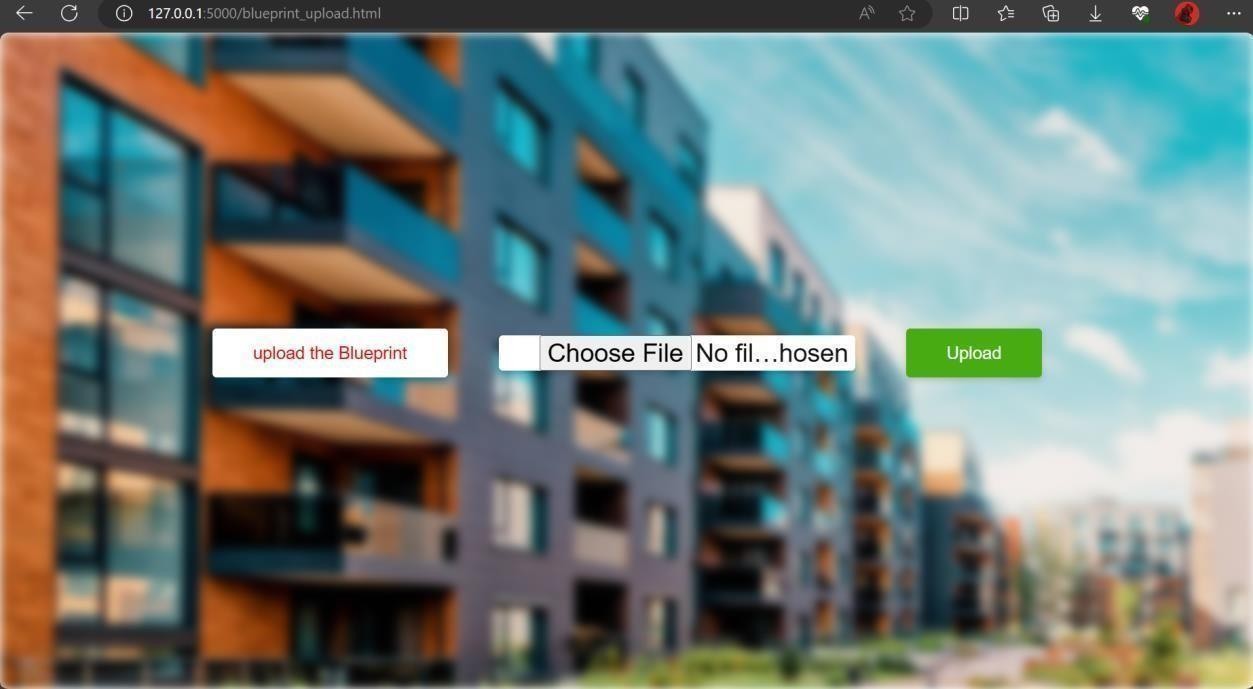
**Registration Page**



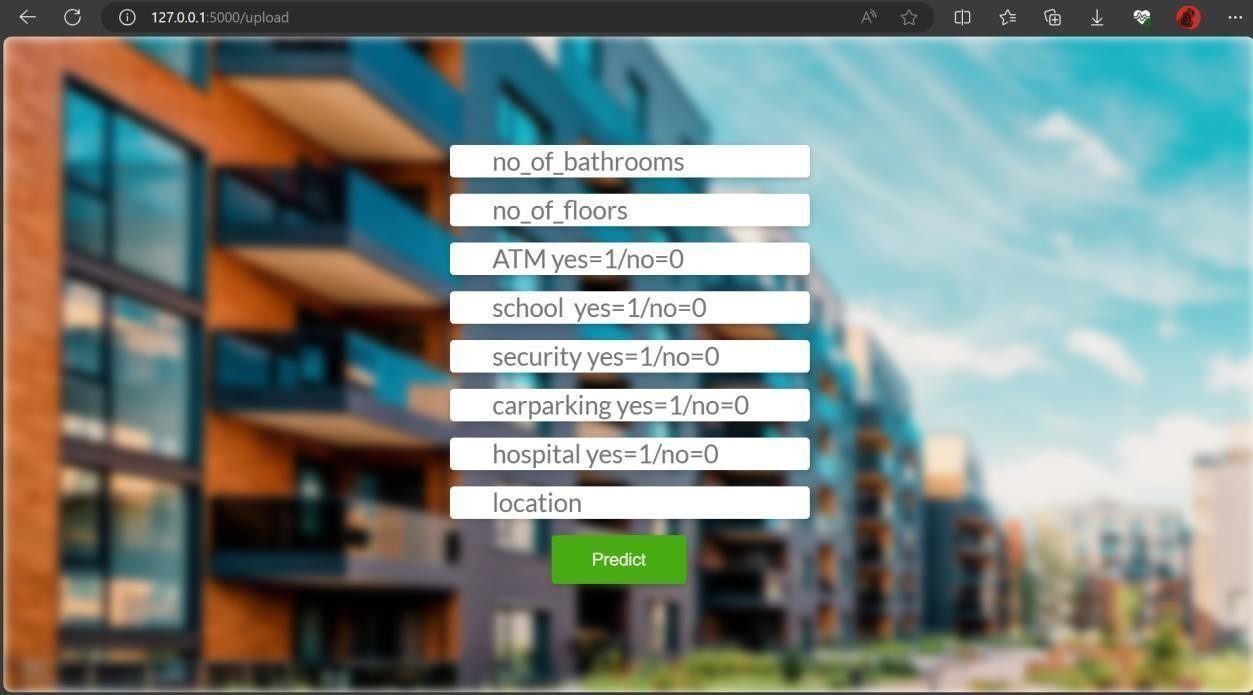
**Home page**



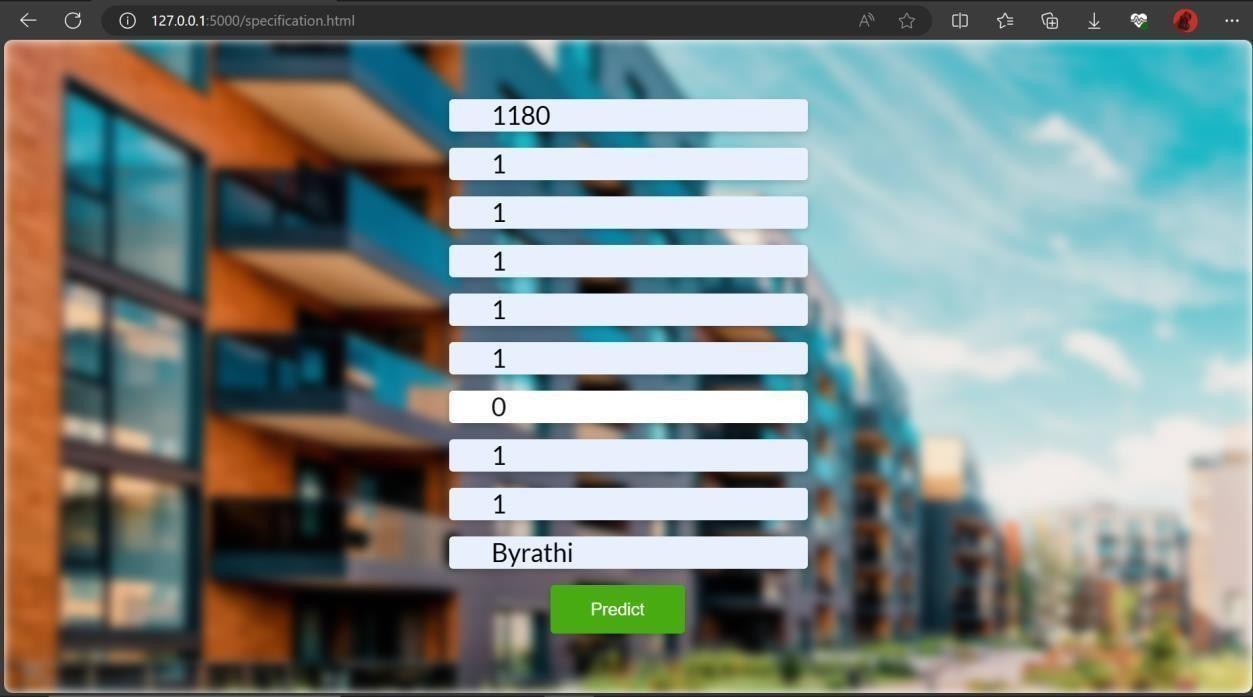
# Blueprint\_upload Page



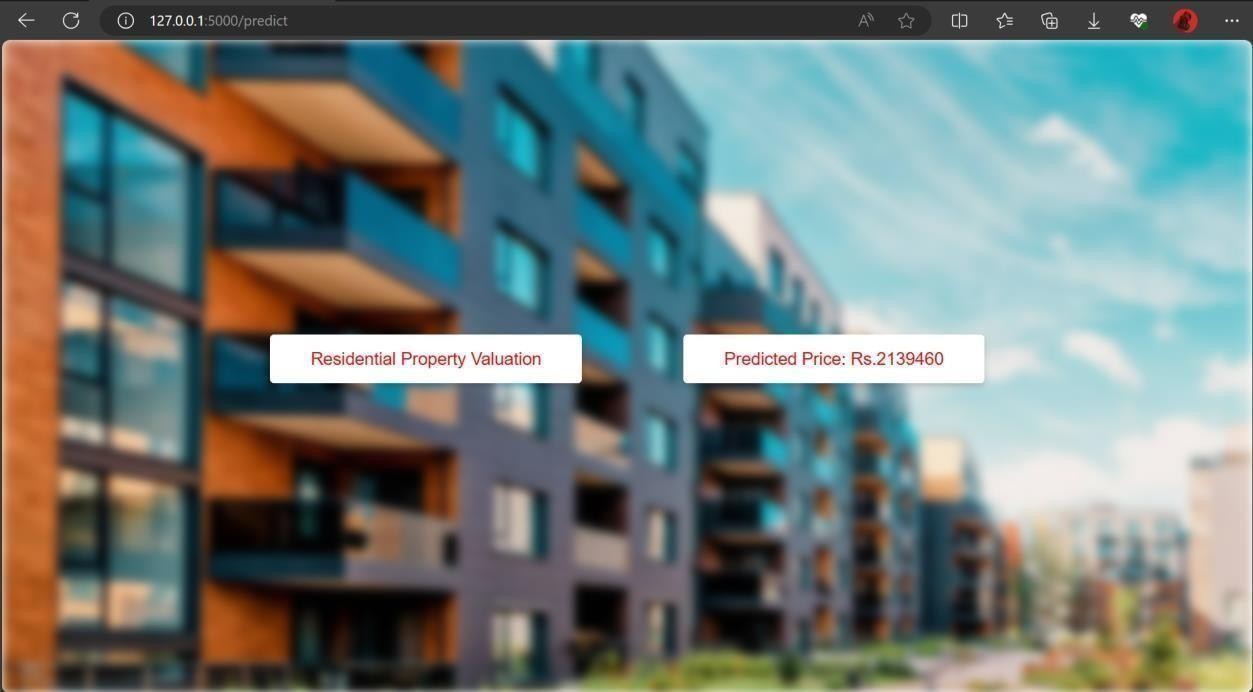
**Blueprint\_specification Page**



# Specifications Page



**Result Page**



# REFERENCES

1. Tea Lallukka, Anoushka Millear, Amanda Pain, Monica Cortinovis, and Giorgia Giussani. Gbd 2015 mortality and causes of death collaborators. global, regional, and national life expectancy, all-cause mortality, and cause-speciﬁ c mortality for 249 causes of death, 1980-2015: a systematic analysis for the global burden of disease study 2015 (vol 388, pg 1459, 2016). *Lancet*, 389(10064):E1–E1, 2017.
2. Global status report on noncommunicable diseases 2014. geneva, world health orga- nization, 2014. *Available from:* [*http://www.who.int/nmh/publications/ncd-status-report-*](http://www.who.int/nmh/publications/ncd-status-report-) *2014/en/*.
3. World health organization. density of physicians. [*http://www.who.int/gho/health*](http://www.who.int/gho/health) *work- force/physicians density/en/, 2017. [Online; accessed 18-May-2018]*.
4. KC Santosh. Speech processing in healthcare: Can we integrate? In *Intelligent Speech Signal Processing*, pages 1–4. Elsevier, 2019.
5. Himadri Mukherjee, Subhankar Ghosh, Shibaprasad Sen, Obaidullah Sk Md, KC Santosh, Santanu Phadikar, and Kaushik Roy. Deep learning for spoken lan- guage identiﬁcation: Can we visualize speech signal patterns? *Neural Computing and Applications*, 31(12):8483–8501, 2019.
6. Himadri Mukherjee, Sk Md Obaidullah, KC Santosh, Santanu Phadikar, and Kaushik Roy. Line spectral frequency-based features and extreme learning machine for voice activity detection from audio signal. *International Journal of Speech Technol- ogy*, 21(4):753–760, 2018.
7. Victor A McKusick, John T Jenkins, and George N Webb. The acoustic basis of the chest examination; studies by means of sound spectrography. *American review of tuberculosis*, 72(1):12–34, 1955.
8. ARA Sovijarvi, J Vanderschoot, and JE Earis. Standardization of computerized res- piratory sound analysis. *European Respiratory Review*, 10(77):585–585, 2000.
9. Volker Gross, LJ Hadjileontiadis, Thomas Penzel, Ulrich Koehler, and C Vogelmeier. Multimedia database” marburg respiratory sounds (mars)”,”. In *Proc. 25th Annual Int Engineering in Medicine and Biology Society Conf. of the IEEE*, volume 1, pages 456– 457, 2003.
10. Rale: A computer-assisted instructional package. *Respir Care 1990;35:1006.*
11. William H Wolberg, W Nick Street, and Olvi L Mangasarian. Machine learning tech- niques to diagnose breast cancer from image-processed nuclear features of ﬁne nee- dle aspirates. *Cancer letters*, 77(2-3):163–171, 1994.
12. Sotiris B Kotsiantis, I Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classiﬁcation techniques. *Emerging artiﬁcial intelligence applications in com- puter engineering*, 160(1):3–24, 2007.
13. Arumugam Kandaswamy, C Sathish Kumar, Rm Pl Ramanathan, S Jayaraman, and N Malmurugan. Neural classiﬁcation of lung sounds using wavelet coefﬁcients. *Computers in biology and medicine*, 34(6):523–537, 2004.
14. Jack V Tu. Advantages and disadvantages of using artiﬁcial neural networks versus logistic regression for predicting medical outcomes. *Journal of clinical epidemiology*, 49(11):1225–1231, 1996.
15. Marcin Raniszewski. The edited nearest neighbor rule based on the reduced ref- erence set and the consistency criterion. *Biocybernetics and Biomedical Engineering*, 30(1):31–40, 2010.
16. Sameer Alsmadi and Yasemin P Kahya. Design of a dsp-based instrument for real-
17. time classiﬁcation of pulmonary sounds. *Computers in biology and medicine*, 38(1):53– 61, 2008.
18. Geert Meyfroidt, Fabian Gu¨ iza, Jan Ramon, and Maurice Bruynooghe. Machine learning techniques to examine large patient databases. *Best Practice & Research Clin- ical Anaesthesiology*, 23(1):127–143, 2009.
19. Shijun Wang and Ronald M Summers. Machine learning and radiology. *Medical image analysis*, 16(5):933–951, 2012.
20. I˙nan Gu¨ ler, Hu¨ seyin Polat, and Uc¸man Ergu¨ n. Combining neural network and ge- netic algorithm for prediction of lung sounds. *Journal of Medical Systems*, 29(3):217– 231, 2005.
21. Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *expert systems with applications*, 36(10):11994–12000, 2009.
22. Jyh-Shing Roger Jang, Chuen-Tsai Sun, E Mizutani, and Y-C Ho. Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence. *PROCEEDINGS-IEEE*, 86(3):600–603, 1998.
23. Himadri Mukherjee, Sk Md Obaidullah, Santanu Phadikar, and Kaushik Roy. Misna- a musical instrument segregation system from noisy audio with GTCC-s features and extreme learning. *Multimedia Tools and Applications*, 77(21):27997–28022, 2018.
24. Himadri Mukherjee, Ankita Dhar, Sk Md Obaidullah, KC Santosh, Santanu Phadikar, and Kaushik Roy. Linear predictive coefﬁcients-based feature to identify top-seven spoken languages. *International Journal of Pattern Recognition and Artiﬁcial Intelligence*, 34(06):2058006, 2020.
25. Bruno M Rocha, Dimitris Filos, Lu´ıs Mendes, Gorkem Serbes, Sezer Ulukaya, Yasemin P Kahya, Niksˇa Jakovljevic, Tatjana L Turukalo, Ioannis M Vogiatzis, Eleni Perantoni, et al. An open access database for the evaluation of respiratory sound classiﬁcation algorithms. *Physiological measurement*, 40(3):035001, 2019.
26. Murat Aykanat, O¨ zkan Kılıc¸, Bahar Kurt, and Sevgi Saryal. Classiﬁcation of lung sounds using convolutional neural networks. *EURASIP Journal on Image and Video Processing*, 2017(1):1–9, 2017.
27. Himadri Mukherjee, Sk Md Obaidullah, KC Santosh, Santanu Phadikar, and Kaushik Roy. A lazy learning-based language identiﬁcation from speech using mfcc- 2 features. *International Journal of Machine Learning and Cybernetics*, 11(1):1–14, 2020.
28. Renard Xaviero Adhi Pramono, Syed Anas Imtiaz, and Esther Rodriguez-Villegas. Evaluation of features for classiﬁcation of wheezes and normal respiratory sounds. *PloS one*, 14(3):e0213659, 2019.
29. Jyotibdha Acharya and Arindam Basu. Deep neural network for respiratory sound classiﬁcation in wearable devices enabled by patient speciﬁc model tuning. *IEEE transactions on biomedical circuits and systems*, 14(3):535–544, 2020.
30. Zu¨ mray Dokur. Respiratory sound classiﬁcation by using an incremental supervised neural network. *Pattern Analysis and Applications*, 12(4):309–319, 2009.
31. Vinita Shivakumar. Classiﬁcation of respiratory sounds.
32. Pedro Sousa Faustino. Crackle and wheeze detection in lung sound signals using convolutional neural networks. 2019.
33. Yi Ma, Xinzi Xu, and Yongfu Li. Lungrn+ nl: An improved adventitious lung sound classiﬁcation using non-local block resnet neural network with mixup data augmen- tation. In *Interspeech*, pages 2902–2906, 2020.
34. Dimitra Emmanouilidou, Eric D McCollum, Daniel E Park, and Mounya Elhilali. Computerized lung sound screening for pediatric auscultation in noisy ﬁeld envi- ronments. *IEEE Transactions on Biomedical Engineering*, 65(7):1564–1574, 2017.
35. Ipek Sen, Murat Saraclar, and Yasemin P Kahya. A comparison of svm and gmm- based classiﬁer conﬁgurations for diagnostic classiﬁcation of pulmonary sounds. *IEEE Transactions on Biomedical Engineering*, 62(7):1768–1776, 2015.
36. Fatih Demir, Aras Masood Ismael, and Abdulkadir Sengur. Classiﬁcation of lung sounds with cnn model using parallel pooling structure. *IEEE Access*, 8:105376– 105383, 2020.
37. Hai Chen, Xiaochen Yuan, Zhiyuan Pei, Mianjie Li, and Jianqing Li. Triple- classiﬁcation of respiratory sounds using optimized s-transform and deep residual networks. *IEEE Access*, 7:32845–32852, 2019.
38. Xuen Hoong Kok, Syed Anas Imtiaz, and Esther Rodriguez-Villegas. A novel method for automatic identiﬁcation of respiratory disease from acoustic recordings. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2589–2592. IEEE, 2019.
39. Gae¨tan Chambres, Pierre Hanna, and Myriam Desainte-Catherine. Automatic de- tection of patient with respiratory diseases using lung sound analysis. In *2018 In- ternational Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2018.
40. Gokhan Altan, Yakup Kutlu, and Novruz Allahverdi. Deep learning on computer- ized analysis of chronic obstructive pulmonary disease. *IEEE journal of biomedical and health informatics*, 24(5):1344–1350, 2019.
41. Adam Rao, Emily Huynh, Thomas J Royston, Aaron Kornblith, and Shuvo Roy.
42. Acoustic methods for pulmonary diagnosis. *IEEE reviews in biomedical engineering*, 12:221–239, 2018.
43. Arnon Cohen and Dorota Landsberg. Analysis and automatic classiﬁcation of breath sounds. *IEEE Transactions on Biomedical Engineering*, (9):585–590, 1984.
44. M Bahoura and C Pelletier. Respiratory sounds classiﬁcation using cepstral analysis and gaussian mixture models. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 9–12. IEEE, 2004.
45. P Mayorga, C Druzgalski, RL Morelos, OH Gonzalez, and J Vidales. Acoustics based assessment of respiratory diseases using gmm classiﬁcation. In *2010 Annual Inter- national Conference of the IEEE Engineering in Medicine and Biology*, pages 6312–6316. IEEE, 2010.
46. Rajkumar Palaniappan, Kenneth Sundaraj, and Sebastian Sundaraj. A comparative study of the svm and k-nn machine learning algorithms for the diagnosis of respi- ratory pathologies using pulmonary acoustic signals. *BMC bioinformatics*, 15(1):1–8, 2014.
47. MARIO Milicevic, IGOR Mazic, and MIRJANA Bonkovic. Classiﬁcation accuracy comparison of asthmatic wheezing sounds recorded under ideal and real-world con- ditions. In *15th International Conference on Artiﬁcial Intelligence, Knowledge Engineering and Databases (AIKED 2016), Venice*, 2016.
48. BM Rocha, L Mendes, I Chouvarda, P Carvalho, and RP Paiva. Detection of cough and adventitious respiratory sounds in audio recordings by internal sound analysis. In *International Conference on Biomedical and Health Informatics*, pages 51–55. Springer, 2017.
49. Gorkem Serbes, Sezer Ulukaya, and Yasemin P Kahya. An automated lung sound
50. preprocessing and classiﬁcation system based onspectral analysis methods. In *Inter- national Conference on Biomedical and Health Informatics*, pages 45–49. Springer, 2017.
51. Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
52. J Earis. Lung sounds. *Thorax*, 47(9):671, 1992.
53. Hans Pasterkamp, Steve S Kraman, and George R Wodicka. Respiratory sounds: advances beyond the stethoscope. *American journal of respiratory and critical care medicine*, 156(3):974–987, 1997.
54. SK Chowdhury and AK Majumder. Frequency analysis of adventitious lung sounds.*Journal of biomedical engineering*, 4(4):305–312, 1982.
55. Zbigniew Korona and Mieczyslaw M Kokar. Lung sound recognition using model- theory-based feature selection and fusion. *Applied Signal Processing*, 5(3):152, 1998.
56. Howelldcj.in:geoffreyjl,stevends,editors. signs of respiratory disease: lung sounds.
57. RB Urquhart, J McGhee, JES Macleod, SW Banham, and F Moran. The diagnostic value of pulmonary sounds: a preliminary study by computer-aided analysis. *Com- puters in biology and medicine*, 11(3):129–139, 1981.
58. Mcgee s. auscultation of the lungs. philadelphia: W.b. saunders;. 2012: 251–66 [chap- ter 28].
59. Steven m. auscultation of the lungs. saint louis: W.b. saunders;. 2007: 326–45 [chapter 27].
60. Sankar K Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, classiﬁaction.1992.Kun-Hsi Tsai, Wei-Chien Wang, Chui-Hsuan Cheng, Chan-Yen Tsai, Jou-Kou Wang, Tzu-Hao Lin, Shih-Hau Fang, Li-Chin Chen, and Yu Tsao. Blind monaural source separation on heart and lung sounds based on periodic-coded deep autoencoder. *IEEE Journal of Biomedical and Health Informatics*, 24(11):3203–3214, 2020.
61. Mukherjee Himadri, Hanan Salam, and KC Santosh. Lung health analysis: Ad- ventitious respiratory sound classiﬁcation using ﬁlterbank energies. *IEEE Journal of Biomedical and Health Informatics*, 2021.