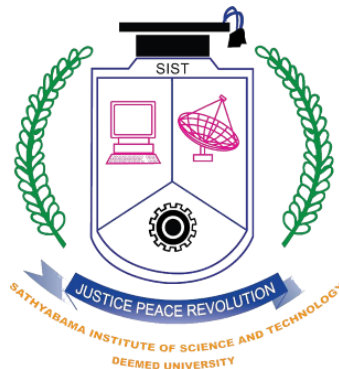


WIRELESS STETHOSCOPE FOR DIAGNOSIS OF COMMUNITY ACQUIRED PNEUMONIA

Submitted in partial fulfillment of the requirements of the award of Master of
Technology degree in Medical Instrumentation

by

INDU.S (42440002)



DEPARTMENT OF BIOMEDICAL ENGINEERING

SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project report is the bonafide work of **INDU.S (42440002)** who carried out the project entitled "**Wireless Stethoscope for Diagnosis of Community Acquired Pneumonia**" under our supervision from **June 2023 to April 2024**.

Dr. J. BETHANNEY JANNEY M.E., Ph.D.,

Internal Guide

Dr. T. SUDHAKAR, M.Sc., Ph.D.,

Head of the Department

Head of the Department
Department of Biomedical Engineering
Sathyabama Institute of Science & Technology
(Deemed to be University)
Category I University by UGC
Jeyapalan Nagar, Rajiv Gandhi Salai,
Chennai - 600 119

Submitted for Viva-voce Examination held on 20-04-2024.

Internal Examiner

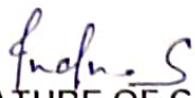
20/4/24

External Examiner

DECLARATION

I, **INDU.S (42440002)** hereby declare that the Project report entitled "**Wireless Stethoscope for Diagnosis of CommunityAcquired Pneumonia**" done by me under the guidance of **Dr. J. Bethannev Janney, M.E., Ph.D.,** Department of Biomedical Engineering is submitted in partial fulfillment of the requirements for the award of **Master of Technology** degree in **Medical Instrumentation**.

Date: 22-04-2024


SIGNATURE OF CANDIDATE

Place: Chennai.

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **BOARD OF MANAGEMENT OF SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SUDHAKAR, M.Sc., Ph.D.**, Head of the Department, Department of Biomedical Engineering for providing me with the necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J. Bethanney Janney, M.E., Ph.D.** for her valuable guidance, suggestions, and constant encouragement that paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Biomedical Engineering who were helpful in many ways for the completion of the project.

I would like to express my thanks to my parents for their unwavering support to me during these years.

ABSTRACT

Globally, community-acquired pneumonia (CAP) is a major cause of death because there are no widely used monitoring methods. Since tachypnea and crackling lung sounds are known to be typical symptoms of community-acquired pneumonia, the wireless stethoscope has great potential for application in clinical settings. In this planned effort, a clinical experiment with a range of age participants examines the feasibility of using a wireless stethoscope for CAP diagnosis. The lung sounds of individuals with CAP are recorded on both the left and right sides and classified as mild, medium, or severe. The software in this proposed work uses a deep-learning algorithm to detect pneumonia and categorize lung sounds into phases. Medical professionals use a wireless stethoscope to record lung sounds. Medical professionals use a wireless stethoscope to record lung sounds. Therefore, compared to X-ray and CT scans, the suggested design can be employed as an application tool for pneumonia diagnosis that is simpler, less expensive, and less dangerous. Additionally, it shortens the time needed to diagnose pneumonia.

Keywords — *Pneumonia, wireless stethoscope, CAP, Deep learning.*

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	LIST OF ABBREVIATIONS	viii
	LIST OF FIGURES	ix
	LIST OF TABLES	xi
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 PNEUMONIA	2
	1.2.1 Types of Pneumonia	3
2	LITERATURE REVIEW	7
3	AIM AND SCOPE	17
	3.1 AIM	17
	3.2 EXISTING SYSTEM	17
	3.2.1 Disadvantages	17
	3.3 PROPOSED SYSTEM	17
	3.3.1 Advantages	18
	3.3 SCOPE	18
4	MATERIALS AND METHODS	19
	4.1 BLOCK DIAGRAM	19
	4.2 HARDWARE REQUIREMENTS	20
	4.2.1 Arduino Uno	20
	4.2.2 LM393 Sound Sensor Module	23
	4.2.3 TTL to USB Converter Module	25
	4.3 HARDWARE SETUP	27
	4.4 CIRCUIT DIAGRAM	28
	4.5 SOFTWARE REQUIREMENTS	29
	4.5.1 Arduino IDE	29
	4.5.1.1 To Download Software	31
	4.5.1.2 To Create and Run Program	31
	4.5.2 Google Colab	31
	4.5.2.1 To Create and Run	32
	4.6 ABOUT THE DATASET	33
	4.6.1 Data Annotation	34

	4.7 TOOLS USED FOR IMPLEMENTATION	34
	4.8 VISUALIZATION OF SOUND FILES OF DIFFERENT CATEGORY	35
	4.9 DATA PRE-PROCESSING	36
	4.9.1 Data Augmentation	36
	4.9.2 Denoising	37
	4.9.3 Filtering Process	39
	4.9.4 Power Spectrum	41
	4.10 FEATURE EXTRACTION	43
	4.11 CONVOLUTIONAL NEURAL NETWORK	46
	4.11.1 Implementation of CNN Model	46
	4.11.2 Compiling of CNN Model	51
	4.11.3 Generating of CNN Model	52
5	RESULTS AND DISCUSSION	53
	5.1 MODEL EVALUATION	53
	5.2 CONFUSION MATRIX	57
	5.3 CLASSIFICATION REPORT	59
	5.3.1 Interpretation of Metrics	60
	5.3.2 Overall Performance	61
	5.4 AUC-ROC CURVE	62
	5.5 TESTING THE DATA	65
6	SUMMARY AND CONCLUSION	66
	6.1 SUMMARY	66
	6.2 CONCLUSION	68
	6.3 FUTURE SCOPE	69
	REFERENCES	71

LIST OF ABBREVIATIONS

ABBREVIATIONS	DESCRIPTION
CAP	Community-Acquired Pneumonia
CNN	Convolutional Neural Network
Colab	Collaboratory
COPD	Chronic Obstructive Pulmonary
DCT	Discrete Cosine Transform
FFT	Fast Fourier Transform
HAP	Hospital Acquired Pneumonia
OvA	One-vs-All
OvR	One-vs-Rest
PCB	Printed Circuit Board
RSV	Respiratory Syneytial Virus
STFT	Short-time Fourier Transform

LIST OF FIGURES

FIGURE No.	DESCRIPTION	PAGE No.
1.1	Pneumonia Affected Alveoli	3
1.2	Community Acquired Pneumonia	4
1.3	Hospital Acquired Pneumonia	5
3.1	Flow chart of Proposed Method	18
4.1	Block diagram of System	19
4.2	Arduino Uno	21
4.3	LM393 Sound Sensor	24
4.4	TTL to USB Converter	26
4.5	Block diagram of Wireless Stethoscope	27
4.6	Circuit diagram	28
4.7	Wireless Stethoscope	29
4.8	Wave plot of each category	36
4.9	Data distribution after data augmentation	37
4.10	Block diagram of spectral subtraction	39
4.11	Signal obtained after denoising	39
4.12	Block diagram of butter-worth bandpass filter	41
4.13	Signal obtained after filtering	41
4.14	Power spectrum of each category	43
4.15	Block diagram of log mel spectrogram	44
4.16	log Mel Spectrogram of each category	46
4.17	Architecture of CNN Model	50
4.18	Summary of CNN Model	51
5.1	Distribution of datasets	54
5.2	Accuracy curve	55
5.3	Loss curve	55
5.4	Distribution of testing data	56

5.5	Testing accuracy curve	56
5.6	Testing loss curve	57
5.7	Comparison chart of testing and validation chart	57
5.8	Model Confusion Matrix	58
5.9	Confusion Matrix	59
5.10	Classification Report	60
5.11	Classification Metrics	61
5.12	AUC-ROC curve	63
5.13	AUC-ROC score	64
5.14	Testing Result	65
6.1	Comparison of Model Accuracies	68
6.2	Comparison of Model AUC	68
6.3	Comparison of Model F1-score	68

LIST OF TABLES

TABLE No.	DESCRIPTION	PAGE No.
2.1	Literature Review Comparison Chart	15
4.1	Data Annotation	34
5.1	Model Performance	65
6.1	Comparison Table of Proposed Method with other Existing Methods	67

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Respiratory health stands as a cornerstone of human well-being, underpinning our ability to thrive and engage fully in daily activities. From the rhythmic expansion and contraction of the chest to the exchange of oxygen and carbon dioxide within the lungs, every breath sustains life itself (<https://www.who.int/news-room/fact-sheets/detail/pneumonia>). Indeed, the integrity of the respiratory system profoundly influences the quality and longevity of human existence. Central to the notion of respiratory health is the early detection and diagnosis of respiratory conditions. The ability to discern subtle changes in respiratory function and identify signs of pathology holds profound implications for treatment outcomes and patient prognosis. Respiratory conditions, ranging from common ailments like bronchitis to life-threatening diseases such as pneumonia and chronic obstructive pulmonary disease (COPD), can exert a profound toll on individual health and societal well-being. Early detection of respiratory conditions allows for timely intervention and tailored treatment strategies, mitigating the progression of disease and alleviating symptom burden. By identifying respiratory abnormalities in their incipient stages, clinicians can implement targeted therapeutic interventions, optimize medication regimens, and provides personalized care plans tailored to individual patient needs. Moreover, early diagnosis facilitates the implementation of preventive measures and lifestyle modifications aimed at minimizing disease exacerbations and promoting respiratory health maintenance. The impact of timely diagnosis extends beyond individual patients to encompass broader public health considerations. Rapid identification of respiratory outbreaks, such as influenza epidemics or emerging infectious diseases like COVID-19, enables proactive public health responses, including containment measures, vaccination campaigns, and epidemiological surveillance. Through vigilant monitoring of respiratory health indicators, healthcare systems can effectively allocate resources, coordinate emergency responses, and safeguard population health on a global scale (Hayes et al., 2018). Furthermore, the ability to detect and diagnose respiratory conditions early serves as a linchpin for advancing medical knowledge and scientific inquiry. By elucidating the pathophysiological mechanisms underlying respiratory

diseases and exploring novel diagnostic modalities, researchers can foster innovation, drive therapeutic development, and propel the evolution of evidence-based clinical practice. Through interdisciplinary collaboration and translational research endeavours, the frontiers of respiratory medicine continue to expand, offering hope for improved outcomes and enhanced quality of life for individuals affected by respiratory disorders. In essence, the ability to discern the subtle nuances of respiratory health and detect deviations from the norm represents a pivotal aspect of modern healthcare. By recognizing the intrinsic value of respiratory well-being and embracing a proactive approach to diagnosis and intervention, we can strive towards a future characterized by vitality, resilience, and flourishing for all individuals, irrespective of their respiratory challenges.

1.2 PNEUMONIA

Pneumonia, an inflammatory condition of the lungs, represents a significant global health burden, characterized by its diverse etiology, clinical variability, and potential for serious complications. Understanding the intricacies of pneumonia is paramount for developing effective diagnostic and therapeutic strategies aimed at mitigating its impact on public health (McIntosh K et al., 2002). Pneumonia can arise from various infectious and non-infectious etiologies, with bacterial, viral, fungal, and parasitic pathogens implicated in its pathogenesis. *Streptococcus pneumoniae*, *Haemophilus influenza*, and *Mycoplasma pneumoniae* are among the common bacterial culprits, while influenza viruses, respiratory syncytial virus (RSV), and adenovirus predominate among viral pathogens. Pneumonia represents a leading cause of morbidity and mortality worldwide, particularly among vulnerable populations such as young children, older adults, immune compromised individuals, and those with underlying comorbidities. The incidence of pneumonia varies geographically and temporally, influenced by factors such as socioeconomic status, environmental conditions, vaccination coverage, and antimicrobial resistance patterns. In 2017, pneumonia claimed the lives of more than 808,000 kids under the age of five, accounting for 15% of all pediatric fatalities (Hunton et al., 2019). The clinical presentation of pneumonia can range from mild respiratory symptoms, such as cough, fever, dyspnoea, and chest pain, to severe systemic manifestations, including respiratory failure, septic shock, and multi-organ dysfunction. The variability in clinical features poses challenges for accurate diagnosis and timely

initiation of appropriate treatment interventions. For a healthy person, the tiny sacs called alveoli are filled with air while breathing. When someone has pneumonia, their breathing becomes difficult as their alveoli are filled with pus and fluid. Pneumonia represents a complex and dynamic clinical entity characterized by its multifactorial etiology, heterogeneous clinical presentation, and diagnostic challenges. While existing diagnostic approaches play a crucial role in pneumonia management, the integration of innovative techniques such as respiratory sound analysis holds promise for improving diagnostic precision, optimizing patient outcomes, and advancing the field of respiratory medicine in the years to come. Fig. 1.1 shows the difference between normal alveoli and alveoli when affected from pneumonia.

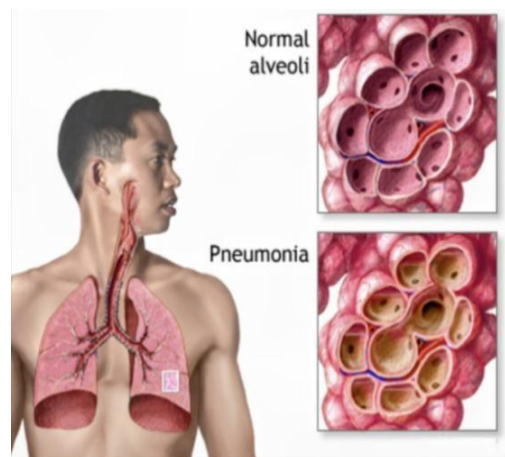


Fig:1.1:Pneumonia affected alveoli

(courtesy- <https://medlineplus.gov/ency/article/000145.htm>)

1.2.1 Types of Pneumonia

(a) Community-Acquired Pneumonia

Community-Acquired is the term used to describe pneumonia that is obtained outside of a medical setting. Fig. 1.2 shows the cause of CAP. Some causes are:

Bacteria: The most common cause of CAP is infection with the bacteria *Streptococcus pneumoniae*, commonly known as pneumococcal disease. Meningitis, sinus infections, and ear infections can all be caused by pneumococcal disease. Typical pneumonia, often with milder symptoms, is caused by the bacteria *Mycoplasma pneumoniae*. *Haemophilus influenza*, *Chlamydia pneumoniae*, and *Legionella* (which causes Legionnaires' disease) are additional bacteria that causes CAP.

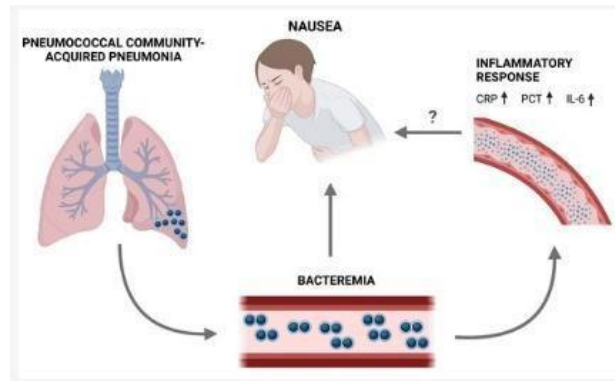


Fig:1.2:Community Acquired Pneumonia

(courtesy - <https://www.news-medical.net/health/Pneumonia-Classification.aspx>)

Viruses: Pneumonia can sometimes be caused by the same virus that causes the common cold, influenza (flu), COVID-19, and respiratory syncytial virus (RSV).

Fungi (molds): Fungi are rare causes of pneumonia, including *Cryptococcus*, *Pneumocystis jirovecii*, and *Coccidioides*. The risk of fungal pneumonia is higher in people with weak immune systems.

Protozoa: *Toxoplasma* cause rare form of pneumonia.

(b) Hospital-Acquired Pneumonia

While obtaining care in a hospital or healthcare institution for another ailment or procedure, a patient may have hospital-acquired pneumonia (HAP). In general, HAP is more hazardous than community-acquired pneumonia because it is frequently brought on by bacteria that are resistant to antibiotics, like methicillin-resistant *Staphylococcus aureus* (MRSA) (Jorge R et al., 2020). Therefore, HAP can make you worse and harder to treat.

Fig. 1.3 shows the affected lungs due to hospital acquired pneumonia.

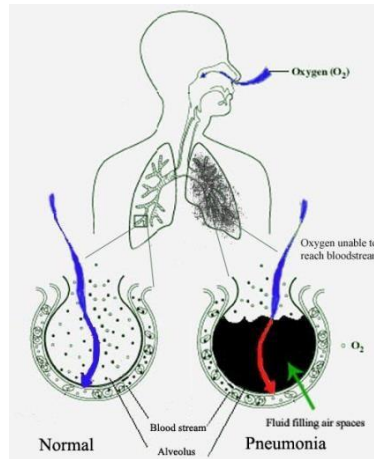


Fig:1.3:Hospital Acquired Pneumonia

(courtesy-https://www.wikidoc.org/index.php/Hospital-acquired_pneumonia_pathophysiology)

We shall concentrate on community-acquired pneumonia (CAP) in this paper. "Community-acquired pneumonia" is pneumonia contracted outside of the healthcare system. Globally, the number of children suffering from CAP is on the rise. Detection of CAP at an early stage can help save the lives of children.

Clinically speaking, everyday medical activities are significantly impacted by the monitoring of various CAP treatment stages. Wireless stethoscope-based chest auscultation is currently the most used way to monitor CAP since it may be utilized for the whole course of treatment, including the diagnosis, follow-up exam, and prognosis. However, its listeners' subjective experiences with its clinical consequences often result in inaccurate clinical judgments. Lung sound analysis has been utilized to address this, highlighting the potential of machine learning-based algorithms for unbiased lung state evaluation.

Recent developments in deep learning algorithms have significantly improved the ability to recognize abnormal lung sounds. As of latest update in January 2022, there have been notable advancements in the application of deep learning techniques for respiratory sound classification. These developments have significantly enhanced the accuracy and efficiency of respiratory sound analysis, paving the way for improved diagnostic capabilities and patient care. CNNs have emerged as a powerful tool for feature learning in respiratory sound classification tasks. By leveraging hierarchical feature extraction capabilities, CNNs can automatically learn discriminative features from raw audio signals, enabling more

robust and accurate classification performance. Transfer learning, a technique where pre-trained neural network models are fine-tuned on task-specific datasets, has gained popularity in respiratory sound classification. By leveraging features learned from large-scale audio datasets, transfer learning facilitates the development of more efficient and generalizable classification models, particularly in scenarios with limited training data. Attention mechanisms have been integrated into deep learning architectures to dynamically weigh the importance of different segments of respiratory sound recordings. These mechanisms enable the model to focus on salient features while ignoring irrelevant noise, improving classification accuracy and interpretability. Recent research efforts have explored the integration of multi-modal data, such as respiratory sound recordings and clinical metadata (e.g., patient demographics, medical history), to enhance classification performance. Multi-modal learning approaches leverage complementary information from diverse data sources, providing more comprehensive understanding of respiratory conditions and facilitating more accurate diagnostic predictions. Data augmentation techniques, such as time stretching, pitch shifting, and noise injection, have been employed to augment the size and diversity of training datasets for respiratory sound classification (Hemalatha M et al., 2022). These techniques help mitigate overfitting and improve the generalization capabilities of deep learning models, particularly in scenarios with limited annotated data. With advancements in hardware acceleration and signal processing algorithms, real-time respiratory sound classification systems have become increasingly feasible. These systems enable rapid and automated analysis of respiratory sounds at the point of care, facilitating timely diagnosis and treatment decision-making. The availability of open-source respiratory sound datasets and benchmarking platforms has accelerated research in respiratory sound classification. Datasets such as the R.A.L.E. dataset, the Freesound dataset, and the MIMIC-CXR dataset provide researchers with access to annotated respiratory sound recordings for model development and evaluation. Overall, recent developments in deep learning techniques have revolutionized respiratory sound classification offering new opportunities for early detection, diagnosis, and monitoring for respiratory conditions. These methods allow CAP to be divided into mild and severe cases.

CHAPTER 2

LITERATURE REVIEW

Dongmin Huang et al., (2023) undertook a multi-center clinical experiment at four hospitals to see if a wireless stethoscope could be used to identify CAP and gauge its prognosis. Children with CAP are tracked from the time of diagnosis through recovery from the trial and after improvement. A bilateral pulmonary audio auxiliary model (BPAM) is suggested for the study of lung sounds. Phases of BPAM include pre-processing, feature extraction, and classification. Through pre-processing, invalid and severely damaged fragments are first eliminated. The logMel spectrogram is then used to describe the lung sound by applying the necessary temporal-frequency data. The logMel spectrogram's structured and contextual data are ultimately extracted using a deep learning algorithm. According to the clinical validation, the specificity and sensitivity of BPAM are over 92% for the CAP diagnosis and prognosis in the subject-dependent experiment and over 50% and 39%, respectively, for the CAP diagnosis and prognosis in the subject-independent experiment.

Dongmin Huang et al., (2023) in order to enhance the recognition of lung sounds in kids with CAP, a contrastive embedding-based domain adaptation network was created. The data obtained from the wireless stethoscope exhibits the same data distribution when used in recognition using different techniques. Another restriction is the lack of data. As a result, the author has suggested that CEDANN gives more input combinations by creating different contrast tuples out of a number of audio samples from different classes. This process uses embedding to improve intra-class compactness and inter-class separability, which can typically address the issue of data scarcity. In order to address the inconsistent distribution of individuals, the domain classifier used by CEDANN enables the embedding layer to acquire subject-independent but task-dependent properties.

Seyed Amir Latifi et al., (2023) developed a system to root certain features in eight different kinds of respiratory sounds. The author proposed two novel lung sound point lines, the Maximum Entropy Gabor Filter Bank (MEGFB) and the Maximum Entropy Mel Filter Bank (MEMFB). Deep convolutional neural networks have trained on 50% of the data in order to move on to classification. The frequency characteristics from the lung signal with the least amount of redundancy are

selected using MAMFB filter banks anchored on those frequency characteristics. The energy of each filter after the signal has been processed by the filter bank is calculated and given to the neural network as an input. DCNN pulls out 24 characteristics. Performance of the system is improved by DCNN characteristics based on the combined MEGFB and MFCC algorithms and the time-frequency aspects of the lung sound signal. Thus, it is possible to identify this kind of lung infection. Among the algorithms tested, MAMFB demonstrated a high degree of delicacy (93), making it dependable and helpful for hospitals.

Sanchit Vashisht et al., (2023) suggested a CNN-GAN-based model to categorize the pneumonia illness. The online tool Kaggle provided the dataset that was used in this investigation. Depending on whether they are viral pneumonia or bacterial pneumonia, the obtained images are classed. The datasets are enhanced via the generative adversarial network (GAN) function. Following the collection of all the data, a Jupyter notebook is utilized to send it to Python for preparation. Following standardization, the data is divided into training and testing datasets, and the CNN model is used to train and test the model over these datasets. The accuracy of the suggested model was 98%.

Chitra R. et al., (2023) designed a digital stethoscope that monitors lung sounds, and a CNN model is used to identify potential illnesses. A sensor called Electrodermal Activity (EDA) uses Wi-Fi Pro to transmit signal to record the sporadic signal of lung activity. NodeMCU processes the lung signal using the convolution neural network (CNN) model, and the output is the recognized ailment, such as pneumonia or upper respiratory tract infection. A graphical representation of the detected disease is displayed in the THINKSPEAK program as the Pushover application sends a notification. Hence, this method provides an accuracy of 95% in detecting lung disease.

Trang Nguyen Minh et al., (2023) based on the Arduino UNO R3, we created a low-cost electronic stethoscope with a hearing jack and real-time heart sound realization. A chest piece and a microphone are features of the newly created stethoscope. The bell half, a smaller circular end of the chest piece, picks up low frequency noises. Additionally, the diaphragm, a bigger disk-shaped end of the chest piece, is responsible for picking up higher-frequency noises. The microphone serves as a transducer, transforming the acquired signal into an electrical signal that

is equivalent. Analog Front End (AFE) amplifies the electrical signal in two stages. Filtering out noise is done by the RC filter. After signal processing, an analog signal is heard by the earphone. The corresponding digital signal is obtained in EXCEL via Arduino UNO and Bluetooth. The data stored in Excel is processed in MatLab to detect the peak of the signal.

Sowrav Chowdhury et al., (2022) produced three inexpensive digital stethoscopes, and the effectiveness of their six-time and frequency domain characteristics is contrasted. For data collection, three stethoscopes—Steth 1, Steth 2, and Steth 3—are connected to a microphone. The signals are captured and processed using Thinklabs Phonocardiography software, an open-source cross-platform program. The filter equalization feature is used to first filter the signals to improve lung sound quality. The collected signal is divided into non-overlapping segments known as epochs. Then, for each epoch, the time and frequency domain features are computed. Finally, Pearson's linear correlation coefficients are used to analyze the features that were gathered. Steth 1 had the most accurate result out of the three stethoscopes.

Dhyanendra Jain et al., (2022) for the treatment and diagnosis of pneumonia, a convolutional neural network (CNN) model has been developed. The CNN model was trained using a dataset of chest X-rays and scans. The model was created using Tensor Flow and the Keras neural network framework. The suggested modified CNN contains five Conv2D layers and layers with the most possible pooling. After undergoing Conv2D and pooling layers, the input pictures are flattened. The photos are subsequently transferred to the thick layer or completely linked layer. The network creates a forecast based on the features that are selected and extracted by each layer in this process. The suggested model contains two activation functions: a sigmoid activation function for binary classification and a rectified linear function that returns zero for negative input and values for positive input. A 96.76% accuracy rate is provided by this customized CNN model.

Benediktas Cicenas et al., (2022) presented a study that uses transfer learning and a convolutional neural network (CNN) architecture to identify pneumonia. For this study, chest X-ray images of pneumonia were employed. 5865 X-ray pictures are included in the collection. This dataset consists of three sets of data: "train," "test," and "Val." Each set has two labels on the images: "NORMAL" and

"PNEUMONIA". Datasets were equally distributed. Random zoom and random contrast techniques were used for data augmentation. Then the various CNN models were investigated. All these models are trained by adding global average pooling, dropout layers, and a single neuron dense layer. All of the models were run on Google Colab. The highest accuracy was achieved by MobileNet V2, with AUC and F1 scores of 97.83% and 93.31%, respectively. Other models had few drawbacks.

Seong Won Jo et al., (2022) examined the techniques for deep learning-based pneumonia classification utilizing images from the chest x-ray. For training and assessment, a dataset of chest X-ray images for the categorization of pneumonia was used. A grid search technique is used to test the ResNet hyper-parameters, comparing various combinations of optimizers, learning rates, momentum factors, epochs, and the number of layers. Images of different sizes are included in the dataset used for training and evaluation, and straightforward augmentation techniques are used. The hyper-parameters derived from ResNet findings are compared with various CNN models, such as VGG16, Inception V3, and DenseNet. It was discovered that DenseNet-121 performs better at classifying data with fewer layers and shorter time between epochs. The implementation of other data augmentation techniques, such as non-rigid transforms, is also explored in the proposed study. By combining these techniques with common augmentation, classification performance is increased by 5%. The experiments are carried out with CUDA and Python. The findings imply that ResNet-200 and DenseNet-121 have promising outcomes.

Rajesh Kumar Tripathy et al., (2022) proposed an effective approach for automatically detecting pulmonary disorders (PDs) from signals from the lungs, termed lung sound (LS). The models of the LS signal are evaluated using an empirical wavelet transform with specified boundary points. Shannon entropy, a time-domain property, and peak amplitude and peak frequency, a frequency-domain property, have each been assigned to each mode. In order to automatically detect PD using the properties of the LS signal, support vector machines, random forests, extreme gradient boosting, and light gradient boosting machines (LGBM) have been chosen as the classifiers. An LS signal derived from a public database has been used to evaluate the effectiveness of the proposed technique. The detection

accuracy values using the LGBM classifier with fivefold cross-validation were 80.35, 83.27, 99.34, and 77.13%. For normal, pneumonia, and asthma, the proposed method has a classification accuracy of 84.76%.

Vidhya B et al., (2022) employed digital stethoscope recordings of lung sounds to create a software program for diagnosing pneumonia. Based on lung sounds, the software successfully recognizes pneumonia patients using a gradient-boosting technique. The software uses Audacity software to process the lung sounds that doctors have captured, separating the necessary sound from extraneous noises. To train the algorithm, the audio files of healthy people are labelled as 0, while those of pneumonia sufferers are labelled as 1. The gradient boosting approach surpassed both the multilayer perceptron and convolutional neural network in terms of accuracy when its performance was compared to that of other deep learning models. The accuracy rate was about 97%. The proposed algorithm also has the potential to be used for predicting lung sounds affected by COVID-19. The proposed algorithm provides a faster, cheaper, and non-invasive method for diagnosing pneumonia.

Shreyas S. Joshi et al., (2021) created a Bluetooth-based wireless digital stethoscope. With this device, contact-free auscultation is possible. The system consists of two parts: an integrated chest-piece system that collects and wirelessly transmits the heart sound using Bluetooth, and a mobile application that receives and processes the data to record, listen to, and display the waveforms gathered. To assure end-to-end auscultation sound transmission and reception, the system employs a loop algorithm. The user makes use of smartphone app. The program calculates the average person's body mass index (BMI) and beats per minute (BPM) using the login information provided by the user. The information is kept in .wav format. As a result, the proposed prototype produces a high-quality signal in both indoor and outdoor settings without significantly changing the auscultation's diagnostic content.

Krishna More et al. (2021) extracted characteristics from chest X-ray images and classify the images to detect the existence of pneumonia in a patient, a number of deep convolutional neural network (CNN) designs have been developed. This datasets were obtained from the National Library of Medicine. Four models: VGG16, VGG19, XCEPTION, and Inception ResNet V2 have been developed using the

datasets. CNN and transfer learning were the modeling methods employed. A basic CNN numerous microscopic components known as neurons make up a neural network. A neural network is created by layering these neurons together. With the use of hyper-parameter tuning, optimization techniques are applied to every neural architecture node that contains weight. Classification and convolution are involved in training the model. This pre-trained CNN model classifies and detects pneumonia. All other models were outperformed by the VGG19 model.

Swapnil Singh et al., (2021) compared convolutional neural networks (CNN) and multilayer perceptrons, and proposed a technique for automating the detection of pneumonia using chest X-rays. Custom CNN and multilayer perceptrons are applied to the dataset obtained from Kaggle. A GUI has been developed that accepts a chest x-ray, forecasts the presence of pneumonia, and displays the proportion of congestion. CNN and multilayer perceptron model accuracy were 92.63% and 77.56%, respectively. A GUI that was developed produces a favorable outcome. CNN outperformed multilayer perceptrons in terms of performance. Hence, GUI uses CNN.

Dimpy Varshni et al., (2019) created a method for swiftly identifying Streptococcus pneumonia. The data were extracted using Kaggle. After extracting features from a variety of pre-trained CNN model iterations, DenseNet-169 is utilized. After feature extraction, a variety of classifiers, including Random Forest and Support Vector Machine, were used for the classification task. However, it was shown that the best outcomes were obtained when an SVM was used as the problem's classifier. For abnormal (pneumonia-labelled) and normal chest X-ray pictures, the customized CNN model, which combines CNN-based feature extraction with a supervised classifier algorithm, yields the best classification outcomes. This is due to the fact that DenseNets offers practical substantive characteristics before the hyper-parameter values for the SVM classifier.

Shehla Inam et al., (2019) developed a cost-effective digital stethoscope for auscultation. The digital stethoscope collects heart sounds through an electret condenser microphone. The chest piece's heart sounds are amplified and then fed via an anti-aliasing filter. The Arduino Due boards digitalize the signal, calculate BPM, and show real-time heartbeats on a TFT graphic LCD. The heart sounds are recorded in a.wav files and processed using MATLAB for frequency analysis and

SNR estimation. The proposed model is cost-effective and provides a better signal-to-noise ratio than the conventional stethoscope.

Yi Ma et al., (2019) presented a deep-learning architecture for a quick diagnosis of illnesses with a respiratory component. Whistle and crackling, two accidental breath noises, are studied. The features are extracted using wavelet analysis and the short-time Fourier transforms (STFT). A bilinear ResNet neural network has been simultaneously trained and learned to classify various adventitious lung sounds. The LungBRN application, based on the above neural network, shows the state of respiration of the subject when tested using a digital stethoscope.

Md.Riadul Islam et al., (2019) developed an electronic stethoscope that can connect to a smartphone to hear children's heart sounds. In addition to an automated process that detects anomalies in children's heart sounds. The proposed electronic stethoscope is cost-effective and bearable to use. The electronic stethoscope contains a chest component that houses the condenser microphone, which takes a recording of the heartbeat and transforms it into a digital signal. The signal is amplified and connected to an FM module, which records the heartbeat and stores the data in a mobile application. Time-frequency characteristics provide the foundation for the isolation of cardiac sound. Then the signal is processed using the Mel-Frequency Cepstral Coefficients method. A Support Vector Machine (SVM) kernel is used to further classify the signal. The suggested approach aids in the early identification of congenital cardiac conditions in children.

Adnan Hassal Falah et al., (2019) developed an approach for lung sound classification. The proposed technique integrated feature selection with support vector machines (SVM) as the classifier and stacked auto encoders (SAE) for unsupervised learning. To compare the proposed method's performance with existing feature extraction techniques utilizing discrete wavelet transforms, these techniques were also used. The proposed technique achieved an accuracy of 86.51%.

Funda Cinyol et al., (2017) examined the effect of CNN depth (number of layers) on classification accuracy in a study that classified respiratory sounds as either normal or pathological using a convolutional neural network (CNN) model. To ensure that all the data is in the same data space, respiratory sound recordings that were received from multiple sources are down-sampled to a similar sampling

frequency and normalized. The normal and diseased sounds are randomly selected for training and test data. For three separate trials, CNN's architecture has 1, 2, and 3 convolution layers, and each model has one fully connected layer after the convolution layers. The study showed that CNN architecture has great accuracy in the medical field.

W.Y. Shi et al., (2016) examined the acoustic characteristics of heart and lung sounds, a wearable wireless stethoscope was created. A stethoscope diaphragm, a PUM-5250 microphone, an operational amplifier, a wireless transceiver, and a USB wireless adaptor make up the proposed digital stethoscope. When the diaphragm of the stethoscope is placed over the chest's aortic area, the heartbeat is recorded via a microphone that is connected to the diaphragm stem through a tube. The microphone voltage signal is subsequently enhanced, wirelessly transferred, recorded, and displayed in real-time on the PC. The obtained result is represented in the time, frequency, and time-frequency domains.

Hemant Kumar Tiwari et al., (2016) created an embedded stethoscope to use with computer-aided diagnosis (CAD) of cardiac sound to find heart murmurs. The portable device can simultaneously hear real-time heartbeats and show them on a TFT LCD for visual interpretation. For offline analysis, the heart sound signal is recorded on a micro SD card. NI LabVIEW has been used to do temporal and frequency domain analysis on the recorded sound signal. The software module provides the majority of information regarding cardiovascular problems. Hence, the proposed method addresses the scarcity of specialized physicians and assist them in both urban and rural clinics and hospitals.

The Table 2.1 has 10 techniques that are more accurate than the techniques described in approach.

Table 2.1: Literature Review Comparison Chart

Reference	Methodology	Inference
Dongmin Huanget al. (8)	Bilateral pulmonary audio-auxillarymodel (BPAM) , zero crossing rate (ZCR), log Mel spectrogram, Deep learning	The specificity and sensitivity of BPAM for the diagnosis and prognosis of CAP in the subject-dependent experiment are over 92%, and they are over 50% and 39%, respectively, in the subject-independent experiment, according to the clinical validation.
Sanchit Vashisht et al. (11)	Convolutional Neural Network (CNN), Generative Adversarial Network (GAN)	After standardization, the data is divided into training and testing datasets, and the CNN model is used to train and test the model over these datasets. The accuracy of the suggested model was 98%.
Chitra R et al. (12)	Digital stethoscope, CNN model, THINKSPEAK , Pushover	The graphical representation of the detected disease is displayed in the THINKSPEAK program as the Pushover application sends a notification. Hence, this method provides an accuracy of 95% in detecting lung disease.
Dhyanendra Jain et al. (15)	Customized CNN, Multi-layer perceptron (MLP), Random Forest, Logistic Regression, Sequential minimal optimization (SMO)	There are two activation functions in the suggested model.: a rectified linear function that gives zero for negative input and values for positive input, and a sigmoid activation function for binary classification. This customized CNN model provides an accuracy of 96.76%.
Benediktas Cicénas et al. (16)	VGG16, InceptionV3, MobileNetV2, Exception, DenseNet121, ResNet50 convolutional neural network architectures.	CNN models are trained by adding global average pooling, dropout layers, and a single neuron dense layer. The highest accuracy was achieved by MobileNet V2, with AUC and FI scores of 97.83% and 93.31% respectively
Rajesh Kumar Tripathy et al. (18)	Support Vector Machine, Random Forest, Extreme Gradient Boosting, and Light Gradient Boosting Machine (LGBM)	The detection accuracy scores for the LGBM classifier with fivefold cross-validation are 80.35, 83.27, 99.34, and 77.13%. For normal, pneumonia, and asthma, the proposed approach has a classification accuracy of 84.76%.

Vidhya B et al. (19)	Gradient boosting algorithm, k- nearest neighbours (KNN) algorithm	The accuracy rate was about 97%. The proposed algorithm also has the potential to be used for predicting lung sounds affected by COVID-19. The proposed algorithm provides a faster, cheaper, and non-invasive method for diagnosing pneumonia.
Swapnil Singh et al. (22)	Deep Learning, Convolutional Neural Network, Multi-Layer Perceptron	CNN and multilayer perceptron model accuracy were 92.63% and 77.56%, respectively. A GUI that was developed produces a favourable outcome. CNN outperformed multilayer perceptrons in terms of performance
Shehla Inam et al. (24)	Digital stethoscope, Arduino DUE ,MATLAB	The heart sounds are recorded in an a.wav file and processed using MATLAB for frequency analysis and SNR estimation. The proposed model is cost- effective and provides a better signal-to- noise ratio than the conventional stethoscope
Hemant Kumar Tiwari et al. (30)	Embedded stethoscope, Arduino, NI LabVIEW	The software module provides the majority of information regarding cardiovascular problems. Hence, the proposed method addresses the scarcity of specialized physicians and assists them in both urban and rural clinics and hospitals.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

The current project aims towards developing a functional electronic stethoscope that can record and store lung sounds from people with CAP. The software program that has been developed makes it possible to identify and classify pneumonia into phases. The goal of this methodology is to diagnose pneumonia economically, promptly, and accurately.

3.2 EXISTING SYSTEM

Research indicates that numerous concepts for using chest X-rays and CT scans to identify pneumonia have been explored. These techniques mostly use CNN approach to identify pneumonia automatically. The majority of the paperprocessed data that was obtained via Kaggle. The main focus of some papers has been on comparing different deep learning processes for pneumonia detection.

3.2.1 Disadvantages

- Exposure to X-ray or other radiation by patients is harmful. And not suitable for children.
- Process of diagnosing through radiology and laboratory is not accessible in many developing countries.
- The methodology can be expensive and time consuming.

3.3 PROPOSED METHOD

The lung sound of people with pneumonia is taken using wireless stethoscope. The data collected is stored in the device. The raw data collected has to be pre-processed as it may contain artifacts. These artifacts are removed by spectral subtraction process. The Butterworth band pass filter is used to range the signal within certain (100Hz-2000Hz) frequency. Feature Extraction is done by log Mel Spectrogram. Convolutional Neural Network (CNN) methodology is used to classify the signals according to the objective.

Flow chart of the proposed method is shown in Fig. 3.1.

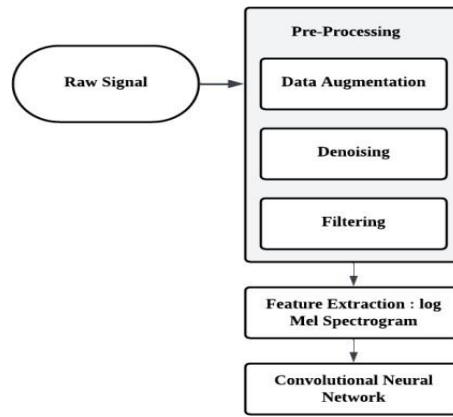


Fig 3.1 Flow Chart of Proposed Method

3.3.1 Advantages

- Electronic Stethoscope is a promising diagnostic tool for children as it is non- invasive and has got radiation free properties.
- The suggested design employed for pneumonia diagnosis is simpler, less expensive and feasible.
- This method shortens the time needed to diagnosis pneumonia.

3.4 SCOPE

This project makes it possible for medical professionals to record lung sounds using a wireless stethoscope. Thus, the proposed design can be used as an application tool for pneumonia diagnosis that is safer, less expensive, and easier than X-rays and CT scans. It also reduces the amount of time needed to detect pneumonia.

CHAPTER 4

MATERIALS AND METHODS

4.1 BLOCK DIAGRAM

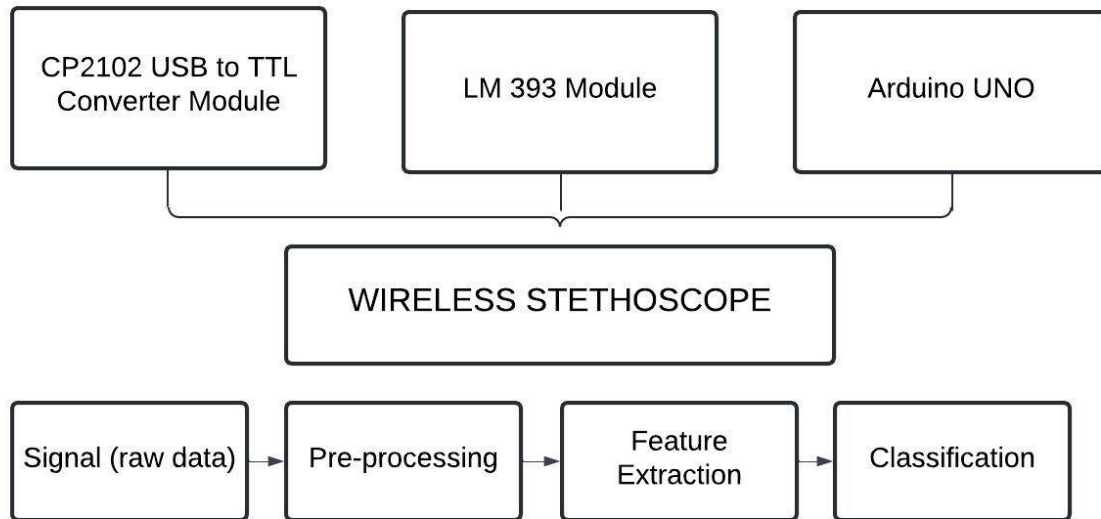


Fig:4.1 : Block Diagram of System

Fig. 4.1 shows the block diagram of system. A wireless stethoscope device intended for recording and analyzing respiratory sounds from both healthy individuals and patients with pneumonia is depicted in the block diagram.

The Arduino UNO functions as a central controller, coordinating the stethoscope's overall operation. Using a microphone, the LM393 Sound Module records respiratory sounds and transforms them into an electrical signal. The TTL to USB Module enables smooth communication between an external device, such as a computer, and the Arduino UNO by converting the TTL signal into a USB-compatible signal. Pre-processing is done on the acquired raw signal to improve its quality for additional analysis. Pre-processed signals are used to extract features; in this study, the "Log-Mel-Spectrogram" is used to characterize the respiratory sounds. In the end, CNN is used to categorize the respiratory sounds according to the qualities that were retrieved.

4.2 HARDWARE REQUIREMENTS

- Arduino Uno
- LM393 Sound Module
- TTL to USB Converter Module

4.2.1 *Arduino Uno*

In essence, the Arduino Uno board is a microcontroller board built around the ATmega 328P microcontroller. The Arduino boards are used to read sensor inputs and convert them into an output, like turning on a motor or light. The boards made by Arduino were developed in 2005 at the Interaction Design Institute in Ivrea, Italy, by David Cuatrecasas and Massimo Banzi. The Arduino Integrated Development Environment (IDE) serves as the foundation for the Arduino software. C and C++ programs are written in the IDE and uploaded to the Arduino platform.

This microcontroller board has a 16MHz ceramic resonator, 6 PWM output-capable digital I/O pins, 6 analog inputs, a USB port, a power jack, an ICSP header, and a reset button. UART, I2C, and SPI can all be used for communications. A barrel plug is the power supply connector used in Arduino UNO boards. The Arduino UNO is equipped with an ATmega16U2 16MHz USB serial processor. Arduino UNO board is shown in Fig. 4.2.

Microchip makes an 8-bit microcontroller called the ATmega 328P. 32 registers, 2KB SRAM, 1KB EEPROM, 32 KB of flash memory, and a 10-bit analog to digital converter are among its features. The RISC architecture forms its foundation. It uses little power and costs nothing. For this reason, Arduino boards employ this kind of microprocessor. The cost of Arduino boards is lower than that of microcontrollers. The Arduino software is compatible with nearly every operating system, including Linux and Windows. The Arduino IDE software is user-friendly and versatile.

Types of Arduino Board:

- Arduino UNO
- Red Board
- Lily Pad Arduino
- Arduino Mega

- Arduino Leonardo

Advantages:

- Open source
- User friendly
- Cost effective
- Portable
- Easy to learn programming
- Simplified program language to execute
- Low power consumption

Both USB and an external power source are options for powering the Arduino Uno board. An automated system selects the power source. An AC-to-DC adapter (wall-wart) or a battery can provide external (non-USB) power. Install a 2.1mm center-positive connector into the board's power jack to connect the adapter. The GND and Vin pin headers on the power connector can be fitted with battery leads. An external supply between 6 and 20 volts can power the board. The board may become unstable if the 5V pin supplies less than five volts, albeit, if less than seven volts is provided. The voltage regulator may overheat and harm the board if more than 12V is supplied. The Arduino Uno is recommended to operate between 7 and 12 volts.

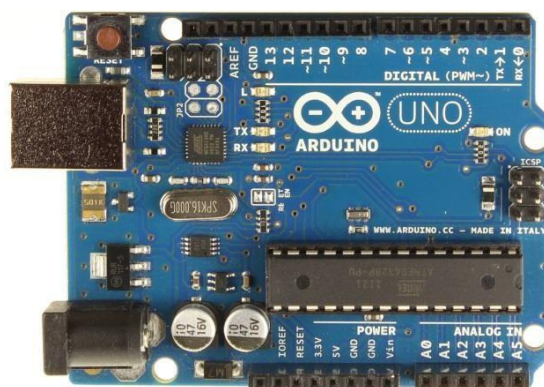


Fig:4.2 : Arduino UNO

The power connectors are as follows:

When the Arduino board is powered by an external source, the voltage applied to it

is called Vin (instead of 5 volts from the USB connection or other regulated power sources). This pin can be utilized for both providing and receiving voltage, provided that it is obtained through the power jack.

The board's regulator provides a controlled 5V to the 5V pin. The DC power jack (7–12V), USB connection (5V), or VIN pin on the board (7–12V) can all be used to power the board. You run the risk of damaging your board if you utilize the 3.3V or 5V pins to supply power instead of the regulator.

A 3.3 Volt supply, or 3V3, is produced by the on-board regulator. The maximum current draw is 50 mA.

The pins that are attached to ground are referred to as GND.

This Arduino board pin provides the voltage reference (IOREF) that the microcontroller operates on. A properly built shield can read the voltage on the IOREF pin and use that information to select the appropriate power source or activate voltage translators on the outputs to operate at either 3.3V or 5V.

The 14 digital input/output pins on the Arduino programming platform can be used as input or output pins by using the `pinMode()`, `digitalRead()`, and `digitalWrite()` methods. A maximum of 40 milliamperes can be delivered or received by each pin, which runs at 5 volts. Additionally, it has an inbuilt 20–50kOhm pull-up resistor that is disconnected by default.

The following describes the distinct responsibilities that some of the 14 pins play:

- Serial Pins 0 (Rx) and 1 (Tx) : The Rx and Tx pins are used to receive and send TTL serial data. The ATmega328P USB to TTL serial chip is used to connect them. You can configure Pins 2 and 3 of the External Interrupt Pins to generate an interrupt when a value changes, the value falls or rises, or both.
- PWM Pins 3, 5, 6, 9, and 11: These pins provide an 8-bit PWM output when the `analogWrite()` function is used.
- SPI Pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK): These pins are used for SPI communications. Integrated LEDs Pin 13: An embedded LED is attached to this pin; the LED is turned on when pin 13 is HIGH and off when

pin 13 is LOW.

In addition to the 14 digital connections, there are 6 analog input pins with a resolution of 10 bits, or 1024 potential values. They gauge between 0 and 5 volts. However, this limit can be increased by utilizing the `analogReference()` function on the AREF pin. Analog pins 4 (SDA) and 5 (SCA) are also used by the wire library for TWI communication. The Arduino Uno has a few additional pins, which are mentioned below:

- With the `analogReference()` function, AREF is used to provide a reference voltage for analog inputs.
- The microcontroller is reset by setting this pin to LOW.

Arduino can be used to communicate with other microcontrollers, computers, and Arduino boards. The ATmega328P microcontroller supports UART TTL (5V) serial communication, which can be accessed by digital pins 0 (Rx) and 1 (Tx). An ATmega16U2 on the board channels this serial communication through USB; to computer software, it appears as a virtual com port. Since the ATmega16U2 firmware makes use of standard USB COM drivers, there is no need for an additional driver. However, a .inf file is required on Windows.

The Arduino software has a serial monitor that makes it possible to send and receive simple text data to and from the Arduino board. With this library, you can use any of the Uno's digital pins for serial connection.

4.2.2 LM393 SOUND SENSOR MODULE

The LM393 Sound Sensor Module typically includes the LM393 IC along with a sound-sensitive microphone or sound sensor, as well as supporting components such as resistors, capacitors, and potentiometers, all integrated onto a PCB (Printed Circuit Board).

Key components of LM393 Sound Sensor Module:

LM393 IC: The heart of the module, providing dual voltage comparator functionality.

Sound Sensor: This is usually a small microphone or sound sensor that converts sound waves into electrical signals. The sensor's output is then fed into one of the comparators within the LM393 IC.

Potentiometer: Often included for sensitivity adjustment. It allows users to set the threshold level at which the comparator triggers an output based on the sound intensity detected by the sensor.

Supporting Components: Resistors, capacitors, and other components are used for filtering, biasing, and setting up the comparator circuitry.

Working Principle:

The LM393 Sound Sensor Module operates by detecting changes in sound intensity captured by the microphone or sound sensor. When sound is detected, the sensor converts the acoustic signal into an electrical voltage. This voltage is then compared against a reference voltage set by the potentiometer using one of the comparators within the LM393 IC. The model of LM393 sound sensor module is shown in Fig. 4.3.

If the input voltage from the sound sensor exceeds the reference voltage threshold set by the potentiometer, the comparators output switches states. This change in output state can trigger an action in the connected circuit, such as activating an alarm, triggering a recording, or any other desired response.

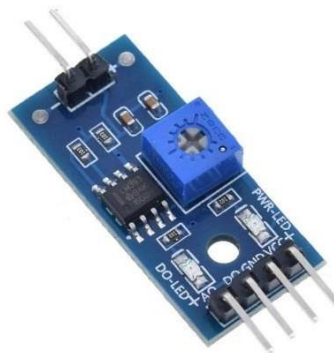


Fig:4.3 : LM393 Sound Sensor

Applications:

LM393 Sound Sensor Modules find applications in various projects and systems that require sound detection or monitoring capabilities, including:

- Sound-activated switches or devices.

- Security systems that respond to sound events.
- Home automation projects for triggering actions based on sound.
- Robotics for sound localization or response.
- Environmental monitoring for detecting noise pollution levels.
- Industrial applications for detecting machine or equipment sounds.

4.2.3 TTL to USB CONVERTER MODULE

A TTL to USB module is a device that facilitates communication between a computer (usually via USB) and other electronic devices that use TTL (Transistor- Transistor Logic) level signals. Here's a detailed overview:

Components of TTL to USB Module:

- **USB Interface:** The USB interface allows the module to connect to a computer or other USB host devices. It typically incorporates a USB Type-A connector for connecting to the host device.
- **TTL Interface:** The TTL interface is used to communicate with external devices that operate at TTL voltage levels. It usually consists of pins or connectors (such as female header pins or wires) that connect to the TTL-level signals of the target device.
- **USB to Serial Converter IC:** The core component of the USB to TTL module is a USB to Serial converter IC. Commonly used converter ICs include the CP2102, FT232RL, and CH340G. These ICs handle the conversion between USB signals and TTL-level serial signals.
- **Voltage Regulator:** Some USB to TTL modules incorporate a voltage regulator to ensure stable operation of the TTL-level signals, especially when the module is powered from the USB port.
- **LED Indicators:** LED indicators may be included to provide visual feedback on power and data transmission status.

Working Principle:

The USB to TTL module converts USB signals from the host computer into TTL-level serial signals that are compatible with electronic devices using UART (Universal Asynchronous Receiver-Transmitter) communication. Fig. 4.4 shows a

TTL to USB converter. It typically operates at common TTL voltage levels such as 3.3V or 5V.

When connected to a computer via USB, the module appears as a virtual COM port to the operating system. Software on the computer can then send and receive serial data through the virtual COM port, which is converted by the USB to Serial converter IC into TTL-level signals for communication with the external device.

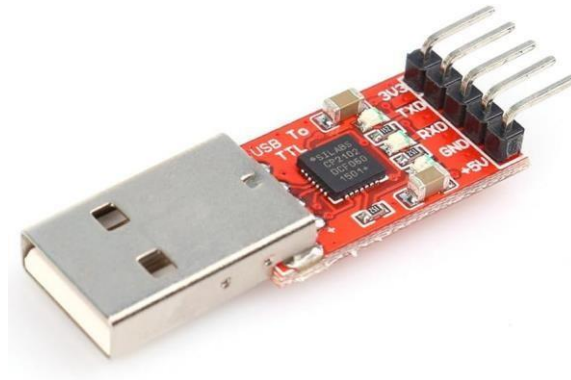


Fig:4.4 : TTL to USB Converter

Applications:

USB to TTL modules find applications in various electronic projects and systems, including:

- Programming and debugging microcontrollers (e.g., Arduino, ESP32, STM32) via UART.
- Serial communication with embedded systems, sensors, and other devices.
- Firmware updates and configuration of devices with serial interfaces.
- DIY electronics projects that require serial communication with a computer.

4.3 HARDWARE SETUP

The hardware setup of the wireless stethoscope using Arduino UNO, LM393 Module and TTL to USB Module, following steps are used:

(a) Arduino Uno Setup: The primary Micro Controller Board is the Arduino. A USB cable is used to connect the Arduino Uno to the PC. The computer has the

Arduino IDE installed. The Arduino code required for the wireless stethoscope to function has been developed.

- (b) Stethoscope head to LM393 Module: Determine the stethoscope head's output terminals. Connect the stethoscope head's output terminals to the LM393 module's input terminals. It could be necessary for you to solder both parts. Verify the connection's stability along with its polarity.
- (c) LM393 Module to Arduino Uno: Determine the LM393 module's output signal pins. It is labelled as OUT. Select an Arduino Uno digital input pin to link to the LM393 module's output. "Pin 3" is utilized in this project. Using jumper wires, join the LM393 module's output pin to the Arduino Uno's selected digital pin. Ensure that the ground (GND) pin on the Arduino Uno and the signal pin on the LM393 module are connected to each other.
- (d) TTL to USB Module to Arduino Uno: Locate the TTL to USB module's transmit (Tx) and receive (Rx) pins. Use jumper wires to link the Tx pin of the TTL to USB module to the Arduino Uno's Rx pin and the Rx pin of the TTL to USB module to the Arduino Uno's Tx pin. For reference, connect the TTL to USB module's ground (GND) pin to an Arduino Uno ground (GND) pin.

Fig. 4.5 shows the block diagram of Wireless Stethoscope.

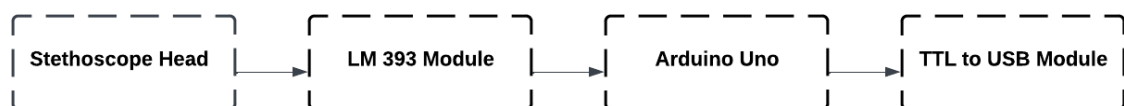


Fig:4.5 : Block Diagram of Wireless Stethoscope

Stethoscope Head: This is where the sound is captured.

LM393 Module: It amplifies the signal received from the stethoscope head.

Arduino Uno: Controls the entire system and process the signal.

TTL to USB Module: It provides communication between the Arduino Uno and Computer via USB.

4.4 CIRCUIT DIAGRAM

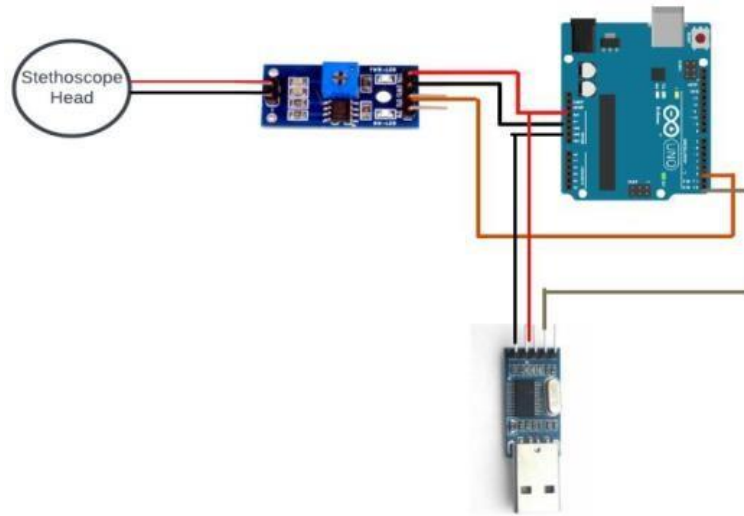


Fig:4.6 : Circuit Diagram

Circuit diagram of Wireless Stethoscope is given in Fig. 4.6.

Stethoscope Head to LM393 Module: The stethoscope head is connected to the input of the LM393 module. The sound waves collected by the stethoscope head generate weak electrical signals, which are fed into the LM393 module for amplification.

LM393 Module to Arduino Uno: The output of the LM393 module, which carries the amplified respiratory signals, is connected to a digital pin 'D2' of the Arduino Uno. The ground (GND) pin of the LM393 module is connected to the ground (GND) pin of the Arduino Uno. The 5V pin of the LM393 module is connected to the 5V pin of the Arduino Uno.

Arduino Uno to TTL to USB Module: The transmit (TX) pin of the Arduino Uno is connected to the receive (RX) pin of the TTL to USB module. The ground (GND) pin of the Arduino Uno is connected to the ground (GND) pin of the TTL to USB module. The 5V pin of the Arduino Uno is connected to the 5V pin of the TTL to USB module. The working model of wireless stethoscope is shown in Fig. 4.7.

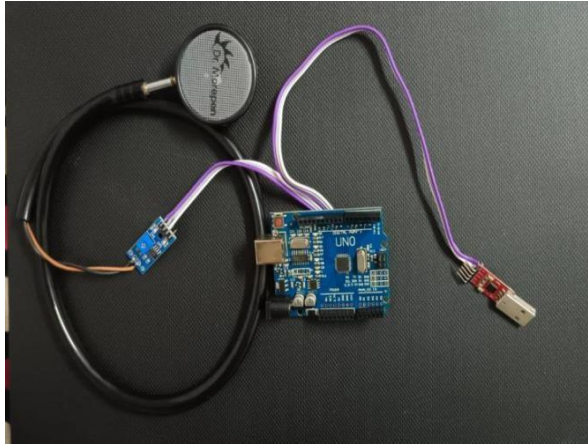


Fig:4.7 : Wireless Stethoscope

Operation:

Sound Collection: The stethoscope head is placed on the patient's chest or back to capture respiratory sounds, which are converted into weak electrical signals.

Signal Amplification: The LM393 module amplifies the weak signals received from the stethoscope head, enhancing their amplitude for effective processing by the Arduino Uno.

Signal Processing: The Arduino Uno processes the amplified respiratory signals received from the LM393 module. It analyses the signals, performs necessary computations, and prepares the data for transmission.

Data Transmission: The processed respiratory data is transmitted from the Arduino Uno to external devices via the TTL to USB module. The TTL to USB module converts the TTL signals into USB-compatible signals, allowing seamless communication with computers or other devices.

4.5 SOFTWARE REQUIREMENTS

Arduino IDE

Google Colab

4.5.1 Arduino IDE

Arduino IDE is an open-source software that helps to write and upload the code to Arduino board. The Arduino Integrated Development Environment (IDE) is a software platform used for writing, compiling, and uploading code to Arduino-compatible microcontroller boards.

Purpose: The Arduino IDE provides a user-friendly interface for programming Arduino boards. It abstracts away many of the complexities of embedded programming, making it accessible to hobbyists, students, and professionals alike.

Cross-Platform: The Arduino IDE is available for multiple operating systems including Windows, macOS, and Linux, making it accessible to a wide range of users.

Open Source: The Arduino IDE is open-source software, meaning that its source code is freely available for modification and redistribution under the GNU General Public License (GPL).

Code Editor: The IDE includes a text editor where users can write their Arduino sketches (programs). It provides features like syntax highlighting, auto-indentation, and code completion to assist programmers.

Sketches: In Arduino terminology, programs are referred to as "sketches." A sketch typically consists of two main functions: `setup()` and `loop()`. The `setup()` function is executed once when the program starts, while the `loop()` function is executed repeatedly for as long as the Arduino board is powered on.

Library Manager: The Arduino IDE includes a Library Manager, allowing users to easily add third-party libraries to their projects. Libraries contain pre-written code that can be used to interface with various hardware components and sensors, saving time and effort in writing code from scratch.

Board Manager: The Board Manager enables users to install support for different types of Arduino-compatible boards, including official Arduino boards as well as third-party variants.

Serial Monitor: The IDE features a built-in Serial Monitor tool that allows users to communicate with their Arduino board via the serial port. This tool is useful for debugging purposes and for sending/receiving data between the Arduino board and a connected computer.

Compilation and Upload: The Arduino IDE compiles the sketch into machine code that can be understood by the microcontroller on the Arduino board. It then uploads the compiled code to the board via a USB connection or other supported interfaces.

Extensibility: The Arduino IDE can be extended with additional features and

functionalities through plugins and extensions, allowing users to customize their development environment according to their specific needs.

4.5.1.1 To Download Software

Step 1: First, launch Internet Explorer and look for the Arduino IDE.

Step 2: Select the link that the search page's top displays. [www.arduino.cc]

Step 3: On that webpage, select Software.

Step 4: Click the Arduino IDE 2.0.4 tab and select the appropriate download options based on your system model.

Step 5: Select “Just Download”, and the download begins in the background.

Step 6: Go to the location of the downloaded file.

Step 7: The file begins to download when you click on it.

Step 8: Select the storage path, grant the necessary permissions, and install it.

4.5.1.2 To Create and Run Program

Step 1: Launch the Arduino IDE.

Step 2: Click New Sketch after opening the file.

Step 3: From the Menu bar, open the Tools and choose Arduino UNO as the Port.

Step 4: Type the code into the workspace and press the Enter key.

Step 5: Check and compile the code.

Step 6: To view the graphical representation, select Serial Plotter from the menu bar, and to monitor the data received as output, click Serial Monitor from the Tools menu bar.

4.5.2 Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based platform provided by Google that allows users to write and execute Python code in a browser-based environment, without requiring any setup or installation. It is particularly popular among researchers, educators, and data scientists for its ease of use, accessibility, and integration with Google Drive.

4.5.2.1 To Create and Run

Step 1: Access Google Colab

- Open your web browser and go to Google Colab.
- Sign in with your Google account if you're not already signed in.

Step 2: Create a New Notebook:

- Once you're logged in, you'll see the Colab dashboard.
- Click on the "New Notebook" button to create a new notebook.
- This will open a new untitled notebook where you can start writing your code.

Step 3: Create a New Notebook:

- Once you're logged in, you'll see the Colab dashboard.
- Click on the "New Notebook" button to create a new notebook.
- This will open a new untitled notebook where you can start writing your code.

Step 4: Write Python Code:

- In a code cell, you can write Python code just like you would in a local Python environment.
- Type or paste your Python code into the code cell. Type or paste your Python code into the code cell.

Step 5: Execute Code Cells:

- To execute a code cell, click on the play button (triangle icon) on the left side of the cell.
- Alternatively, you can use the keyboard shortcut "Shift + Enter" to execute the cell. When you run a code cell, Colab sends the code to Google's servers, executes it, and displays the output below the cell.

Step 6: Add Markdown Text:

- In a text cell, you can write documentation, explanations, and formatted text using Markdown syntax.
- Markdown allows you to add headings, lists, links, images, and more to

your notebook.

- Write your text content using Markdown syntax in the text cell.

Step 7: Execute Markdown Cells:

- Text cells don't need to be executed like code cells.
- They are rendered immediately after you finish writing them.

Step 8: Save and Name Your Notebook:

- To save your notebook, click on "File" in the menu bar and then select "Save" or press "Ctrl + S" (Cmd + S on Mac).
- You can also give your notebook a meaningful name by clicking on "Untitled" at the top of the notebook and entering a name.

Step 9: Share and Collaborate:

- You can share your Colab notebook with others by clicking on the "Share" button at the top right corner of the notebook.
- You can share the notebook via link or invite collaborators by entering their email addresses.

4.6 ABOUT THE DATASET

Audio recordings were gathered from individuals with pneumonia and healthy individuals across different age groups. The recordings range in duration from 10 to 90 seconds. A total of 3800 respiratory cycles are captured in 1.077 hours of recordings. All age categories of patients are included: adults, children, and the elderly. The neural network model would classify or predict the respiratory sounds in the data set into several categories, such as Healthy, Mild Pneumonia, and Severe Pneumonia. The respiratory sounds from a healthy individual are included in the "healthy category", whereas the sounds from patients with certain medical disorders are included in the other categories.

4.6.1 Data Annotation

Data annotation refers to the process of labeling or tagging data with relevant information, such as categories, classes, or attributes, to make it understandable

and usable by machine learning algorithms. In the context of data classification using Convolutional Neural Network (CNN) models, data annotation involves assigning labels or categories to images or other types of data to train the CNN for classification tasks.

Here, Patient ID, Gender, Age and Condition which is shown in Table 4.1.

Table 4.1: Data Annotation

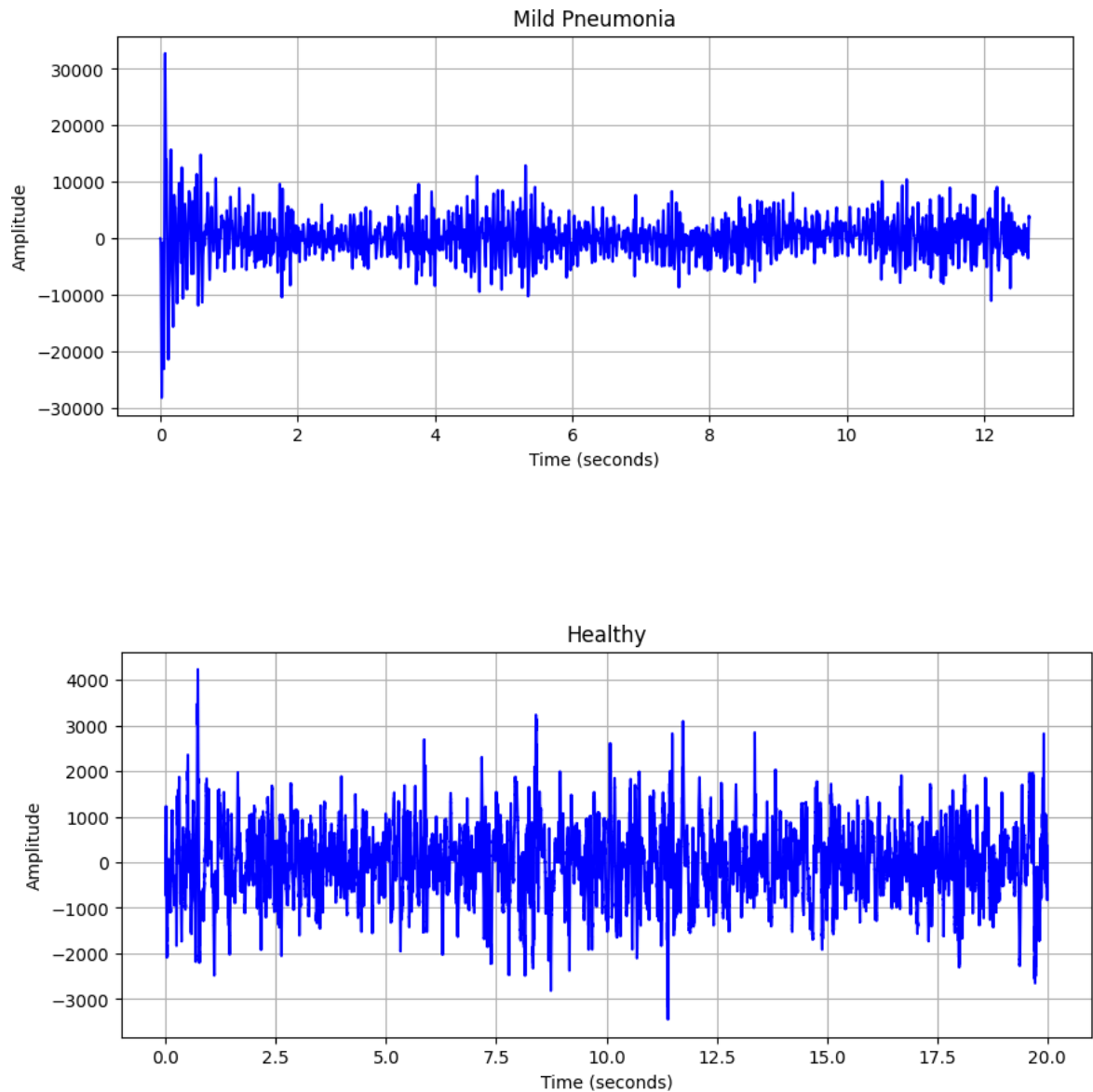
Patient ID	Age	Gender	Condition
100	75	Male	Severe Pneumonia
101	52	Female	Severe Pneumonia
102	30	Male	Severe Pneumonia
169	18	Female	Healthy
170	25	Female	Healthy
171	45	Male	Healthy
240	62	Male	Mild Pneumonia
241	07	Female	Mild Pneumonia
242	37	Male	Mild Pneumonia

4.7 TOOLS USED FOR IMPLEMENTATIONS

Python is the programming language utilized in this project to implement the model. Librosa package is used for feature extraction from the audio data. The matplotlib library is used to visualize data. TensorFlow is utilized in the backend of Keras, a Python deep learning toolkit, to build the deep neural network.

4.8 VISUALIZATION OF SOUND FILES OF DIFFERENT CATEGORY

For this project, the wave plot is displayed and the audio files are loaded using Librosa. The wave plot representations for each category of the sound file utilized in this project are shown in Fig. 4.8.



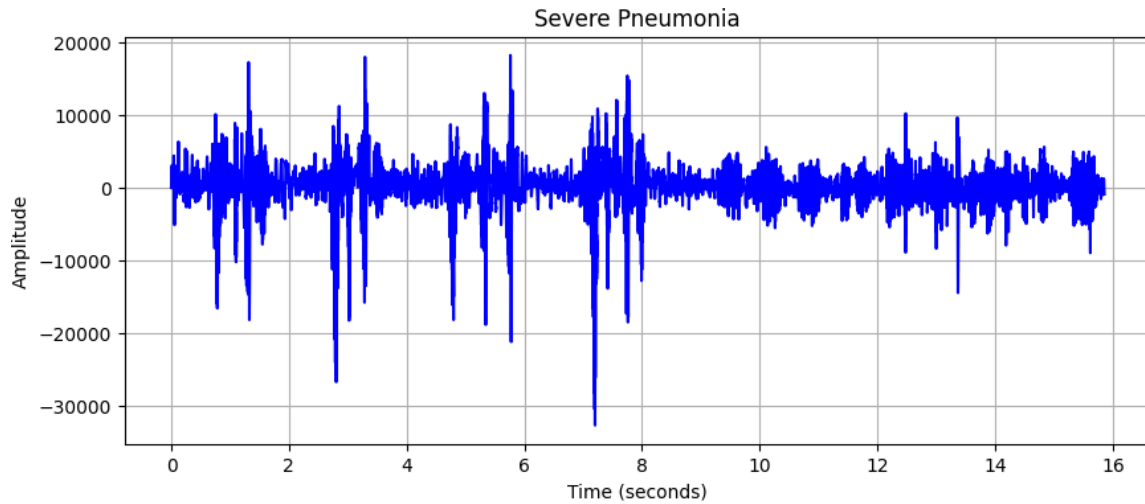


Fig:4.8 : Wave plot of each category

4.9 DATA PRE-PROCESSING

4.9.1 Data Augmentation

When it comes to CNN classification, dealing with a dataset that has only three classes and a low count is really difficult. When there is less data, there is a greater chance of overfitting, which occurs when the model learns to memorize the training samples instead of adapting well to new data. Data augmentation was used as a strategic way to address this problem. In this instance, during the augmentation procedure, Background noise with a noise factor of 0.05 and 0.10 was added to each of the three classes. The CNN model is exposed to a wider variety of training instances by implementing noise augmentation, simulating real-world circumstances where data may be noisy and variable. In addition to enriching the dataset, this augmentation strategy aids in the model's learning of robust and generalized features—a prerequisite for precise classification. Therefore, in the face of the difficulty presented by the little dataset, data augmentation becomes an important tool for improving the CNN's classification performance, allowing it to function well even in difficult situations. Data distribution of data after data augmentation is shown in Fig. 4.9.

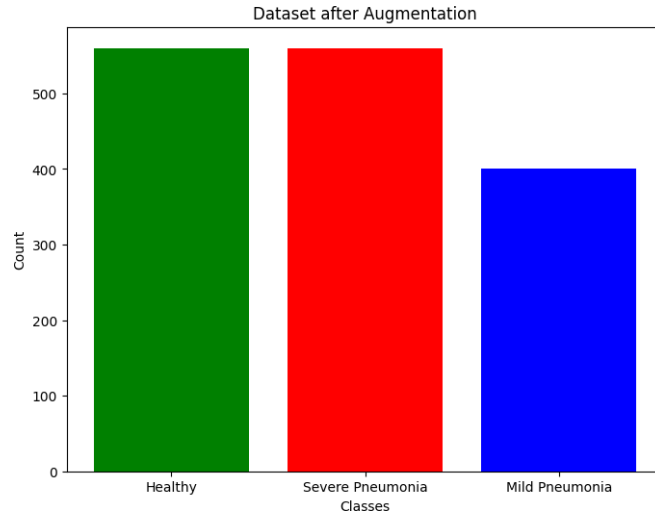


Fig:4.9 : Data Distribution after Data Augmentation

4.9.2 Denoising

A wireless stethoscope's ability to detect respiratory audio signals is essential for the diagnosis of respiratory conditions. Nevertheless, a variety of noises, including as ambient noise, movement artifacts, and interference from medical equipment, frequently distort the recorded data. The respiratory dataset used in this project comprises recordings from both healthy individuals and patients with various range of pneumonia. The gathered dataset includes a variety of noise environments that are commonly seen in healthcare settings. Noise can distort important breathing patterns and impair the precision of diagnosis. The implementation of the denoising process becomes essential in response to these difficulties. A widely utilized method in audio processing called spectral subtraction presents a viable way to reduce noise in respiratory audio signals. Spectral subtraction is a technique used to improve the clarity and diagnostic usefulness of respiratory recordings by calculating and subtracting the noise spectrum from the original signal spectrum. The following is the methodology used in Spectral Subtraction:

Signal Representation: Generally, methods such as the Fast Fourier Transform (FFT) are used to represent the audio signal in the frequency domain. This releases the signal's spectral content by breaking down the time-domain signal into its frequency components.

Noise Estimation: A portion of the signal that is either estimated or caught only when background noise (and no desired signal) is present. The appropriate length for this

noise-only segment is to capture the statistical characteristics of the background noise without a discernible presence of the intended signal. The frequency-domain representation of this noise-only section is obtained by applying the FFT to it. This illustrates the background noise's spectrum properties.

Signal and Noise Spectra Calculation: The frequency-domain representation of the noisy signal is obtained by applying the FFT to the full signal. For every frequency bin, the amplitude spectrum of the noisy signal is subtracted from the amplitude spectrum of the noise-only section. This is how spectral subtraction works fundamentally.

Adjustment for Negative Values: In some frequency bins, the amplitude that results from subtraction may occasionally be negative. This is due to the possibility that the noise spectrum estimation may not accurately reflect the noise present in the signal. A floor value or minimum threshold is frequently added to the result following subtraction in order to prevent negative numbers.

Inverse FFT: The inverse FFT is used to convert the signal's altered frequency-domain representation—which has been subtracted—back into the time domain. In doing so, the altered time-domain signal is retrieved.

Overlap and Add: To prevent artifacts caused by the FFT and inverse FFT processes, the signal is often split into overlapping segments in various implementations. The denoised audio signal is reconstructed by overlapping and adding each segment after it has been processed. Fig. 4.10 shows the block diagram of spectral subtraction method.

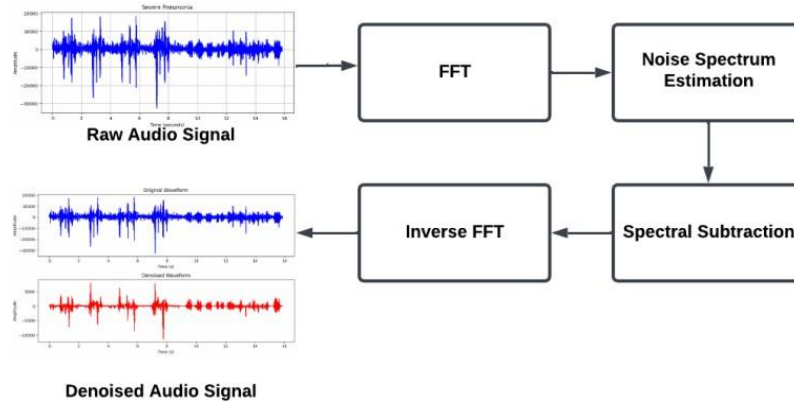


Fig:4.10 : Block diagram of Spectral Subtraction

This project uses the NumPy library and Python to implement Spectral Subtraction for numerical computations. Signal obtained after denoising is shown in Fig. 4.11.

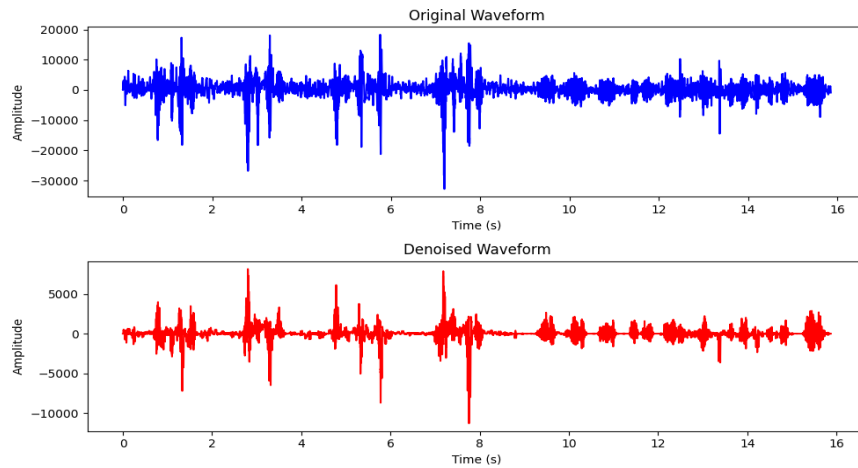


Fig:4.11 : Signal obtained after Denoising

4.9.3 Filtering Process

Following the removal of artifacts, a 4th order Butterworth band pass filter with cut-off frequencies of 100 Hz and 2000 Hz is used to further treat the respiratory sounds that were collected. With a wireless stethoscope, the 4th order Butterworth band pass filter aids in maintaining relevant respiratory properties during recording. The methodology involved in 4th order Butterworth band pass filter is mentioned below.

- **Input Signal (Respiratory Sounds):** This is the initial raw signal containing respiratory sounds. It's typically a time-domain signal sampled at a certain rate.

- Normalized Cutoff Frequencies (Low, High): The cutoff frequencies are normalized to the Nyquist frequency, which is half the sampling rate. This normalization ensures that the cutoff frequencies are expressed relative to the maximum frequency that can be represented in the digital signal.
- Butterworth Filter Coefficients (b, a): The normalized cutoff frequencies (Low, High) along with the desired filter order (4th order in this case) are used to design the Butterworth filter. The Butterworth filter design algorithm generates coefficients b and a, which represent the numerator and denominator coefficients of the filter transfer function, respectively.
- Apply Filter Coefficients: The designed filter coefficients (b, a) are applied to the input signal using the `filtfilt` function from the SciPy library. `Filtfilt` applies the filter forward and backward to achieve zero-phase filtering, which means that it minimizes phase distortion in the filtered signal. Applying the filter coefficients to the signal effectively filters out frequencies outside the specified bandpass range while preserving the frequencies of interest within the passband.
- Filtered Respiratory Sounds: The output of the filtering process is the denoised respiratory sounds. These sounds have been processed through the bandpass Butterworth filter, resulting in attenuation of unwanted frequencies (such as noise) while retaining the frequencies of interest within the specified passband.

In this project, 'Butter' and 'filtfilt' are imported for signal processing functions, and NumPy is used to create the Butterworth band pass filter in this project. The fourth order Butterworth band pass filter is being employed here. The frequency range of the cutoff is 100Hz–2000Hz. Fig. 4.12 shows the block diagram of butterworth bandpass filter.

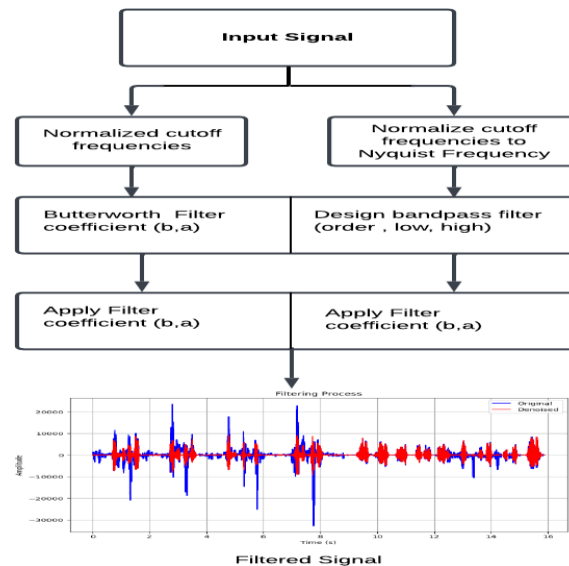
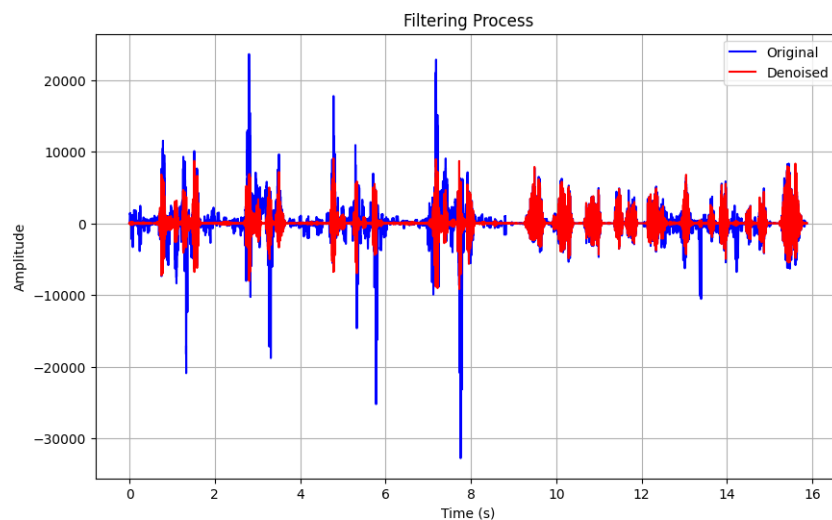


Fig:4.12 : Block diagram of Butterworth bandpass filter

The filtered signal obtained is shown in Fig. 4.13.



Fig;4.13 : Signal obtained after Filtering

4.9.4 Power Spectrum

The power spectrum of an audio signal represents how the power of the signal is distributed across different frequencies. The power spectrum of an audio signal is a fundamental tool for understanding, analyzing, and processing audio data in various domains, including music, speech, telecommunications, and multimedia. It serves as a basis for numerous audio-related applications and contributes to

advancements in audio technology and research. Importance of power spectrum includes:

- **Frequency Analysis:** The power spectrum allows us to analyze the frequency components present in an audio signal. By examining the power distribution across different frequencies, we can identify dominant frequencies, harmonics, and other spectral characteristics of the signal.
- **Feature Extraction:** In applications such as speech recognition, music genre classification, and sound recognition, features extracted from the power spectrum are commonly used. These features capture important information about the spectral content of the audio signal, which can be utilized by machine learning algorithms for classification and identification tasks.
- **Equalization and Audio Processing:** Understanding the power spectrum of an audio signal is essential for audio processing tasks such as equalization, filtering, and noise reduction. By analyzing the power spectrum, one can identify specific frequency bands that may need adjustment or enhancement to achieve desired audio quality.
- **Visualization and Interpretation:** Visualizing the power spectrum provides insights into the frequency composition of the audio signal. It helps audio engineers, musicians, and researchers understand the characteristics of the sound, identify potential issues, and make informed decisions during audio production and analysis.
- **Compression and Coding:** In audio compression algorithms such as MP3 and AAC, knowledge of the power spectrum is utilized to efficiently represent and encode audio signals. By focusing on perceptually important frequency components while discarding less significant ones, compression algorithms can achieve high compression ratios with minimal loss of perceived audio quality.

The power spectrum plot representations for each category of the sound file utilized in this project are shown in Fig. 4.14.

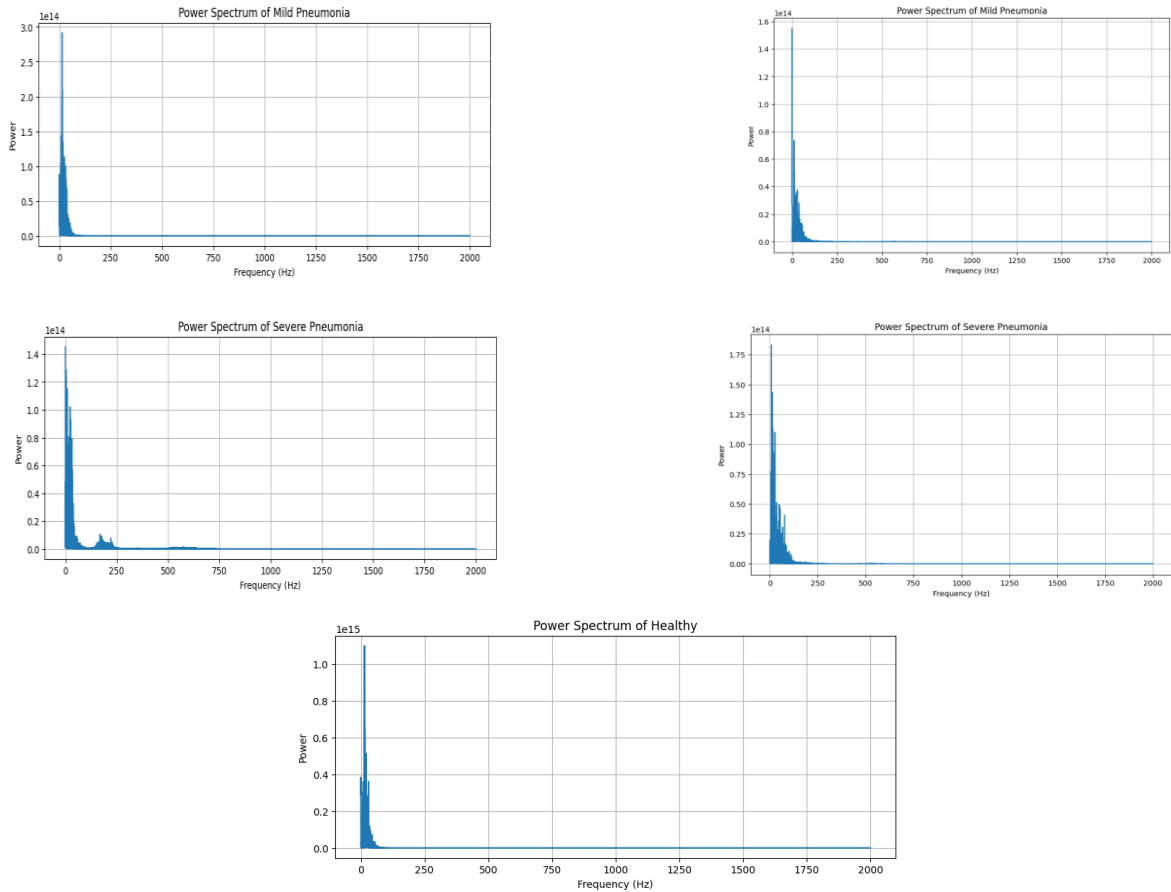


Fig:4.14 : Power Spectrum of each category

4.10 FEATURE EXTRACTION

The log is the predominant component of lung sound is in the low frequency region and the Mel Spectrogram can simulate the nonlinear human auditory system, it is utilized to quantify temporal frequency variations. The sample's amplitude is first normalized. A mathematical technique called the Fourier Fast Transform (FFT) is used to translate a signal from the time domain into its frequency domain representation. When used in audio processing, the FFT calculates the relative intensities of various frequency components within a window of the audio stream. Fig. 4.15 shows the block diagram of log Mel Spectrogram. Mel filter bank is applied to the output spectrum following FFT computation. The FFT spectrum is multiplied by each of the triangular filters in the filter bank to achieve this.

$$\text{Mel (f)} = 2595 \log_{10} (1 + f/700)$$

The amplitude spectrum of the lung sound sample obtained by FFT is multiplied by

each Mel filter and added together based on the aforementioned actions.

A set of values indicating the energy contained in each frequency range defined by the filter bank is the result of this multiplication. A log function is used to alter the resulting Mel spectrogram. This is due to the logarithmic aspect of how loudness is perceived by humans. Normalization of the log Mel Spectrogram to decibels (dB) is done to guarantee uniformity among the audio signals. The log Mel Spectrogram can be thought of as a 2-D representation of an audio signal, with the color intensity signifying the log energy or power within each frequency bin at each time frame, and the x- and y-axes representing frequency (Mel scale).

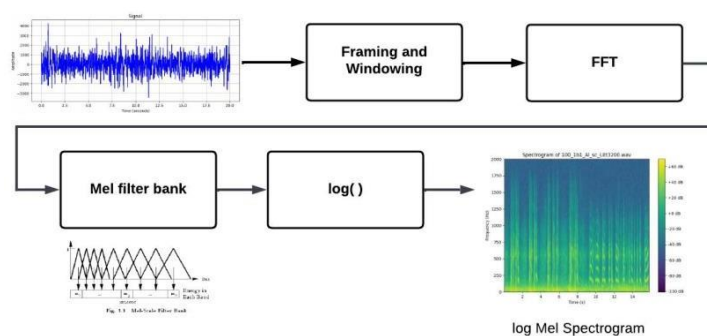


Fig:4.15:Block diagram of log Mel Spectrogram

The methodology of log mel spectrogram using the Fast Fourier Transform (FFT) and inverse FFT involves a series of steps to transform audio signals into a spectrogram representation that emphasizes the frequency components most relevant to human auditory perception.

Framing and Windowing: The first step involves segmenting the audio signal into short overlapping frames. Each frame typically ranges from 20 to 50 milliseconds and is windowed using a window function such as the Hamming window to reduce spectral leakage. The windowed frames are then passed through the FFT algorithm to compute their frequency-domain representations. The FFT decomposes each frame into its constituent frequency components, providing a snapshot of the signal's frequency content over time.

Fast Fourier Transform (FFT): From the complex-valued FFT output, the magnitude spectrum is computed by taking the magnitude of each frequency bin. The magnitude spectrum represents the distribution of signal energy across different frequency bands at each time frame.

Mel Filterbank: Next, the magnitude spectrum is multiplied element-wisewith a

bank of Mel filters. The Mel filterbank consists of a series of triangular filters spaced evenly on the Mel frequency scale, which is a perceptually meaningful scale of pitch. Each Mel filterbank filter is designed to capture the energy within a specific frequency range corresponding to human hearing sensitivity. The filters are applied to the magnitude spectrum to compute the energy within each frequency band.

Logarithmic Compression: The filterbank energies are then subjected to logarithmic compression, typically using the natural logarithm or base 10 logarithm. Logarithmic compression serves to compress the dynamic range of the filterbank energies and enhances the representation of low-energy components while mitigating the impact of outliers.

Inverse Operations: To reconstruct the log Mel spectrogram, the process is reversed using the inverse operations. The inverse DCT is applied to the compact representation to obtain the log filterbank energies. The exponential function is applied to reverse the logarithmic compression, yielding the filterbank energies. The inverse Mel filterbank is applied to reconstruct the magnitude spectrum. Finally, the inverse FFT is applied to reconstruct the time-domain representation of the audio signal.

The resulting log mel spectrogram is a two-dimensional representation of the audio signal, where the x-axis represents time, the y-axis represents frequency on the Mel scale, and the colour intensity represents the log energy level of each frequency band at each time frame. log Mel spectrograms are commonly used as input features for machine learning models in various audio processing tasks due to their ability to capture salient spectral characteristics of audio signals in a perceptually meaningful manner.

The log Mel Spectrogram extracted from each category of the audio file utilized in this project are shown in Fig. 4.16.

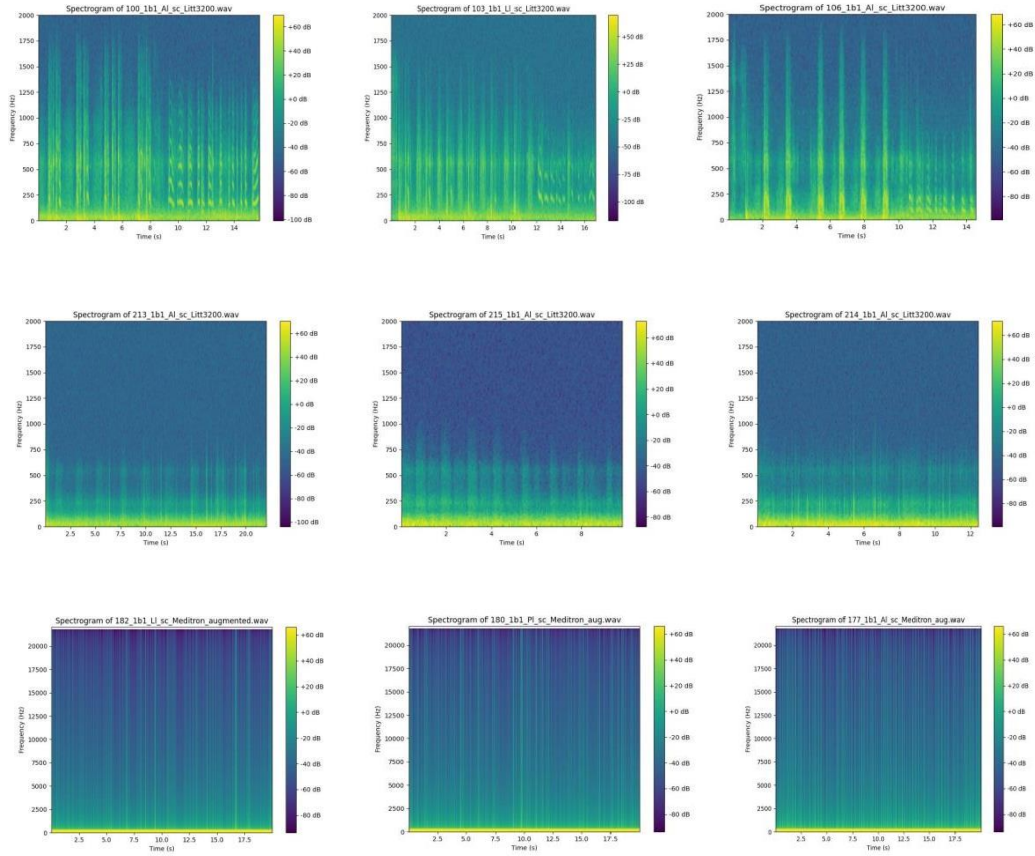


Fig:4.16 : log Mel Spectrogram of each category

4.11 CONVOLUTIONAL NEURAL NETWORK (CNN)

4.11.1 Implementation of CNN Model

A Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed for processing structured grid-like data. Originally developed for computer vision tasks, CNNs have since been applied to various other domains, including audio processing, natural language processing, and medical image analysis. Convolutional Neural Networks (CNNs) play a crucial role in audio classification tasks due to their ability to effectively capture hierarchical features from audio data. Here are some key reasons why CNNs are important in audio classification:

- *Hierarchical Feature Learning:* CNNs are adept at learning hierarchical representations of audio features. They automatically extract low-level features such as edges, textures, and spectral patterns in the early layers, while higher layers learn more abstract and discriminative features relevant

to the classification task. This hierarchical feature learning enables CNNs to capture complex patterns and variations in audio signals, making them well-suited for audio classification tasks.

- *Translation Invariance:* CNNs exhibit translation invariance, meaning they can identify patterns irrespective of their location in the input data. In audio classification, this property allows CNNs to recognize relevant features regardless of their temporal position within the audio signal. For example, CNNs can identify characteristic sound patterns associated with specific events or classes regardless of when they occur within the audio clip.
- *Parameter Sharing:* CNNs leverage parameter sharing across different regions of the input data, which helps reduce the number of parameters and computational complexity compared to fully connected networks. This parameter sharing enables CNNs to efficiently learn spatial and temporal patterns in audio signals while maintaining model compactness and scalability.
- *Local Receptive Fields:* CNNs use local receptive fields to capture spatial and temporal dependencies within the input data. By focusing on small, overlapping regions of the audio signal, CNNs can capture short-term and long-term temporal patterns, enabling robust classification performance across various audio genres, languages, and acoustic environments.
- *Pooling Layers:* CNNs often incorporate pooling layers, such as max pooling or average pooling, to downsample feature maps and enhance translation invariance. Pooling layers help reduce the spatial dimensionality of feature maps while preserving their salient features, facilitating robust feature extraction and improving model generalization.
- *Regularization Techniques:* CNNs can leverage various regularization techniques, such as dropout, batch normalization, and weight regularization, to prevent overfitting and improve model generalization. These techniques help CNNs learn more robust and discriminative representations of audio features, leading to improved classification accuracy on unseen data.
- *End-to-End Learning:* CNNs support end-to-end learning, allowing them to directly map raw audio waveforms to class labels without relying on

handcrafted feature extraction methods. This end-to-end learning paradigm enables CNNs to learn task-specific representations directly from the input data, potentially capturing subtle nuances and contextual information that may not be evident in manually engineered features.

The CNN Model can be trained once the log Mel Spectrogram has been extracted. Five different types of layers are used in the creation of the proposed neural network model: convolutional layers, pooling layers, flatten layers, dense layers, and dropout layers. The detailed functions of each layer in our model are provided in Fig. 4.17.

Convolutional Layers: 'Conv2D' Convolutional layers perform convolution operations on the input data, extracting features using learnable filters. These layers are essential in extracting spatial hierarchies of features from input images. In our model, seven convolutional layers ('Conv2D') are used.

Pooling Layers: 'MaxPooling2D' Pooling layers reduce the spatial dimensions of the input data by selecting the maximum value within a specified window. This helps reduce computational complexity and control overfitting by focusing on the most important features. In our model, four MaxPooling layers ('MaxPooling2D') are used.

Flatten Layer: 'flatten' The flatten layer converts the multi-dimensional output from the previous layers, which are convolutional layers and pooling layers, into a one-dimensional array, which can be fed into the fully connected layers. In our model, one flattening layer is used.

Dense Layer: 'Dense' Dense layers are fully connected neural network layers where each neuron is connected to every neuron in the previous layer. These layers help in learning complex patterns in the data and performing classification or regression tasks. In our model, two dense layers have been used. First dense layer is with 128 neurons. Each neuron in this layer is connected to every neuron in the previous layer, which could be the output of a flattening layer. Here, the '**relu**' [**Rectified Linear Unit**] activation function is applied to the output of this dense layer. It introduces non-linearity into the network and helps the network learn complex patterns in the data. There are three neurons in the second dense layer. The number of classes in this layer is often represented by the number of neurons, which in this case is '3'. This dense layer's output is subjected to the '**softmax**' activation function. The neural network's raw output scores are transformed into

probabilities using softmax. It is appropriate for multi-class classification since it guarantees that probabilities in every class equal 1.

Dropout Layer: The dropout layer is used for regularization. It is a regularization technique applied during training that randomly sets a fraction of input units to zero, helping to prevent over fitting. The dropout rate between every two dense layers is 0.2.

In summary, the convolutional layers extract features from the input data, the max pooling layers reduce spatial dimensions, the dropout layer helps prevent overfitting by randomly dropping units during training, the flatten layer prepares the data for the fully connected layers, and the dense layers perform classification based on the extracted features. Together, these layers form a convolutional neural network (CNN) architecture suitable for classification tasks. The summary of the CNN Model is shown in Fig 4.18. ***Total params and trainable params obtained are 91683, and non-trainable params are 0.***

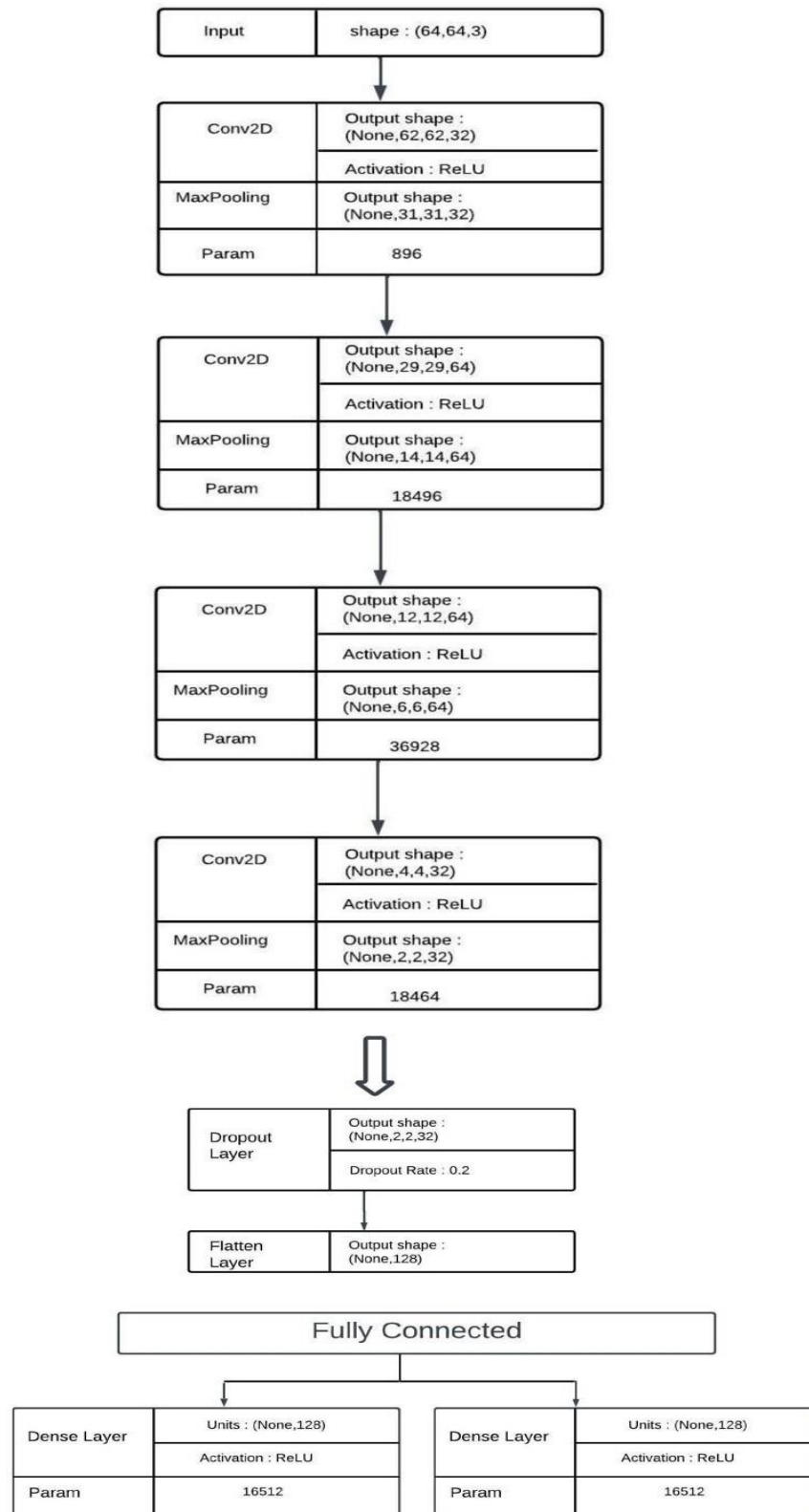


Fig:4.17 : Architecture of CNN Model

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_3 (Conv2D)	(None, 4, 4, 32)	18464
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 32)	0
dropout (Dropout)	(None, 2, 2, 32)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 128)	16512
dense_1 (Dense)	(None, 3)	387
Total params: 91683 (358.14 KB)		
Trainable params: 91683 (358.14 KB)		
Non-trainable params: 0 (0.00 Byte)		

Fig:4.18 : Summary of CNN Model

4.11.2 Compiling of CNN Model

The loss function employed in the model compilation process is categorical cross-entropy. Adamax has also been applied as an optimizing algorithm, or optimizer.

Optimizer ('adam' is the optimizer): The model's weight updates during training are decided by the optimizer. The adaptive learning rate optimization algorithm known as the **"adam"** optimizer combines the benefits of the **AdaGrad** and **RMSProp** algorithms. Adam enhances training efficiency and convergence by dynamically adjusting the learning rate for every parameter.

Loss Function ("loss" = "categorical_crossentropy"): It measures how well the model performs on the training data and guides the optimization process. **"categorical_crossentropy"** is used for multi-class classification. It calculates the difference between the predicted probability distribution and the true distribution of the labels using the cross-entropy formula. The goal during training is to minimize this value.

Metrics (metrics = "accuracy"): During training and assessment, metrics are used to track the model's performance. In this case, the parameter called **"accuracy"** is defined as the percentage of correctly identified. To sum up, model compilation is an essential phase in the training process that establishes the model's learning and performance assessment mechanisms.

4.11.3 Generating the CNN Model

After compiling, the next process is visualization and analysis of the model using model generators. For this, `Model.fit_generator` has been used. The "`Model.fit_generator`" function in Keras is used to train a deep learning model using data generated batch by batch by a Python generator. We have adopted this, as it is useful when working with large datasets that cannot fit into memory all at once. Steps per epoch are set to the total number of training samples divided by the batch size. The epochs parameter defines the number of times the model will iterate over the entire training dataset. Here, 50 epochs are used. During each epoch, the model's parameters are updated based on the optimization algorithm specified during compilation.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 MODEL EVALUATION

The acquired log Mel spectrogram images are used to train the deep learning model on the data. First, an 80-20 split ratio is used to divide the patient data into training and testing sets. The "Image Data Generator" class from Keras is used to pre-process image data. Real-time data augmentation and standardization are possible with this class.

The "data frame" approach is used to construct the training set, and log Mel Spectrogram images are loaded from the training directory. A 75-25 split ratio is used for training and validation, respectively, to further divide the training set into training and validation subsets. Out of total 1520 datasets 912 datasets are used for training, 304 datasets are used for validation and 304 datasets are used for testing. Distribution of datasets are shown in Fig. 5.1. The pixel values are rescaled to fall between $[0, 1]$ and the images are resized to a target size of 64x64 pixels. Resizing images is done because deep learning models often require fixed-size inputs. By resizing images to a consistent size, we ensure that all images fed into the model have the same dimensions, which simplifies the computational process. In this model, the images are resized to 64x64 pixels, likely to balance computational efficiency and information retention. Rescaling pixel values to the range $[0, 1]$ is done because neural networks are sensitive to the scale of input data. Rescaling pixel values to a range between 0 and 1 standardizes the input data, making it easier for the model to learn patterns and relationships in the data. It also prevents issues related to exploding gradients during training, where large values could cause numerical instability. Rescaling pixel values is achieved by dividing the original pixel values by the maximum pixel values.

The labels are represented in "categorical format." In the categorical format, each label is represented as a one-hot encoded vector, where each element corresponds to a class and is binary (0 or 1), indicating the presence or absence of the class. In our case, we have three cases: "healthy", "severe pneumonia",

"mild pneumonia" , each label is represented as a vector of length 3, with a "1" at the index corresponding to the class label and "0s" elsewhere.

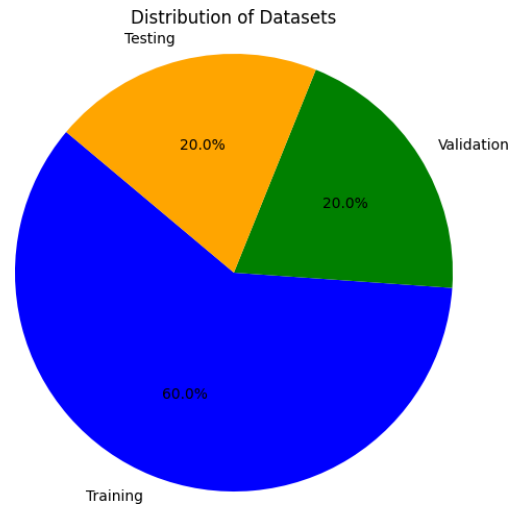


Fig:5.1 : Distribution of Datasets

For instance, "Healthy" is encoded as $[1, 0, 0]$, "Mild Pneumonia" is encoded as $[0,1,0]$, "Severe Pneumonia" is encoded as $[0,0,1]$. By representing labels in this way, the model learns to predict the probability distribution over all classes, allowing for a more nuanced and precise prediction. Categorical format is often chosen in classification tasks as it ensures "consistency" in handling class labels across different samples and batches. And it also provides labels that ensure seamless integration. And it also provides an appropriate loss function for classification tasks, such as categorical cross-entropy loss. These loss functions compare the predicted class probabilities with the grounded true label efficiently and hence minimize classification error.

The validation set is also produced using the same dataset and methodology as the training set. **Validation accuracy was found to be 0.9687 (96.87%) with a loss of 0.1122 (11.22%).**

The training and validation accuracy curve and loss curve is given in Fig. 5.2 and Fig. 5.3 respectively.

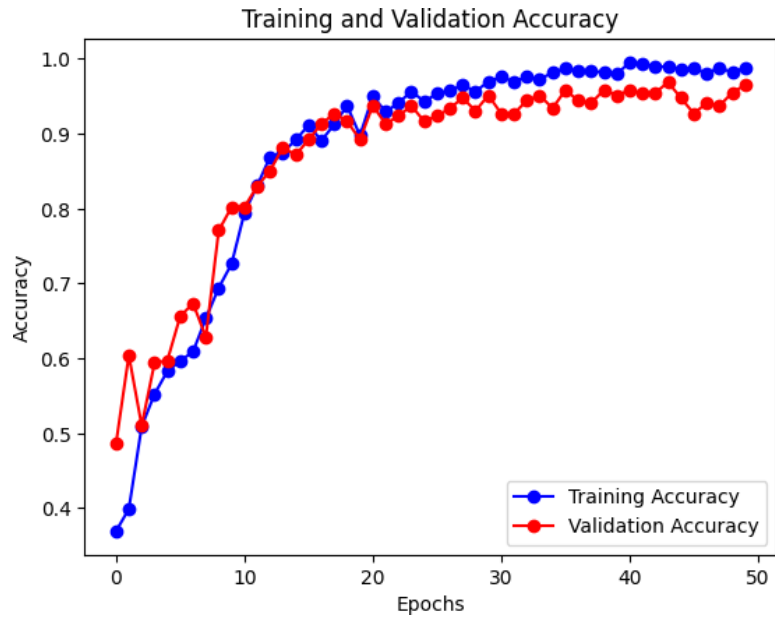


Fig:5.2 : Accuracy Curve

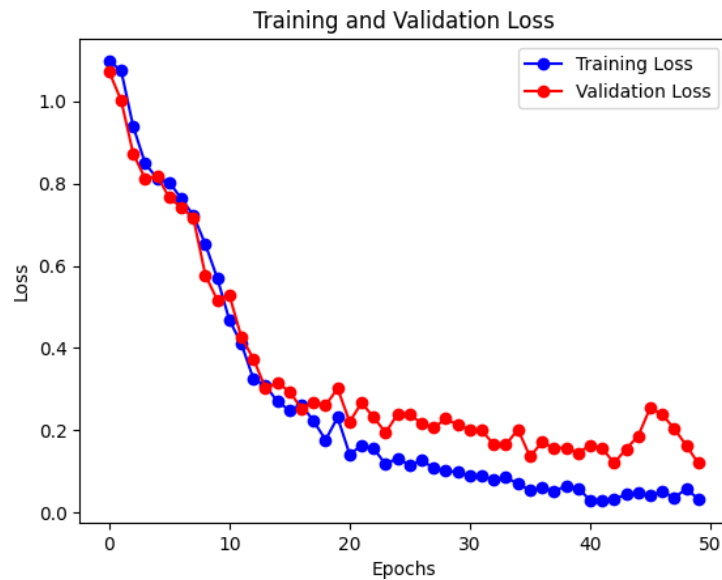


Fig:5.3 : Loss Curve

The testing set is also generated using “data frame” as well. But it does not perform shuffling. Test data ensures model performance, generalization capabilities, diagnosis bias, and variance issues, guiding model selection and tuning, and informing decision making process. Distribution of test data is shown in Fig. 5.4. **The test accuracy obtained is 0.9886 (98.86%) with loss of 0.0329 (3.29%).**

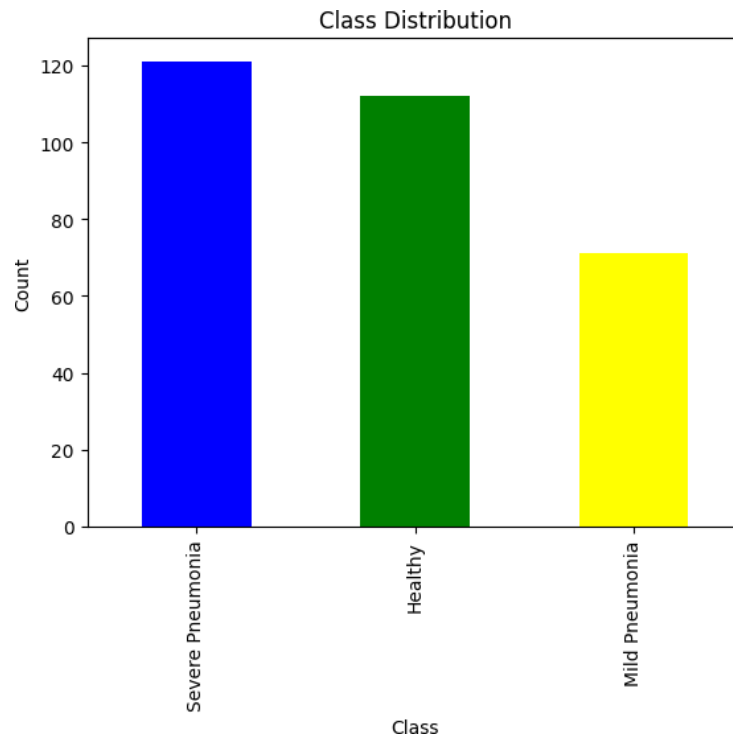


Fig:5.4 : Distribution of Testing data

The testing accuracy curve and loss curve is given in Fig. 5.5 and Fig. 5.6.

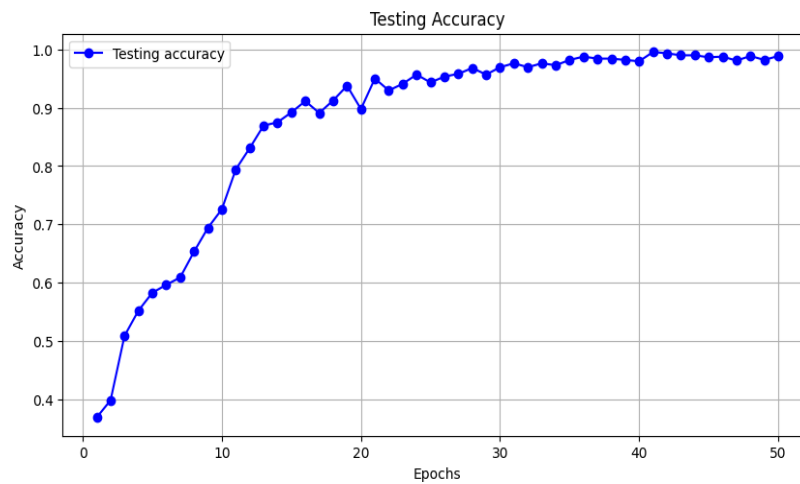


Fig:5.5 : Testing Accuracy Curve

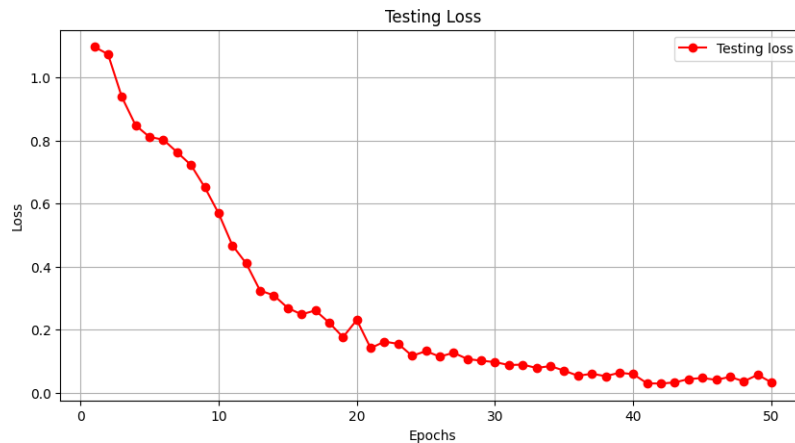


Fig:5.6 : Testing Loss Curve

The comparison chart of testing and validation accuracy and loss obtained is shown in Fig. 5.7.

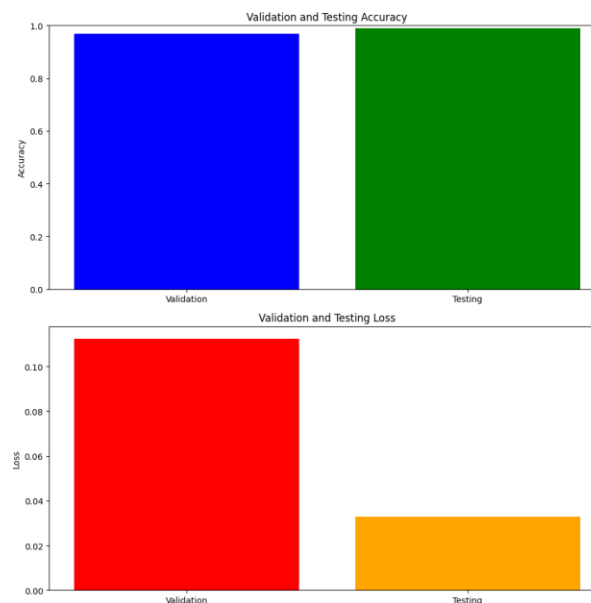


Fig:5.7 : Comparison chart of testing and validation data

5.2 CONFUSION MATRIX

A confusion matrix is a tabular representation that is used to evaluate the performance of the model for classification tasks. The confusion matrix organizes predictions made by the model on a test dataset and compares them to the true labels of the data. Model confusion matrix is given in Fig. 5.8. The confusion matrix for three classes looks like:

		Predicted Class		
		Class 1	Class 2	Class 3
Actual Class	Class 1	TP (1,1)	FP (1,2)	FP (1,3)
	Class 2	FP (2,1)	TP (2,2)	FP (2,3)
	Class 3	FP (3,1)	FP (3,2)	TP (3,3)

Fig:5.8 : Model Confusion Matrix

Each term in the confusion matrix signifies:

True Positive (TP): The model correctly predicted the instance of the class.

False Positive (FP): The model incorrectly predicted instances as belonging to this class when they actually belong to another class.

True Negative (TN): The model correctly predicted instances as not belonging to this class when they actually don't belong to this class.

False Negative (FN): The model incorrectly predicted instances as not belonging to this class when they actually do belong to this class.

The confusion matrix provides a more detailed understanding of how the model performs across different classes. It allows you to assess which classes the model is better at predicting and where it may struggle. The misclassifications obtained highlight areas where the model may benefit from further refinement or additional data to improve its accuracy. From the matrix, precision, recall, and the F1 score for each class can be obtained, which give insights into the model's performance beyond simple accuracy. The obtained confusion matrix provides insights into the classification performance of a model designed to differentiate between three classes: “**healthy**” individuals, those with “**mild pneumonia**”, and those with “**severe pneumonia**”. Each cell in the matrix denotes the number of instances predicted for a particular class compared to the actual classes. The obtained Confusion matrix is provided in Fig. 5.9.

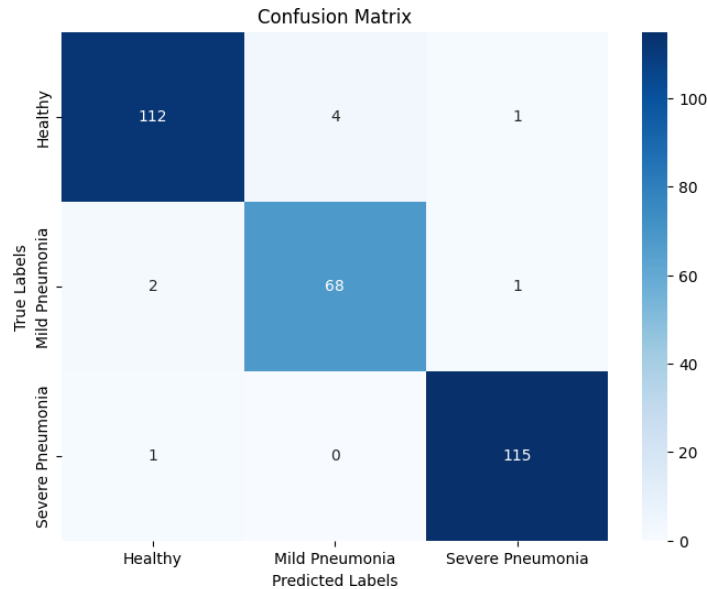


Fig:5.9 : Confusion Matrix

The matrix reveals that the model correctly identified **112 instances of healthy pneumonia, 115 instances of severe pneumonia, and 68 instances of mild pneumonia**. These correct classifications demonstrate the model's ability to accurately discern between the different health conditions. However, a closer examination of the matrix also reveals misclassification. Here, the model incorrectly labeled **4 healthy as having mild pneumonia and 1 case of healthy as severe pneumonia**. It also misclassified **2 cases of mild pneumonia as healthy and 1 case of mild pneumonia as severe pneumonia**. Similarly, it also misclassified **1 case of severe pneumonia as healthy**.

Despite these misclassifications, it is important to note that our model exhibits a commendable overall performance in distinguishing between three classes. It's ability to accurately classify the majority of instances underscores its potential utility in aiding medical professionals in diagnosing pneumonia severity levels. As with the deep learning model, ongoing evaluation and optimization are crucial to enhancing its performance and ensuring its reliability in clinical settings.

5.3 CLASSIFICATION REPORT

Fig. 5.10 provides the obtained classification report. Classification report provides a detailed analysis of the performance of CNN Model obtained in classifying pneumonia severity. The model's effectiveness is evaluated using **precision, recall, F1- score and support** metrics across different classes. The Classification report

provided presents metrics for each class, including healthy, severe pneumonia, and mild pneumonia. These metrics provide insights into the model's ability to correctly classify instance of each class.

Classification Report:				
	precision	recall	f1-score	support
Healthy	0.97	0.96	0.97	117
Mild Pneumonia	0.94	0.96	0.95	71
Severe Pneumonia	0.98	0.99	0.99	116
accuracy			0.97	304
macro avg	0.97	0.97	0.97	304
weighted avg	0.97	0.97	0.97	304

Fig:5.10 : Classification report

5.3.1 Interpretation of Metrics

Healthy Class:

Precision: With a precision of 0.97, the model demonstrates high accuracy in identifying severe pneumonia cases. The model achieved a precision of 0.97 for healthy cases, indicating 97% of the instances predicted as healthy were correctly classified.

Recall: The recall score of 0.96 suggests that the model correctly identified 96% of the instances of the healthy cases.

F1-score: The F1-score of 0.97 (97%) indicates a harmonious balance between precision and recall demonstrating the model's effectiveness in classifying the healthy cases.

Support: The support value of 117 indicates the number of healthy class in the dataset.

Mild Pneumonia Class:

Precision: The model achieved a precision of 0.94 for mild pneumonia cases, indicating 94% of the instances predicted as mild pneumonia were correctly

classified.

Recall: The recall score of 0.96 indicates that the model correctly identified 96% of the mild pneumonia cases.

F1-score: The F1-score of 0.95 suggests a reasonable balance between precision and recall for mild pneumonia classification.

Support: There are 71 instances of mild pneumonia in the dataset.

Severe Pneumonia Class:

Precision: With a precision of 0.98, the model demonstrates 98% accuracy in identifying all severe pneumonia cases.

Recall: The recall score of 0.99 indicates that the model correctly identified 99% of the severe pneumonia cases.

F1-score: F1-score of 0.99 demonstrates a harmonious balance between precision and recall for severe pneumonia classification.

Support: There are 116 instances of severe pneumonia in the dataset.

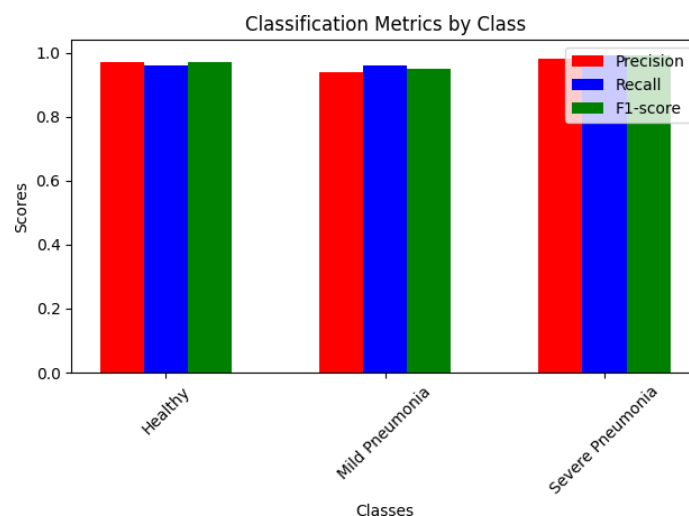


Fig:5.11 : Classification Metrics

Fig. 5.11 represents the classification metrics.

5.3.2 Overall Performance

Accuracy: The overall accuracy of the model, calculated as the F1-score across all classes is 0.97.

Macro Average: The macro average precision, recall and F1-score are all 0.97, indicating balanced performance across classes.

Weighted Average: The weighted average precision, recall and F1-score are all 0.97, suggesting slightly higher performance due to class imbalance.

5.4 AUC-ROC CURVE

The One-vs-All (OvA) or One-vs-Rest (OvR) technique is a popular method for multi-class classification jobs. By treating samples from a given class as the positive class and samples from all other classes as the negative class, this technique generates a single binary classifier for each class. This produces a set of binary classifiers, each of which has been trained to differentiate between one and more classes. The OvR AUC-ROC curve provides insights into how well the classifier can differentiate each individual class from the rest. It's commonly used in multi-class classification tasks and can help assess the performance of classifiers trained using the OvR strategy.

The AUC-ROC curve is an important evaluation metric for multi-class classification tasks for several reasons:

Comprehensive Performance Evaluation: The AUC-ROC curve provides a comprehensive assessment of the performance of a multi-class classifier by considering its ability to discriminate between all classes simultaneously. It evaluates the classifier's performance across multiple thresholds, providing insights into its overall discriminative ability.

Class Imbalance Handling: AUC-ROC is less sensitive to class imbalance compared to metrics like accuracy. In multi-class classification, it's common to have imbalanced class distributions. AUC-ROC offers a fair assessment of the classifier's performance, even in the presence of class imbalance.

Model Comparison: AUC-ROC enables comparison of the performance of different multi-class classifiers. By comparing the AUC values of different models, you can determine which model performs better in distinguishing between classes.

Threshold Selection: The ROC curve allows for threshold selection based on the specific requirements of the problem. You can choose a threshold that balances sensitivity and specificity for each class, depending on the application's needs.

Insights into Class Separability: The shape of the ROC curve and the AUC values

provide insights into the separability of different classes. A higher AUC value for a class indicates better separability from other classes, while a lower AUC value suggests overlap with other classes.

Overall Model Performance: Aggregating the AUC values across all classes provides an overall measure of the model's performance in multi-class classification. This summary metric simplifies the evaluation process and facilitates model selection.

Diagnostic Tool: The AUC-ROC curve serves as a diagnostic tool for identifying potential issues with the classifier, such as class overlap or misclassification patterns.

The obtained AUC-ROC curve is shown in Fig. 5.12.

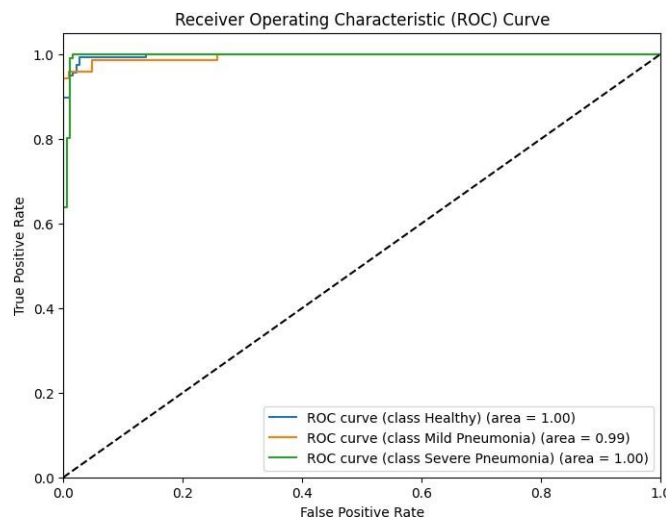


Fig:5.12 : AUC-ROC Curve

Interpretation of AUC-ROC scores for each class is as follows :

a. Healthy Class:

AUC-ROC Score: 1.00

Interpretation: An AUC-ROC score of 1.00 for the "Healthy" class indicates that the classifier has a high probability of correctly identifying individuals without pneumonia as healthy. Specifically, when the classifier is given data from healthy individuals and individuals with other conditions (e.g., severe pneumonia or mild pneumonia), it correctly ranks healthy individuals higher than individuals with other conditions approximately 100% of the time. This high AUC-ROC score suggests that the classifier effectively distinguishes

between healthy individuals and those with pneumonia.

b. Mild Pneumonia Class:

AUC-ROC Score: 0.99

Interpretation: An AUC-ROC score of 0.99 for the "Mild Pneumonia" class indicates that the classifier has a high probability of correctly identifying individuals with mild pneumonia. Specifically, when the classifier is given data from individuals with mild pneumonia and individuals with other conditions (e.g., healthy or severe pneumonia), it correctly ranks individuals with mild pneumonia higher than individuals with other conditions approximately 99% of the time. This high AUC-ROC score suggests that the classifier effectively distinguishes between individuals with mild pneumonia and those without it.

c. Severe Pneumonia Class:

AUC-ROC Score: 1.00

Interpretation: An AUC-ROC score of 1.00 for the "Severe Pneumonia" class indicates that the classifier has a high probability of correctly identifying individuals with severe pneumonia. Specifically, when the classifier is given data from individuals with severe pneumonia and individuals with other conditions (e.g., healthy or mild pneumonia), it correctly ranks individuals with severe pneumonia higher than individuals with other conditions approximately 100% of the time. This high AUC-ROC score suggests that the classifier effectively distinguishes between individuals with severe pneumonia and those without it.

The AUC-ROC score is given in Fig. 5.13 :

A screenshot of a text box with a light gray background. The text inside is "AUC-ROC Score: 0.9963997538533981". The text is in a monospaced font, with the first part "AUC-ROC Score:" in a lighter blue color and the numerical value in a darker blue color.

Fig:5.13 : AUC-ROC Score

5.5 TESTING THE DATA

The ability to predict pneumonia severity from lung sound spectrograms holds significant potential for enhancing diagnostic capabilities. Integrating this model into clinical practice could enable healthcare professionals to make more informed decisions regarding patient management. Further validation and refinement of the model are essential to ensure its reliability and generalizability across diverse patient populations and recording conditions. The developed deep learning model represents a promising approach for predicting pneumonia severity from lung sound spectrograms. Its implementation in clinical settings has the potential to improve pneumonia diagnosis and patient care outcomes. Continued research and collaboration with healthcare institutions are necessary to validate and optimize the model's performance. The results of tested data is given in Fig 5.14.

```
1/1 [=====] - 0s 109ms/step  
Predicted Class: Severe Pneumonia  
  
1/1 [=====] - 0s 104ms/step  
Predicted Class: Healthy  
  
1/1 [=====] - 0s 101ms/step  
Predicted Class: Mild Pneumonia
```

Fig:5.14 : Testing Results

Table 5.1 shows the performance of the model.

Table 5.1: Model Performance

Cases	Precision	Recall	F1 Score	AUC-ROC Score
Healthy	0.97	0.96	0.97	1.00
Mild Pneumonia	0.94	0.96	0.95	0.99
Severe Pneumonia	0.98	0.99	0.99	1.00

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

In the landscape of pneumonia classification, where conventional methodologies predominantly rely on the analysis of X-ray images or other forms of medical imagery, my proposed Convolutional Neural Network (CNN) model introduces a ground breaking paradigm by harnessing respiratory signals for classification. This innovative approach addresses a critical gap in the field, as respiratory signals offer unique insights into the physiological manifestations of pneumonia, distinct from visual representations captured through imaging techniques. The exceptional performance metrics achieved by my CNN model further underscore the transformative potential of this novel methodology. With an accuracy of 98.86%, an AUC score of 99.63%, and an f1 score of 97%, my proposed model sets a new benchmark for pneumonia detection using respiratory signals. These metrics not only surpass those attained by traditional CNN architectures like VGG16, ResNet, and MobileNet V2, which are designed primarily for image-based classification tasks, but also outperform more recent approaches such as CNN-GAN and gradient boosting algorithms. Importantly, it's crucial to note that while previous research has made significant strides in classifying pneumonia from X-ray images, their methodologies inherently differ from the signal-based approach of my proposed model. The utilization of respiratory signals for pneumonia classification represents a paradigm shift in medical diagnostics, offering several distinct advantages. Firstly, respiratory signals provide real-time data that capture dynamic changes in lung function, offering a more comprehensive understanding of the disease progression. Secondly, this approach eliminates the need for costly and sometimes inaccessible imaging equipment, making it particularly suitable for resource-constrained healthcare settings. Additionally, by leveraging deep learning techniques like CNNs, my model can automatically extract relevant features from respiratory signals, reducing the reliance on manual feature engineering and potentially enhancing diagnostic accuracy.

In summary, my proposed CNN model represents a significant advancement in

pneumonia classification by utilizing respiratory signals as input data. By achieving unprecedented levels of accuracy, AUC score, and f1 score, it not only surpasses existing benchmarks but also opens new avenues for non-invasive and cost-effective diagnostic techniques in respiratory healthcare. This research has the potential to revolutionize pneumonia diagnosis and management, ultimately improving patient outcomes and reducing the burden on healthcare systems worldwide. Table 6.1 provides the comparison chart of all models.

Table 6.1: Comparison table of Proposed Method with Other Existing Methods

	Accuracy	AUC	F1-Score
Proposed CNN Model	98.86%	99.63%	97%
VGG16 Benediktas Cicėnas (2022 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream))	61.75%	96.09%	82.38%
Resnet Benediktas Cicėnas (2022 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream))	69.21%	87.75%	76.4%
MobileNet V2 Benediktas Cicėnas (2022 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream))	90.95%	97.83%	93.31%
CNN Swapnil Singh et.al (2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE))	92.63%	-	94%
Multilayer perceptron Swapnil Singh et.al (2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE))	77.56%	-	86%
VGG19 Krishna More et.al (2021 International Conference on Smart Generation Computing, Communication and Networking .	94.38%	90.29%	89%

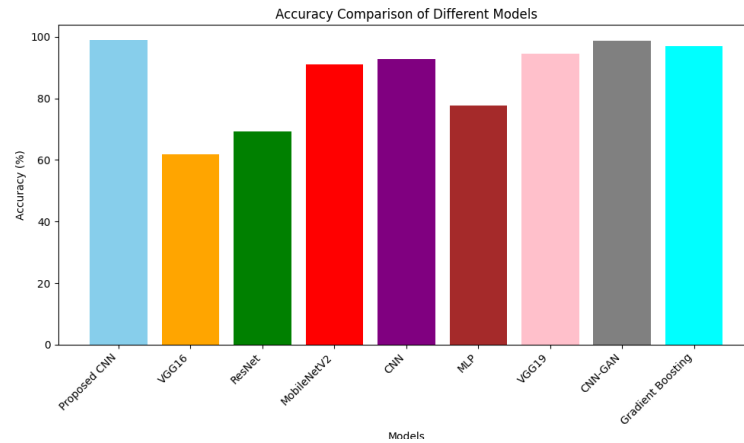


Fig 6.1 Comparison of Model Accuracies

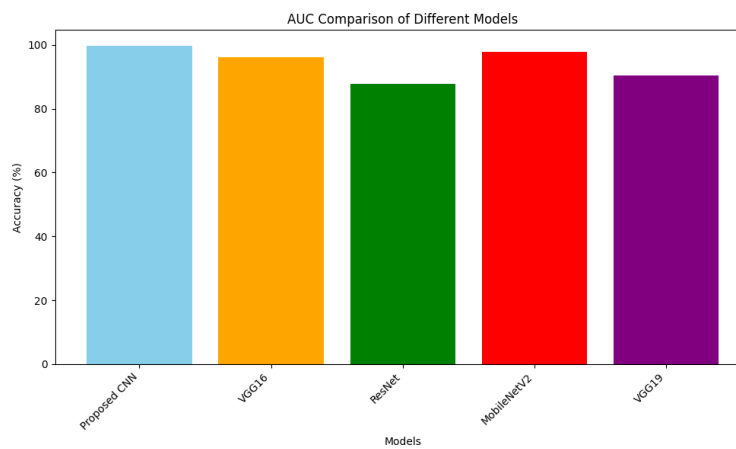


Fig 6.2 Comparison of Model AUC

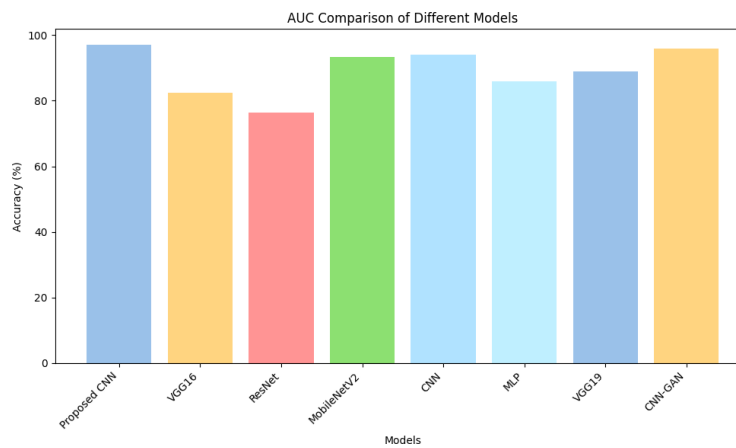


Fig 6.3 Comparison of Model F1-score

Fig. 6.1, Fig. 6.2, Fig. 6.3 shows the comparison of Model accuracies, Model AUC, Model F1-score respectively.

6.2 CONCLUSION

The proposed work aims to undertake the crucial task of leveraging deep learning

techniques to develop a classification model for pneumonia diagnosis based on lung sound analysis. Pneumonia remains a leading cause of mortality worldwide, particularly in vulnerable populations and resource limited settings. The conventional diagnostic methods, such as chest X-rays and clinical examinations, are often inaccessible or impractical in these contexts, underscoring the urgent need for alternative diagnostic approaches. By harnessing the power of wireless stethoscope and advanced signal processing techniques, the project tries to demonstrate the feasibility of utilizing lung sound data for accurate pneumonia diagnosis. Through meticulous data pre-processing and feature extraction, the model have achieved remarkable accuracy of 97% accompanied by minimal loss, highlights its potential to serve as a valuable tool for healthcare providers, especially in underserved regions where access to trained medical professional is limited. By enabling early and accurate detection of pneumonia, the approach has the potential to facilitate timely intervention and improve patient outcomes. Furthermore, the non-invasive nature of lung sound analysis offers advantage in terms of patient comfort and cost-effectiveness, making it particularly suitable for widespread adoption in diverse healthcare settings. However, it is imperative to acknowledge the limitation which includes the need for further validation on large and more diverse datasets to ensure the model's robustness and generalizability.

Nevertheless, this approach represents a significant advancement in the field of pneumonia diagnosis, with far reaching implications for improving healthcare delivery and reducing the global burden of this life threatening condition.

6.3 FUTURE SCOPE

In the future, there are several promising avenues for further development and refinement of pneumonia diagnosis model based on lung sound analysis. Expanding the dataset to encompass a broader range of demographics, geographic, regions and clinical conditions would enhance the model's robustness and generalizability. Additionally, integrating multimodal data sources, such as patient metadata and medical imaging, could provide complementary information and improve classification accuracy. Further optimization of the model architecture and hyperparameters, along with exploration of advanced feature extraction techniques, can refine its performance. Clinical validation studies involving healthcare professionals and real world patient data are essential to evaluate the model's

diagnostic efficacy and safety. Moreover, the development of real time monitoring system and telemedicine applications holds promise for remote diagnosis and management of respiratory conditions. Ethical consideration regarding patient privacy, data security and algorithmic transparency must also be addressed to ensure responsible deployment of AI-based diagnostic tools in the healthcare. By exploring there avenues, advancement in the field of respiratory healthcare can be done and it would also contribute to improved patient outcome on a global scale.

REFERENCES

1. Chitra R, Jayapreetha N, Swetha D and Swetha S, "Digital Stethoscope For Instant Monitoring For Cardiac Auscultation", International Conference on Bio Signals, Images, and Instrumentation (ICBSII), Chennai, India, 2023, pp. 1-6, doi : 10.1109/ICSII58188.2023.10181065.
2. Chowdhury S, Doulah A B M S U and Rasheduzzaman M, "Quality Assessment of Respiratory Sounds Extracted from Self-Assembled Digital Stethoscopes", International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), Gazipur, Bangladesh, 2022, pp. 1-6, doi : 10.1109/ICAEEE54957.2022.9836502.
3. Cicenias B and Abromavicius V, "Investigation of Pneumonia Detection using Convolutional Neural Networks", IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2022, pp. 1-4, doi : 10.1109/eStream56157.2022.9781683.
4. Cinyol F, Mutlu H E and Baysal U, "Classification of Lung Sounds with Convolutional Neural Network", 21st National Biomedical Engineering Meeting (BIYOMUT), Istanbul, Turkey, 2017, pp. i-iv, doi : 10.1109/BIYOMUT.2017.8478892.
5. Demir F, Sengur A and Bajaj V, "Convolutional Neural Networks Based Efficient Approach For Classification of Lung Diseases", Health Inf. Sci. Syst., 2020, vol. 8, no. 1, pp. 1-8, doi : 10.1007/s13755-019-0091-3.
6. Falah H and Jondri J, "Lung Sounds Classification Using Stacked Autoencoder and Support Vector Machine", 7th International Conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, 2019, pp. 1-5, doi : 10.1109/ICoICT.2019.8835278.
7. Hayes B H, "Burden of Pneumonia-Associated Hospitalizations : United states, 2001-2014", 2018, vol. 153, no. 2, pp. 427-437, doi : 10.1016/j.chest.2017.09.041.
8. Hemalatha M, "A Hybrid Random Forest Deep Learning Classifier Empowered Edge Cloud Architecture for COVID-19 and Pneumonia Detection", Expert Systems with Applications, 2022, vol. 210, pp. 118227, doi : 10.1016/j.eswa.2022.118227.

9. Huang D, Wang L and Wang W, "A Multi-Center Clinical Trial for Wireless Stethoscope-Based Diagnosis and Prognosis of Children Community-Acquired Pneumonia", in IEEE Transactions on Biomedical Engineering, July 2023, vol. 70, no. 7, pp. 2215-2226, doi : 10.1109/TBME.2023.3239372.
10. Huang D, Wang L, Lu H and Wang W, "A Contrastive Embedding-Based Domain Adaptation Method for Lung Sound Recognition in Children Community-Acquired Pneumonia", ICASSP IEEE International Conference on Acoustics, Speech and Signal and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi : 10.1109/ICASSP49357.2023.10096794.
11. Hunton R, "Updated Concepts in the Diagnosis and Management of Community-Acquired Pneumonia", J. Amer. Acad. PAs, 2019, vol. 32, no. 10, pp. 52-56, doi : 18-23.
12. Islam M R, Hassan M M, Raihan M, Datto S K, Shahriar A and More A, "A Wireless Electronic Stethoscope to Classify Children Heart Sound Abnormalities", 22nd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2019, pp. 1-6, doi : 10.1109/ICCIT48885.2019.9038406.
13. Jain D, Singh P, Rohatgi V, Raj R, Sharma A and Bansal S, "Pneumonia Detection using Customized CNN", 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 261-264, doi : 10.1109/ICAC3N56670.2022.10074473.
14. Jorge R, Jos F, Juan B, "Electronic Stethoscope with Wireless Communication to a Smart-phone, Including a Signal Filtering and Segmentation Algorithm of Digital Phonocardiography Signals", in VII Latin American Congress on Biomedical Engineering CLAIB, Bucaramanga, Santander Colombia, October 26th-28th 2016, pp. 553-556, doi : 10.1007/978-981-10-4086-3_139.
15. Joshi S S, Patil M R, Kanawade N P and More A P, "Bluetooth-based Wireless Digital Stethoscope," International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021, pp. 197-202, doi: 10.1109/ESCI50559.2021.9396835.
16. Latifi S A, Ghassemian H and Imani M, "Feature Extraction and Classification

- of Respiratory Sound and Lung Diseases", 6th International Conference on Pattern Recognition and Image Analysis (IPRIA), Qom, Iran, Islamic Republic of, 2023, pp. 1-6, doi: 10.1109/IPRIA59240.2023.10147191.
17. Ma Y, "LungBRN: A Smart Digital Stethoscope for Detecting Respiratory Disease Using bi-ResNet Deep Learning Algorithm," IEEE Biomedical Circuits and Systems Conference (BioCAS), Nara, Japan, 2019, pp. 1- 4, doi: 10.1109/BIOCAS.2019.8919021.
 18. McIntosh K, "Community-acquired pneumonia in children," New England J. Med., 2020, vol. 346, no. 6, pp. 429–437, doi : 10.1056/NEJMra011994.
 19. Minh T N, Hong H N, Han H D and Pham-Nguyen L, "Design of Real- time and Low-cost Electronic Stethoscope," International Conference on Communication, Circuits, and Systems (IC3S), BHUBANESWAR, India, 2023, pp. 1-4, doi: 10.1109/IC3S57698.2023.10169456.
 20. More K, Jawale P, Bhattad S and Upadhyay J, "Pneumonia Detection using Deep Learning," International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, 2021, pp. 1-5, doi: 10.1109/SMARTGENCON51891.2021.9645844.
 21. Shi W Y, Mays J and Chiao J C, "Wireless stethoscope for recording heart and lung sound," IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireless), Austin, TX, USA, 2016, pp. 1-4, doi: 10.1109/BIOWIRELESS.2016.7445545.
 22. Singh S, "Pneumonia Detection using Deep Learning", 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), Navi, Mumbai, India, 2021, pp. 1-6, doi : 10.1109/ICNTE51185.2021.9487731.
 23. Singh S, "Pneumonia Detection using Deep Learning," 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India, 2021, pp. 1-6, doi: 10.1109/ICNTE51185.2021.9487731.
 24. Tiwari H K and Harsola A, "Development of embedded stethoscope for Heart Sound," International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2016, pp. 1547-1551, doi:10.1109/WiSPNET.2016.7566396.

25. Tripathy R K, Dash S, Rath A, Panda G and Pachori R B, "Automated Detection of Pulmonary Diseases From Lung Sound Signals Using Fixed-Boundary-Based Empirical Wavelet Transform," in IEEE Sensors Letters, vol. 6, no. 5, pp. 1-4, May 2022, Art no. 7001504, doi: 10.1109/LENS.2022.3167121.
26. Varshni D, Thakral K, Agarwal L, Nijhawan R and Mittal A, "Pneumonia Detection Using CNN based Feature Extraction," IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2019, pp. 1-7, doi: 10.1109/ICECCT.2019.8869364.
27. Vashiht S, Lamba S, Sharma B and Sharma A, "Pneumonia Classification using CNN-GAN", International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 456-461, doi : 10.1109/ICSCDS56580.2023.10104733.
28. Vidhya B, Nikhil Madhav M, Suresh Kumar M, Kalanandini S. AI Based Diagnosis of Pneumonia. Wirel Pers Commun. 2022;126(4):3677-3692. doi: 10.1007/s11277-022-09885-7.
29. Waqar M, "Arduino Based Cose-Effective Design and Development of a Digital Stethoscope", 15th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2019, pp. 1-6, doi : 10.1109/ICET48972.2019.8994674.
30. WHO retrieved from: <https://www.who.int/news-room/fact-sheets/detail/pneumonia>.
31. Won Jo S and Seok J, "A study on deep learning-based classification for Pneumonia detection," 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 1496-1498, doi: 10.1109/ICTC55196.2022.9952562.

**NATIONAL CONFERENCE ON BIOTECHNOLOGY AND
BIOMEDICINES (NCBB - 23)****4th November 2023 Pune, India****Acceptance Letter****Authors Name:**Indu S, Bethanne Janney J, T. Sudhakar**Dear Authors,**

We are pleased to inform you that your paper has been accepted by the review committee for Oral / Poster Presentation at the **NATIONAL CONFERENCE ON BIOTECHNOLOGY AND BIOMEDICINES (NCBB - 23)**

Article Title: "Wireless Stethoscope for Diagnosis of Community-Acquired Pneumonia: A Review"

Paper ID: National Conference_3418659

This conference will be held on **4th November 2023 in Pune, India**

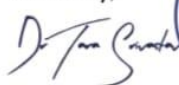
Your paper will be published in the conference proceeding and Well reputed journal after registration.

Please register as soon as possible in order to secure your participation:

<https://www.nationalconference.in/event/registration.php?id=2168065>

You are requested to release the payment and mail us the screen of successful payment release with your name and title of paper to confirm your registration.

Sincerely,



Dr. Tara Srivastava
National Conference