CAPSTONE PROJECT REPORT

DETECTION OF RESPIRATORY DISEASES USING DEEP LEARNING ALGORITHMS BASED ON RESPIRATORY SOUND ANALYSIS

Mentor: Pankaj Agarwal

Contents

TEAM MEMBERS

Praneeth Devanabanda

Forum Joshi

Varnikha Sree A R

Manasa Shetty

B Shyam Sundar

ABSTRACT

Introduction:

Listening to a patient's breathing with a stethoscope has long been the primary method for detecting respiratory diseases, but it can be subjective and lead to misinterpretation. This traditional approach is time-consuming and may not always be feasible in every healthcare setting. With advancements in technology, new tools are being developed to improve the accuracy but making a machine learn with historical data, it boosts the efficiency of diagnosing respiratory conditions, making it more objective and reliable ways to monitor respiratory patterns, revolutionizing the way we detect and manage respiratory diseases. This project aims to further investigate the respiratory disease in the body of a patient.

Methods:

This project utilizes LSTM and CNN for analyzing respiratory sounds, offering significant advancements in diagnosing respiratory diseases within healthcare settings. Deep learning models have showcased impressive accuracy levels exceeding 90% for specific conditions, enabling faster and more precise detection of ailments for timely intervention and treatment. Integrating deep learning algorithms into digital health platforms supports remote monitoring and telemedicine services for patients with respiratory issues. To enhance patient care, deep learning techniques, particularly CNNs, are effective in feature extraction from image datasets. In our experiments, the Librosa machine learning library was utilized for feature extraction, including MFCC, Mel-Spectrogram, and Chroma for processing audio files. By applying optimization techniques, we achieved a high accuracy rate of 95% in our deep learning approach.

Keywords: Deep learning, CNN-LSTM based classification, Medical-assistive technology, Respiratory sound analysis, Machine learning.

PROBLEM STATEMENT

Many respiratory diseases are frequently undiagnosed or misdiagnosed, resulting in delayed treatment and potentially worsening health outcomes. The current diagnostic approaches heavily depend on physical examinations and medical history, which can be subjective and susceptible to human error.

One of the primary obstacles in the identification of respiratory diseases through respiratory sound analysis is the absence of standardized procedures for data collection and analysis. Various healthcare facilities may utilize different tools and methodologies for recording and interpreting respiratory sounds, leading to inconsistencies and possible inaccuracies in the diagnoses. Moreover, the traditional methods of evaluating respiratory sounds often necessitate specialized professionals to interpret the results, which can be time-consuming and resource-intensive.

Utilizing deep learning techniques for analyzing respiratory sounds presents an opportunity to enhance the accuracy and efficiency of detecting respiratory diseases. Deep learning algorithms can automate the analysis of respiratory sounds, improving the precision and speed of disease detection. By leveraging CNN-LSTM hybrid architectural models, these algorithms can learn hierarchical representations of data through interconnected neural networks, enabling them to discover complex features and patterns in the input data. Through training on extensive datasets of respiratory sound recordings, these algorithms can identify distinctive features associated with various respiratory conditions, resulting in more reliable and robust detection models.

Furthermore, deep learning models possess the capability to continually enhance and adjust to fresh data, rendering them more flexible and suitable for medical diagnosis. In contrast, conventional machine learning models usually necessitate manual feature selection and engineering, a process that can be time-consuming and less efficient in capturing all pertinent information from the data.

By harnessing the capabilities of deep learning, healthcare professionals can enhance the efficiency of diagnosing illnesses and enhance patient recovery. This involves establishing consistent procedures for gathering and evaluating data, verifying the accuracy of deep learning algorithms across various demographic groups, and incorporating these algorithms into current healthcare infrastructure. The integration of deep learning technology in detecting respiratory diseases has the capacity to transform the field of respiratory medicine and elevate the standard of patient treatment. Our study seeks to

explore the promise of deep learning in analyzing respiratory sounds for the identification of respiratory conditions.

DATA DESCRIPTION

For this work we have used the 2017 ICBHI dataset which is a comprehensive resource providing researchers with a large database of labeled respiratory sounds. It comprises 920 audio recordings totaling 5.5 hours in duration. These recordings vary in length, ranging from 10 to 90 seconds, and were captured at diverse sampling frequencies, spanning from 4 kHz to 44.1 kHz. The dataset encompasses recordings from 128 patients, each identified as either healthy or presenting one of several respiratory diseases or conditions, including COPD, Bronchiectasis, Asthma, upper and lower respiratory tract infections, Pneumonia, and Bronchiolitis. These respiratory condition labels are associated with the audio recording files.

Within each audio recording, experts have identified four different types of respiratory cycles: Crackle, Wheeze, Both (Crackle & Wheeze), and Normal. These cycles are meticulously labeled, indicating their onset and offset times. The lengths of these cycles vary, ranging from 0.2 up to 16.2 seconds. Moreover, the dataset exhibits an unbalanced distribution of cycles, with 1864, 886, 506, and 3642 cycles respectively for Crackle, Wheeze, Both, and Normal categories.

It also offers recordings captured using various equipment, including the AKG C417 L Microphone, 3 M Littmann Classic II SE Stethoscope, 3 M Littmann 3200 Electronic Stethoscope, and WelchAllyn Meditron Master Elite Electronic Stethoscope. These recordings originate from hospitals in Portugal and Greece, capturing sounds from different chest locations: Trachea, Anterior left, Anterior right, Posterior left, Posterior right, Lateral left, and Lateral right. Additionally, a notable portion of the samples exhibits noise, adding complexity to the classification task. These diverse characteristics render the classification problem more challenging, mirroring real-world scenarios more closely compared to datasets recorded under ideal conditions.

Audio files and necessary datasets link here.

OVERVIEW OF THE FINAL PROCESS

Here's a simplified overview of the final process:

→ We started by gathering all audio files using the glob function. These files had patient numbers along with additional details like audio acquisition mode and recording equipment. Using the patient numbers, we merged them with the 'patient_diagnosis.csv' file, which contains information on the diagnosed respiratory diseases of each patient. This resulted in a dataframe linking audio files to their respective disease diagnoses.

→ Next, we wrote a parser function to extract important features from the audio files. These 189 features extracted from each audio file became our 'X'. The target variable was the respiratory diagnosis associated with each audio file.

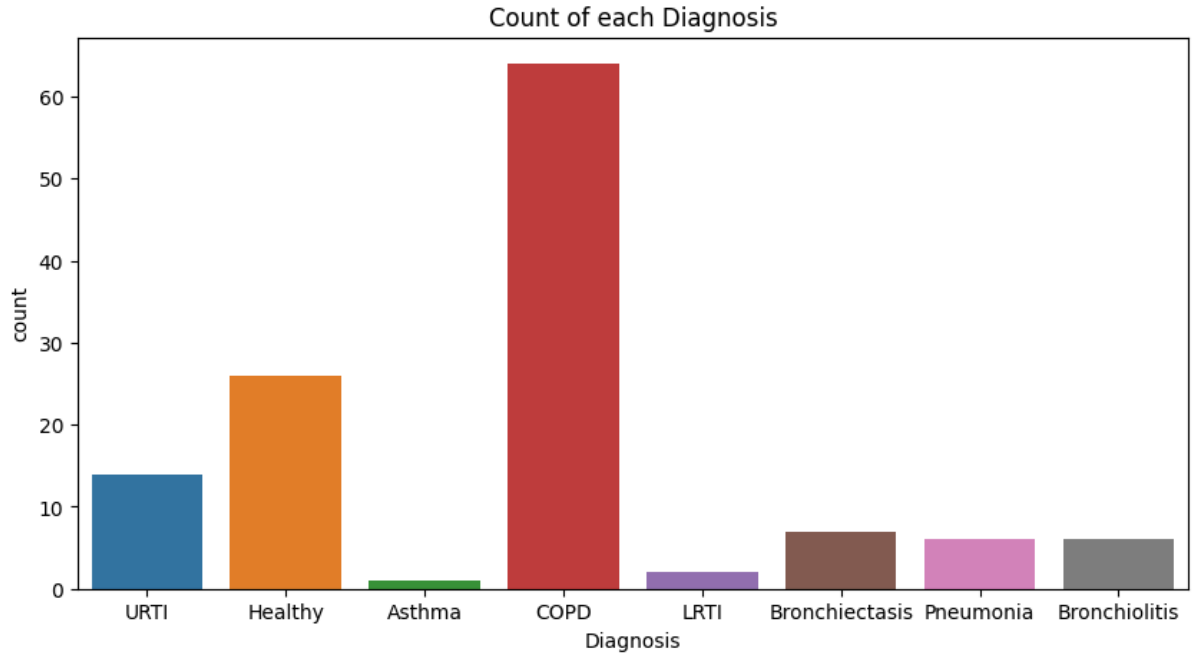
→ To address class imbalance, we used SMOTE to oversample minority classes and downscaled the majority class by half.

→ After encoding the target classes, we split the data into training and testing sets in an 80:20 ratio. We then applied machine learning (ML) and deep learning (DL) models to the data. For each model, we recorded metrics such as accuracy, precision, recall, and F1 scores for each class. Finally, we selected the CNN-LSTM model as our final model, as it best suited our objectives for this dataset.

STEP-BY-STEP WALKTHROUGH

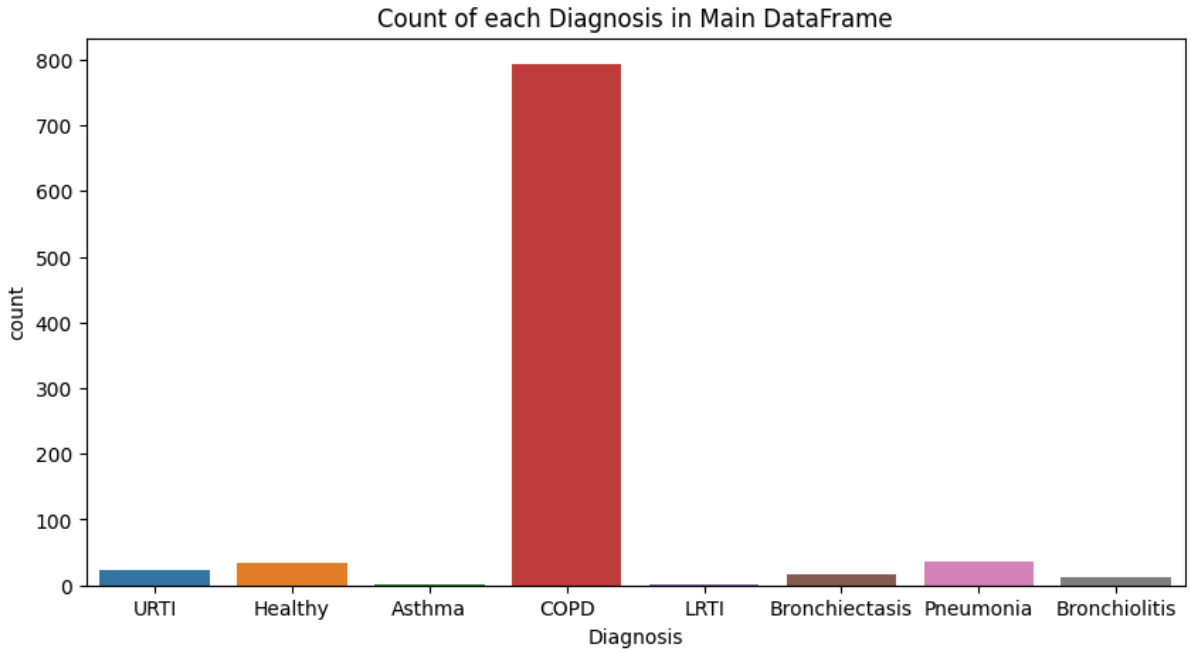
1. Looking into Patient details and Diagnosis diseases

→ We began by examining the 'Patient_Diagnosis.csv' file, which provided information about patient numbers and the respiratory diseases they were diagnosed with. A simple count plot of the distribution of diseases in the dataset is as follows:



→ There were a total of 920 audio files, each with complex filenames containing various details such as patient numbers, recording indices, chest locations, acquisition modes, and recording equipment details. We focused only on the patient numbers and extracted them. Using the glob function, we gathered all audio files into an array for easy access. These audio files were then matched with the previous diagnosis information. The initial rows of the data frame with the assigned target label is as follows:

	Patient number	audio_file_name	Diagnosis
0	101	101_1b1_Al_sc_Meditron.wav	URTI
1	101	101_1b1_Pr_sc_Meditron.wav	URTI
2	102	102_1b1_Ar_sc_Meditron.wav	Healthy
3	103	103_2b2_Ar_mc_LittC2SE.wav	Asthma
4	104	104_1b1_Al_sc_Litt3200.wav	COPD
5	104	104_1b1_Ll_sc_Litt3200.wav	COPD
6	104	104_1b1_Ar_sc_Litt3200.wav	COPD
7	104	104_1b1_Lr_sc_Litt3200.wav	COPD
8	104	104_1b1_Pl_sc_Litt3200.wav	COPD
9	104	104_1b1_Pr_sc_Litt3200.wav	COPD
10	105	105_1b1_Tc_sc_Meditron.wav	URTI
11	106	106_2b1_Pl_mc_LittC2SE.wav	COPD
12	106	106_2b1_Pr_mc_LittC2SE.wav	COPD
13	107	107_2b3_Pl_mc_AKGC417L.wav	COPD
14	107	107_2b3_Al_mc_AKGC417L.wav	COPD
15	107	107_2b4_Ar_mc_AKGC417L.wav	COPD
16	107	107_2b3_Ar_mc_AKGC417L.wav	COPD

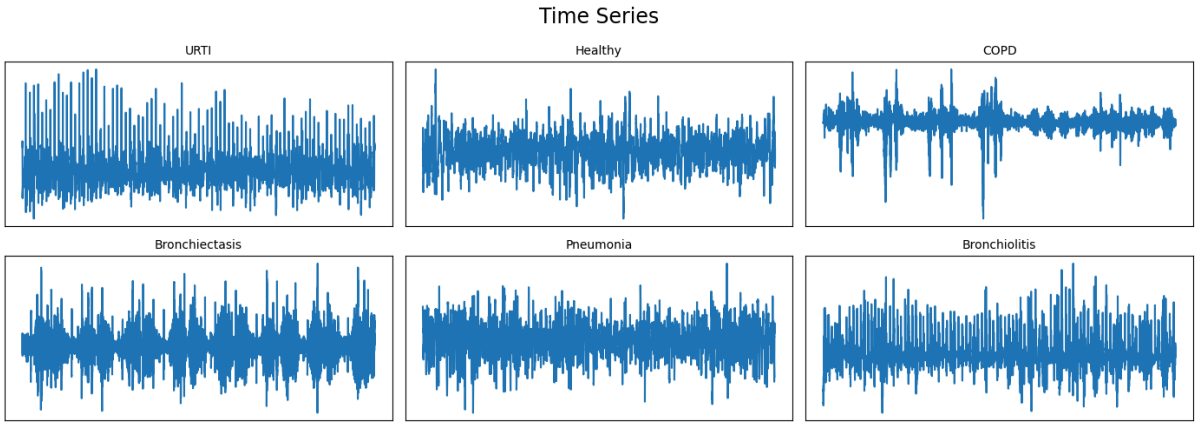


→ Notably, there was a significant class imbalance, with a high number of audio files diagnosed with 'COPD.' Due to the small number of 'Asthma' and 'LRTI' audio files, they were removed entirely from the dataset.

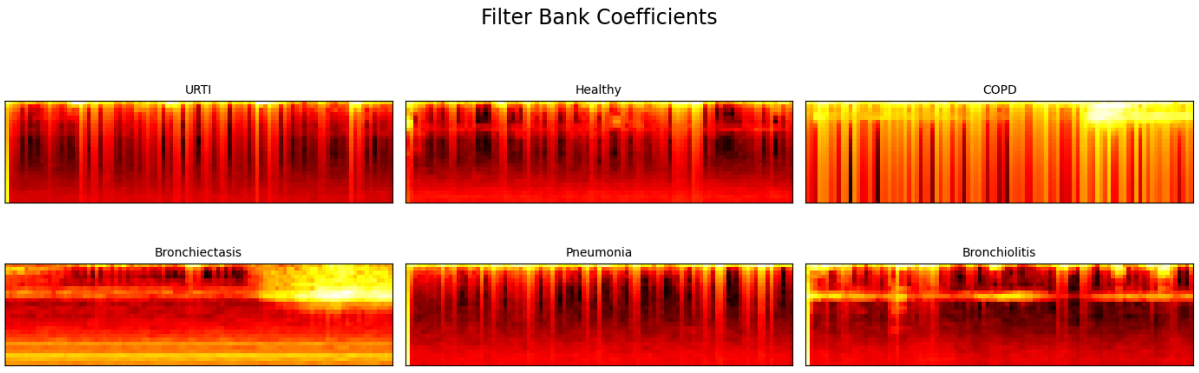
2. Looking into Audio Files

→ We experimented with audio files, using ipd.Audio() to listen to selected samples. We also adjusted the sampling rates of some files to observe any differences. Additionally, we created charts using random audio files from each target class. Some of the charts included:

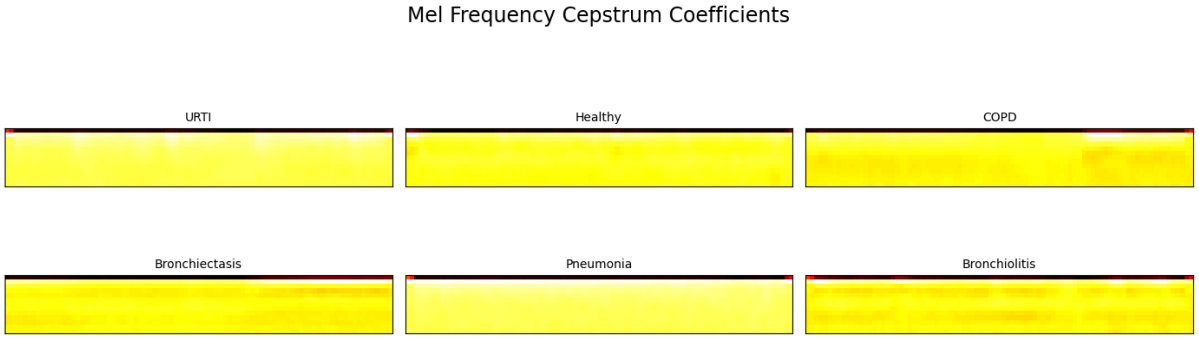
→ Time Series:



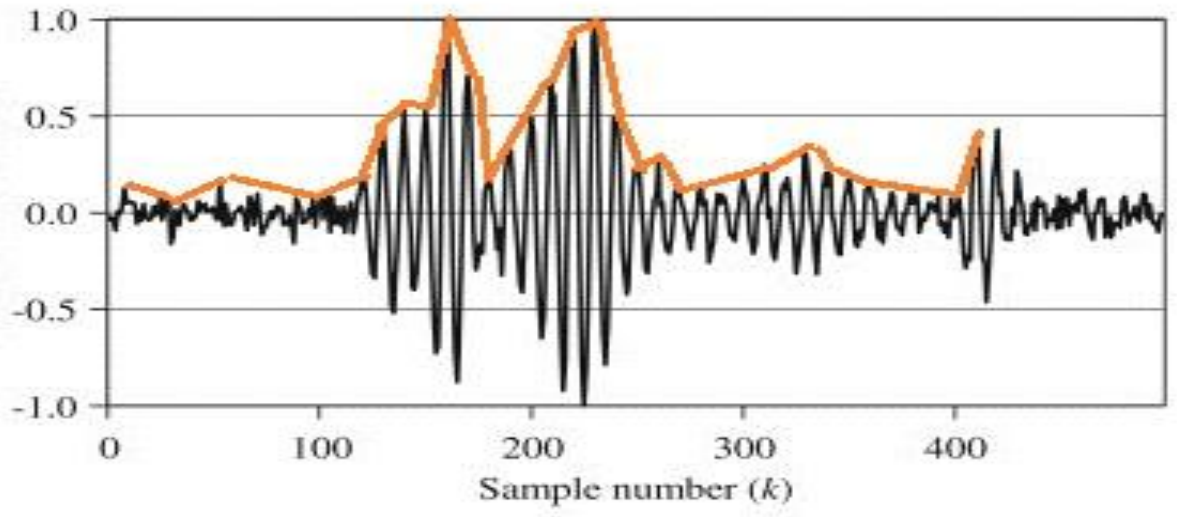
→ Filter Bank Coefficients:



→ Mel Frequency Cepstrum Coefficients:



→ In our preprocessing of audio files, we implemented a technique known as 'Envelope'. This method involves generating a curve that depicts the variation of a specific characteristic of the audio signal over time. The envelope is commonly utilized to capture the overall shape or magnitude of the signal, aiding in the identification of its dynamic changes. An example is as follows:



→ To begin the envelope process, we developed a custom function that takes parameters such as the audio signal (y), sample rate, and a threshold value. Based on this threshold, we removed any portions of the audio file that were deemed to be silent or irrelevant. This cleaning process was applied to the audio files, resulting in enhanced data quality.

3. Feature Engineering and Extraction

→ A brief description of 5 key features extracted are:

a. Mel-frequency cepstrum coefficients (MFCCs): MFCCs represent the power spectrum of a sound, capturing its spectral characteristics. These coefficients are derived using a transformation process involving the log

power spectrum on a mel-frequency scale and discrete cosine transforms. We extract 24 custom numbers as 1D arrays.

b. Chromagram: Chromagrams map audio pitches into a single octave, consisting of 12 semitones. We extract chroma features from each audio recording by combining Q Transform and Short-Time Fourier Transform (STFT). Again, we extract 24 numbers.
c. Mel-scaled spectrogram: The Mel-scaled spectrogram visually represents a time series audio file as a 2D image, with time on the x-axis and frequency on the y-axis. Each pixel's brightness corresponds to a specific point in time and frequency in the sound file. We extract 128 features from this.
d. Spectral contrast: Spectral contrast measures the decibel difference between peaks and valleys in an audio spectrum, providing insights into frequency band contrast over a harmonic spectrum. We extract 6 features from this.
e. Tonal centroids: Tonal centroids represent the central pitches of an audio sequence, summarizing its tonal characteristics and movements over time. We extract 7 features from this.

→ Using the librosa library, we can easily calculate these features. In total, we extract 189 features and concatenate them into a 1D array. This entire process is user defined as a parser function which we run on the entire data frame to finally output 189 features and target label for each audio file.

4. Dealing with Imbalance of Target Classes in the Dataset

→ The distribution of target classes in the dataset is as follows: {Bronchiectasis: 16, Bronchiolitis: 13, COPD: 793, Healthy: 35, LRTI: 2, Pneumonia: 37, URTI: 23}. Approximately 80% of the records belong to the 'COPD' class. To address this imbalance, we utilized the SMOTE function from the Imbalance Learn library. SMOTE, which stands for Synthetic Minority Over-sampling Technique, generates synthetic data points near existing class data points in mathematical space.

→ We customized the upsampling of minority classes and reduced the number of records in the 'COPD' class by half. As a result, the final class frequency distribution became:

Target Labels	Before	After
Bronchiectasis	16	100
Bronchiolitis	13	100
COPD	793	397
Healthy	35	150
Pneumonia	37	100
URTI	23	150

→ After addressing the class imbalance, we used the LabelEncoder() to encode the target variable, making it compatible with DL models. Then, we split the data into training and testing sets in an 80:20 ratio. With the data prepared, it was time to implement some models.

MODELS IMPLEMENTED

1. Ensemble Classifiers

→ Before delving into Deep Learning models, at the suggestion of our mentor, we explored ensemble learning models. To our surprise, these models performed remarkably well in classifying the target classes. The evaluation metrics yielded impressive results, even in the context of multiclass classification. Here's a summarized table detailing the implemented models and their scores on different metrics:

Model Name	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	CV Scores on entire dataset (%)
Random Forest Classifier	95.76	95.5	95.5	95.5	95.29
Gradient Boosting Classifier	97.69	97.5	97.53	97.5	96.79
XG Boosting Classifier	98.16	98	98.01	98	96.69

→ All the above models were hypertuned first using GridSearchCV. It’s evident that the boosting algorithms outperformed the Random Forest model. Specifically, XGBoost demonstrated superior accuracy compared to other models, while Gradient Boosting slightly edged out XGBoost in 5-fold cross-validation accuracy.

→ In conclusion, the ensemble learning models, particularly boosting algorithms such as Gradient Boosting and XGBoost, proved to be highly effective in classifying the target classes. Their robust performance, especially after hyperparameter tuning, highlights their suitability for our classification task.

2. Deep Learning Models

→ Transitioning to Deep Learning models was our primary objective from the inception of the project. While we initially explored simpler neural networks, our ultimate goal was to develop a deep learning model, rather than relying solely on traditional machine learning or ensemble classifiers. This transition required extensive reading and hands-on experimentation to understand how to handle the data and determine the most suitable architecture.

→ After conducting thorough literature surveys and studying research papers, we decided to implement a CNN-LSTM architecture for classification. We have believed that this complex model will be adept at capturing intricate patterns in audio files, offering potential applications in future medical contexts.

→ Before diving into the CNN-LSTM architecture, we experimented with two other deep learning models: a pure LSTM model and a pure 1D CNN model. Both models underwent hyperparameter tuning to optimize their architectures. Here are the hypertuned architectures for these models:

LSTM Model Architecture:

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 189, 1024)         4202496

 dropout_1 (Dropout)         (None, 189, 1024)         0

 lstm_1 (LSTM)               (None, 189, 512)          3147776

 dropout_2 (Dropout)         (None, 189, 512)          0

 lstm_2 (LSTM)               (None, 189, 256)          787456

 dropout_3 (Dropout)         (None, 189, 256)          0

 lstm_3 (LSTM)               (None, 189, 128)          197120

 dropout_4 (Dropout)         (None, 189, 128)          0

 lstm_4 (LSTM)               (None, 189, 64)           49408

 dropout_5 (Dropout)         (None, 189, 64)           0

 lstm_5 (LSTM)               (None, 189, 32)           12416

 dropout_6 (Dropout)         (None, 189, 32)           0

 max_pooling1d (MaxPooling1  (None, 94, 32)            0
 D)

 flatten_2 (Flatten)         (None, 3008)              0

 dense_4 (Dense)             (None, 100)               300900

 dense_5 (Dense)             (None, 6)                 606

=================================================================
Total params: 8698178 (33.18 MB)
Trainable params: 8698178 (33.18 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

1D CNN Model Architecture:

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 187, 64)           256

 dropout (Dropout)           (None, 187, 64)           0

 flatten_1 (Flatten)         (None, 11968)             0

 dense_3 (Dense)             (None, 6)                 71814

=================================================================
Total params: 72070 (281.52 KB)
Trainable params: 72070 (281.52 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

→ Here's the final summarized table detailing the above two models and their scores on different metrics:

Model Name	Accuracy	Precision	Recall	F1-Score

LSTM	0.74	0.77	0.74	0.75
1D CNN	0.925	0.93	0.925	0.9275

→ The LSTM model didn't perform as well as we hoped. One reason might be its complexity. On the other hand, the 1D CNN, despite its simple design, achieved a high accuracy of 92.5% in classifying respiratory diseases.

FINAL MODEL

→ As mentioned earlier, our main goal was to develop a CNN-LSTM model. This model will combine 1D CNN layers initially, followed by some LSTM layers, then additional dense layers, and finally, a prediction layer. This architecture aims to leverage the strengths of both CNNs and LSTMs to effectively capture patterns in the data and improve classification accuracy. We have hypertuned the parameters of the model by checking out various numbers of layers of each type, playing the number of filters in CNN layers and number of dense layers in the final stage. The final architecture is as follows:

CNN-LSTM Model Architecture:

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_1 (Conv1D)           (None, 187, 128)          512

 conv1d_2 (Conv1D)           (None, 185, 64)           24640

 dropout_7 (Dropout)         (None, 185, 64)           0

 lstm_6 (LSTM)               (None, 185, 128)          98816

 dropout_8 (Dropout)         (None, 185, 128)          0

 lstm_7 (LSTM)               (None, 185, 64)           49408

 dropout_9 (Dropout)         (None, 185, 64)           0

 max_pooling1d_1 (MaxPoolin  (None, 92, 64)            0
 g1D)

 flatten_3 (Flatten)         (None, 5888)              0

 dense_6 (Dense)             (None, 100)               588900

 dense_7 (Dense)             (None, 6)                 606

=================================================================
Total params: 762882 (2.91 MB)
Trainable params: 762882 (2.91 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

→ The performance of this model on test set is as follows:

Model Name	Accuracy	Precision	Recall	F1-Score
CNN-LSTM	0.945	0.952	0.945	0.948

→ The accuracy of the model is 94.5%, with an F-score of 0.948. This CNN-LSTM model outperformed the previous two deep learning models when tested on unseen data. But wait, there's more! The exciting part is yet to come. Let's take a look at the normalized confusion matrix of the predictions on the test set:

→ From the normalized confusion matrix above, we can observe that the model achieved a perfect recall score for all target classes except for the 'COPD' class, which is the majority class among the target classes. This outstanding performance demonstrates the effectiveness of the CNN-LSTM hybrid model. It accurately predicted almost all classes, making it a promising model for future applications.

IMPLICATIONS

Our study presents significant implications for both research and practical applications in the realm of biomedical informatics, particularly in the domain of respiratory sound analysis. Through the comparison of Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) model, and a hybrid CNN-LSTM model on the respiratory sound database, we have uncovered valuable insights into the effectiveness of different deep learning architectures. Notably, our results revealed that the CNN-LSTM hybrid model outperformed both individual models, achieving an impressive accuracy of 94.5%. This underscores the importance of leveraging diverse architectures and their combinations to address complex classification tasks effectively.

The practical implications of our findings are profound, particularly in the development of automated systems for respiratory sound analysis in clinical settings. The high accuracy achieved by the CNN-LSTM hybrid model demonstrates its potential for real-world applications, such as automated diagnosis and monitoring of respiratory disorders. For instance, healthcare practitioners can utilize such advanced computational models to streamline diagnosis, improve patient care, and facilitate timely interventions. Moreover, our study highlights the importance of continued research and refinement of deep learning models to address the challenges associated with real-world biomedical data, such as variability in recording conditions and noise.

Furthermore, our findings underscore the importance of advancing computational methods for healthcare data analysis to enhance healthcare outcomes and patient care. By providing insights into the strengths and weaknesses of different deep learning architectures, our study lays the groundwork for future research and innovation in biomedical informatics. Continued exploration of hybrid architectures and optimization techniques holds promise for further improving classification performance and addressing the complexities inherent in biomedical data analysis. Ultimately, our research contributes to the ongoing efforts to harness the power of artificial intelligence and deep learning in healthcare to benefit patients and healthcare practitioners alike.

ASSUMPTIONS AND LIMITATIONS

→ While several classifiers could have been employed for this task, we have opted for the CNN-LSTM architecture. This decision is based on the expectation that this model configuration will yield robust performance. Leveraging the common practice of converting audio data into various types of image spectrograms, we anticipate that the CNN-LSTM model will effectively capture temporal and spatial features, thereby enhancing the classification accuracy of respiratory conditions

→ There is an assumption that the CNN-LSTM architecture has enough capacity to learn the complexities of the respiratory disease classification task without underfitting or overfitting.

→ To balance the dataset SMOTE technique is used to oversample minority classes. This might inadvertently learn biases present in the training data and SMOTE can be sensitive to noisy samples.

→ More hyperparameter tuning often means more complex models. These models might be harder to interpret and explain, which is crucial in many applications, especially in domains like healthcare or finance where decisions have high stakes.

→ Biases can occur if the model is trained predominantly on data from specific devices. It may not generalize well to new devices, leading to reduced performance in real-world applications.

TAKEAWAYS AND CONCLUSIONS

→ The application of deep learning models, particularly those based on respiratory sound analysis, shows promise in the accurate and early detection of various respiratory diseases. These models have the potential to assist healthcare professionals in timely diagnosis and intervention.

→ The developed deep learning models have demonstrated the capability to classify respiratory diseases into multiple classes, such as pneumonia, chronic obstructive pulmonary disease (COPD), and bronchiolitis. This multi-class classification approach enhances the diagnostic capabilities of the system.

→ The CNN and LSTM model , which we had developed, assists us in achieving optimal accuracy as well as aiding in the prediction of specific respiratory diseases using the audio files dataset.