8-24-2022 11:00 AM

Respiratory Pattern Analysis for COVID-19 Digital Screening Using AI Techniques

Annita Tahsin Priyoti, *The University of Western Ontario*

Supervisor: Haque, Anwar, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

Corona Virus (COVID-19) is a highly contagious respiratory disease that the World Health Organization (WHO) has declared a worldwide epidemic. This virus has spread worldwide, affecting various countries until now, causing millions of deaths globally. To tackle this public health crisis, medical professionals and researchers are working relentlessly, applying different techniques and methods. In terms of diagnosis, respiratory sound has been recognized as an indicator of one's health condition. Our work is based on cough sound analysis. This study has included an in-depth analysis of the diagnosis of COVID-19 based on human cough sound. Based on cough audio samples from crowdsourced COVID data, we develop an audio-based framework, deploying traditional Machine Learning (ML), Resampling multiclass ML approach, Cost-Sensitive Multiclass ML, and Multiclass Deep Learning (DL) approaches for COVID-19 digital screening. Our experimental results indicate that the resampling Multiclass ML approach shows the best result for COVID-19 digital prescreening with an AUC of 78.77%. To the best of our knowledge, this is the first COVID-19 detection tool that uses such diverse crowdsourced and largest physician annotated COVID data that uses patients' cough sound data to predict the presence of COVID-19 in those patients by applying multiple multiclass data balance techniques for AI algorithms. Our proposed novel framework and the developed tool will assist in a) automating COVID-19 digital pre-screening, b) making the COVID test more accessible and cost-effective, c) helping to detect an infected individual before a physical COVID test, and d) reducing the risk of infecting others.

Keywords

COVID-19, Crowdsourced Data, Audio Signal Analysis, Cough Sound, Cough Analysis, Cost-Sensitive ML, Machine learning, Deep Learning, Feature Extraction, e-Health.

Summary for Lay Audience

The COVID-19 epidemic has nearly brought the world to a halt since February 2020. Countries were placed on lockdown, millions of people died, healthcare facilities were overburdened, and the global economy saw one of its worst periods as a result of the epidemic. To address this issue, researchers throughout the world are investing in the development of a rapid, reliable, non-invasive diagnostic process. One of the most essential research initiatives is to employ coughs and their accompanying vocal biomarkers to diagnose COVID-19. In this thesis, we proposed a novel Artificial Intelligence (AI) based COVID-19 digital screening technique based on cough audio data from COVID patients. The proposed framework will aid in the promotion of contactless self-screening. Our model demonstrates promising results in detecting COVID-19 in real-time. This will make the COVID test more accessible and cost-effective and reduce the spread of COVID-19 by alerting patients, who will be able to self-isolate and help the public health authorities curb the spreading.

Acknowledgments

Dr. Anwar Haque, to my supervisor, thank you for your ongoing encouragement, which has aided my development as a researcher and, more significantly, as a critical thinker. I'd want to express my sincere gratitude to him, for providing me with the chance to work on such an interesting subject. Under his supervision, I watched myself grow the confidence to face the challenges of research and emerge with significant successes. The research journey is not easy and my journey started at the peak time of the pandemic, which made it even more difficult. But throughout my research, he always supported me and helped me deal with the difficulties that came along the path. He was all ears to my worries and guided me with any issues I had like a true mentor. Thank you for all of your efforts that have resulted in who I am today and for making my master's journey memorable.

I'd also want to express my gratitude to my family. My parents, grandmother, and brothers for always supporting me even though we are so far. My biggest support system here was my sister Aroni, in particular, who supported me during the uncertain period of the current epidemic and helped me to continue working on my thesis. Thank you for constantly believing in me and encouraging me even when I doubted myself.

I'd also want to thank all of my professors for their guidance during my studies and throughout my time as a teaching assistant. Throughout my classes and course work, I learned a lot and matured as a researcher. Each course I aided with during my master's degree helped me learn a lot, and the professors were extremely helpful and nice, which made the entire experience smooth.

I was fortunate enough to create some meaningful pals in a distant nation and new atmosphere. They assisted me without being asked, providing advice and support while I adjusted to my new surroundings. I'd also like to thank my friends back home and here for continuously checking up on me, supporting me, and wishing me luck. I'd also like to thank my labmates, who were always willing to help me expand my knowledge and were a go-to source for any kind of life or research-related concern.

Finally, I would like to express my sincere thanks to the University of Western Ontario for the chance to join its distinguished institute and further develop my intellect and skill set for ongoing endeavors.

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

The World Health Organization (WHO) has proclaimed the COVID-19 (SARS-CoV-2) virus a global pandemic, with over 10 million COVID cases. According to WHO figures, as of April 2022, there are about 6,159,474 fatalities worldwide [1]. This pandemic is changing the whole dynamics of socioeconomic relationships of people all over the world. It has affected all aspects of life such as health, work, business, educational environment, etc. This COVID-19 pandemic has compelled us to look at our way of work and life from a different perspective. Desperate times call for desperate measures, which have made all the communities do their best to mitigate the suffering. The COVID-19 pandemic has galvanized the scientific community to assist front-line medical personnel with ways of mitigation, detection, and prevention through top-notch research [2].

Due to a shortage of suitable supplies, qualified workers, and sample-processing equipment, testing capacity for COVID-19 remains a global concern. These issues are exacerbated in rural and underdeveloped regions. Frequent testing and quarantine on a large scale to minimize transmission is a highly successful strategy for controlling the virus's spread [3]. As a result, there is a necessity for diagnostic solutions that can scale worldwide.

WHO has already determined the key symptoms of COVID-19 are – fever, cough, and respiratory problems – as well as a recently updated list [4], these symptoms are non-specific and overburdening the healthcare system. The most prevalent symptom, fever, is

common for a wide range of diseases. But pairing it with a cough narrows the list of probable respiratory infections. Furthermore, the majority of COVID-19 infected individuals display none of the following symptoms (asymptomatic), yet they function as a catalyst in spreading the virus, making it more difficult to control [4] [5] [6].

The respiratory system is a key conduit via which humans cough and produce voice. As we all are aware of a common cold, respiratory illnesses may affect the sound of someone's breathing, coughing, and voice tone. Given that dry cough is one of the most prominent symptoms of this condition, cough sound analysis has emerged as an emergent study area for various respiratory-related diseases. It utilizes machine learning and signal processing methods to analyze coughs and identify COVID-19 in humans. Following this idea, we explored the COVID-19 signature in cough sounds by audio signal processing from a crowdsourced cough dataset and whether an AI model can detect it.

In this research, we looked at the use of human respiratory sounds as COVID-19 diagnostic markers in a crowdsourcing context. To the best of our knowledge, the dataset we used [7] is the largest crowdsourced public data collection of COVID-19 related cough sounds. Our study involved a novel digital technique for diagnosing COVID-19 illness based on human respiratory sounds i.e., coughing. Using audio cough data from Crowdsourced COVID data, we explored audio-based Machine Learning (ML), Resampling Multiclass ML approach, Cost-Sensitive Multi-Class ML, and Multi-Class Deep Learning (DL) models for COVID-19 digital screening.

1.1 Motivation

Strictly enforced tactics, together with contemporary testing, and the massive economic ramifications that have resulted, have proved sufficient to substantially reduce the number of affected people, but not to the point of eliminating the virus. Unless region-wide containment measures are maintained, these infections will be exceedingly difficult to manage using present diagnostic procedures. This is owing, in part, to the limitations of current viral and antibody testing, as well as a lack of complementary pre-screening approaches for deciding who should be checked. At the same time, COVID-19 clinical diagnosis can be time-consuming and expensive for patients, especially those living in underdeveloped regions with limited COVID-19 clinical resources [8]. According to a WHO report, almost two billion people globally struggle to obtain inexpensive access to essential pharmaceuticals [9]. According to report [10], 81% of COVID-19 carriers do not show severe enough symptoms to seek medical assistance, and instead behave as active spreaders. Other afflicted people exhibit severe symptoms only after many days of infection. These observations inspired us to work on this study issue to develop a new strategy centered on digital prescreening. This will allow for quick action to self-isolate without further spreading and offer medical attention to those who are most vulnerable. The COVID outbreak made worldwide people suffer as it is a highly infectious disease with a high death rate and after-effects. With the idea of contributing to mitigate the horrific sufferings, we were motivated to work on this research using Artificial Intelligence (AI) techniques. We examined the crowdsourced cough sounds and attempted to tackle the problem from several angles. Cough classification using sound is already a difficult problem. This is a multiclass unbalanced cough sound dataset, which raises the level of

difficulty even more. We used Resampling Approaches with ML, Cost-Sensitive ML algorithms, and Deep Learning (DL) algorithms with and without resampling techniques for classification.

1.2 Our Contribution

The followings are the main contribution of this thesis:

- In this thesis, we have thoroughly cleaned metadata as well as audio data and extracted features for COVID detection using multiple feature extraction techniques.
- The COUGHVID data was imbalanced and multi-classed. Both are challenges for ML classification and we applied the resampling approach, Cost-Sensitive classification approach, and Deep Learning algorithms with the resampling technique.
- We have done hyper-parameter tuning for the models to get the best result. Performance analysis and comparison were made using a different approach to come up with the best model for non-invasive COVID-19 prescreening.
- We classified the data into three categories which are COVID-19, Healthy cough data and others category (means symptoms of other respiratory diseases).
- From the application perspective, our proposed digital prescreening of the COVID-19 tool will assist in identifying infected individuals for isolation before the physical COVID-19 test. The developed prototype will make the COVID-19 test more accessible to people. Now patients must leave the house to get tested and

potentially infect others, so our digital prescreening will help to mitigate the possibility of infecting others.

- Last but not the least, the COVID-19 test is expensive [8], so prescreening COVID-19 will lessen the hassle for the patients of going for a swab test as they will be able to prescreen and get the result from the system within a short amount of time.

Chapter 2

2 Background

This chapter provides a high-level summary of background topics relevant to this thesis. This chapter is divided into four primary sections. Audio signal and processing techniques will be discussed in Section 2.1. Section 2.2 introduces audio data feature extraction and provides a brief overview of the four audio feature extraction methodologies. Section 2.3 will go through the field of machine learning and the ML methods utilized in this research. Section 2.4 will go through the concept of deep learning as well as the Deep Learning related techniques used to develop the prescreening tool.

2.1 Audio Signal

Audio is vibration. It is produced as a result of the movement of a sound source. The movement of air molecules propels it through the air or another medium. The movement of a diaphragm is caught by a receiver such as a microphone or human ears. These vibrations include information, known as the audio signal. Vibrations transmit the content of the speaker's message when someone speaks or produces sound. For example, when someone coughs, vibrations from the vocal-chord are generated, and this cough sound contains information.

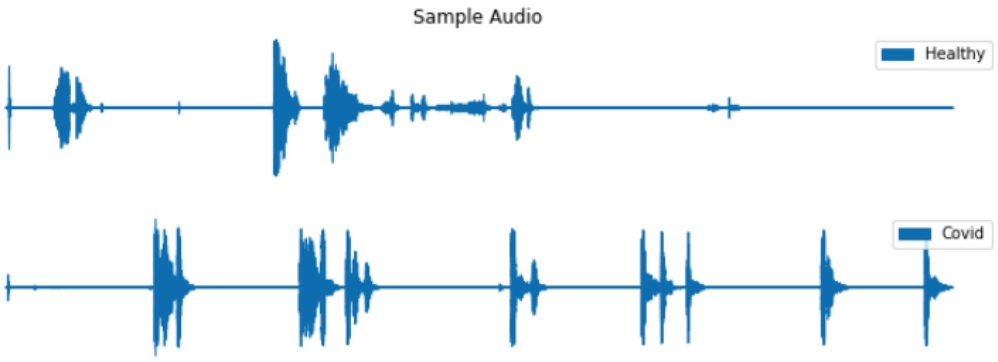


Figure 2.1: Audio signal representation of healthy audio and COVID audio.

2.1.1 Respiratory Sound

Respiratory sounds can also be referred to as lung noises or breath sounds. It is the unique noises made by the passage of air through the respiratory system [11]. Auscultation of the respiratory system with a stethoscope throughout the lung fields, as well as spectrum aspects of lung sounds, can be identified [12]. Typical breathing noises and coughing, wheezes, pleural friction rubs, stridor, etc. are among them.

2.1.2 Audio Segmentation

Audio segmentation is a technique for revealing semantically relevant temporal chunks of an audio source, sometimes known as auditory scenes. These scenes are the literary counterparts of paragraphs and may be fed into supervised or unsupervised audio classification and categorization algorithms. Using semantic indexes, auditory circumstances that are semantically related may be categorized.

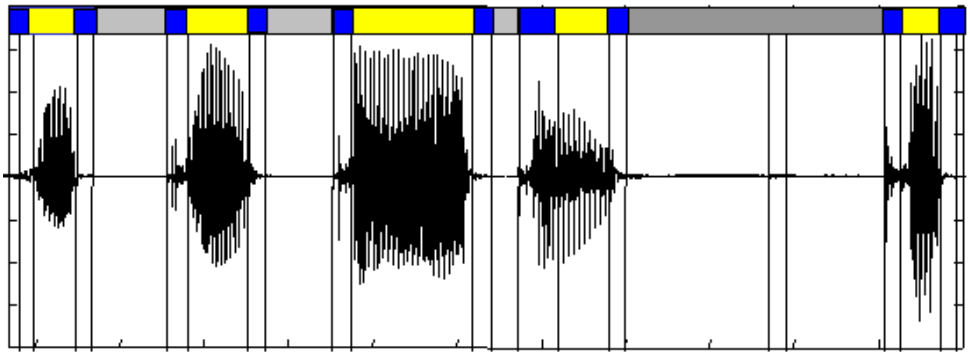


Figure 2.2: Audio segmentation representation [13].

2.1.3 Cough Sound Segmentation

Coughing is the body's protective strategy for cleaning the respiratory system of foreign things that have been inadvertently swallowed or that have been produced internally by infections [3]. It is a characteristic early to the mid-stage symptom of respiratory diseases such as pneumonia, TB, etc. Cough segmentation refers to the process of segmenting semantically significant scenes or pieces of a coughing sound. Coughing sound segmentation also aids in the extraction of vital information from coughing noises that may be utilized for analysis.

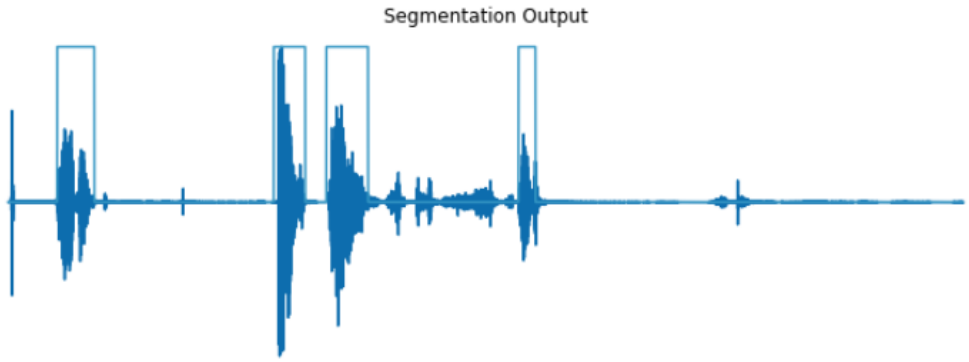


Figure 2.3: Segmented cough sound.

2.1.4 Audio Signal Processing

The employment of complicated algorithms and procedures for audio signals is referred to as audio signal processing. Audio signals are the representation of sound in the form of digital and analog signals. Our hearing range has a lower and maximum limit of 20 to 20,000 Hz [14]. Analog signals have electrical representations, whereas digital signals have binary representations. Converting digital and analog signals removes unwanted noise and balances the time-frequency ranges. It is primarily concerned with computational techniques for sound manipulation. It eliminates or reduces overmodulation, echo, and unwanted noise by employing a variety of processing techniques.

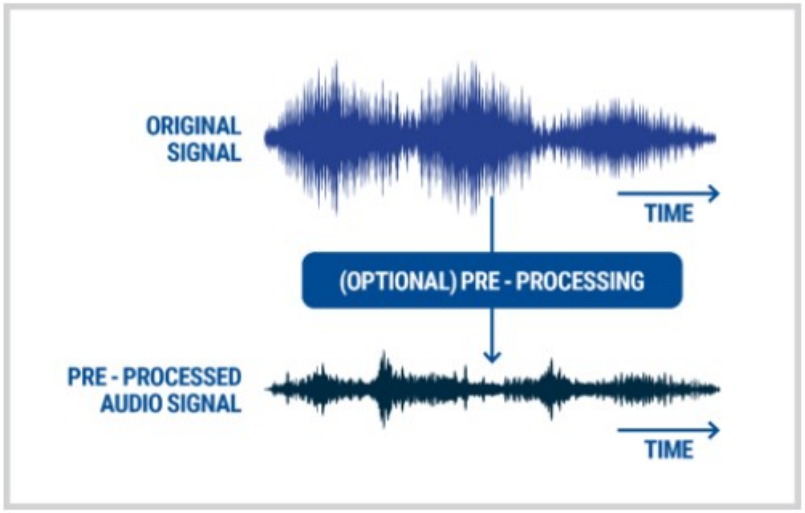


Figure 2.4: Before and after scenario of audio signal preprocessing [14].

2.2 Audio Data Features Extraction

Sound is represented as an audio signal with properties such as frequency, bandwidth, decibel, and so on. A typical audio signal is represented in terms of amplitude and time. The models cannot comprehend the audio data supplied directly, that's when feature extraction comes into the picture. Feature extraction is used to translate the audio data into

an intelligible format. It is a procedure that explains the majority of the facts, understandably. Spectrograms, Mel-spectrograms, and Mel-Frequency Cepstral Coefficients (MFCC) are a few examples of commonly used features extraction technique that outputs the features compatible with AI model architectures.

In this research, we have used four feature extraction techniques, which are briefly discussed below:

2.2.1 Spectrogram

A spectrogram is a very detailed and accurate representation of audio data [4]. It is a graphical depiction of the signal strength, or loudness of a signal as it changes over time at different frequencies contained in a certain waveform. In the spectrogram, each point is represented by colors, each point denoting the amplitude of those dots. As a result, a spectrogram illustrates amplitude fluctuations for each frequency component in the signal. Figure 2.5 represents an audio spectrogram that displays various emotions. The amplitude of emotions such as happiness and anger are illustrated through this spectrogram frequency ranging from 5000Hz to 150000Hz.

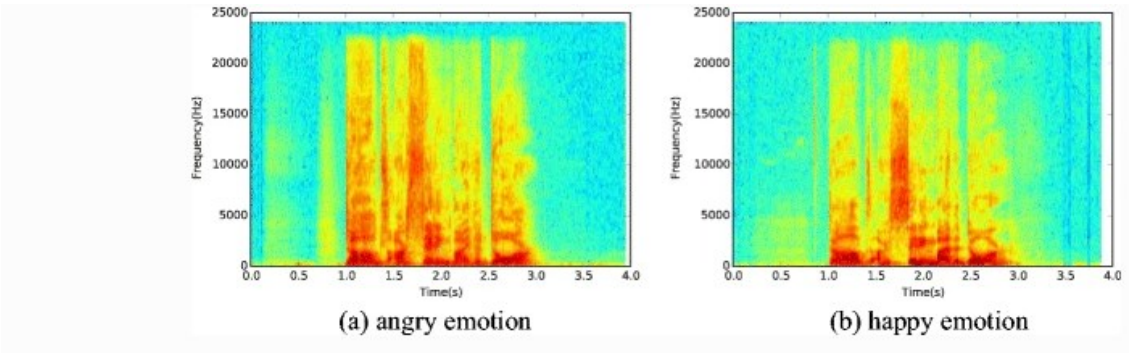


Figure 2.5: Representation of spectrogram of angry and happy emotions [4].

2.2.2 Short-Time Fourier Transform

The Short-Time Fourier Transform or Short-Term Fourier Transform (STFT) is a representation of a windowed signal's sequenced Fourier transforms (FT). A windowed signal is a weighted selection of a segment of a time waveform supplied by a window function for quick FT analysis. It is performed to reduce the amplitude of the discontinuities. STFT provides time-localized frequency information for frequency components of varying signals. The distinction between regular FT and STFT is that FT offers frequency information averaged throughout the whole signal time interval, but STFT does not. STFT is seen using a spectrogram, which is an intensity representation of STFT magnitude across time [5]. The STFT is a powerful and widely used general-purpose audio signal processing tool [15] [16] [17]. We get a particularly valuable class of time-frequency distributions [18] for each signal, with specified complex amplitude vs time and frequency [6]. The Mathematical Equation of STFT is [6]:

$$X_m(\omega) = \sum_{n=-\infty}^{\infty n} x(\mathrm{n}) w(n - mR) e^{-j\omega n}$$

$$= DTFT_{\omega}\left(x\ .\ SHIFT_{mR(w)}\right), \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.1)\ [6]$$

Where, $x(n) = input\ signal\ at\ time\ n$

$\mathrm{w(n)} = \text{Length M of window function}$

$X_m(\omega) =$ DTFT of windowed data centered on time mR

R = hop sizes in samples, between successive DTFTs

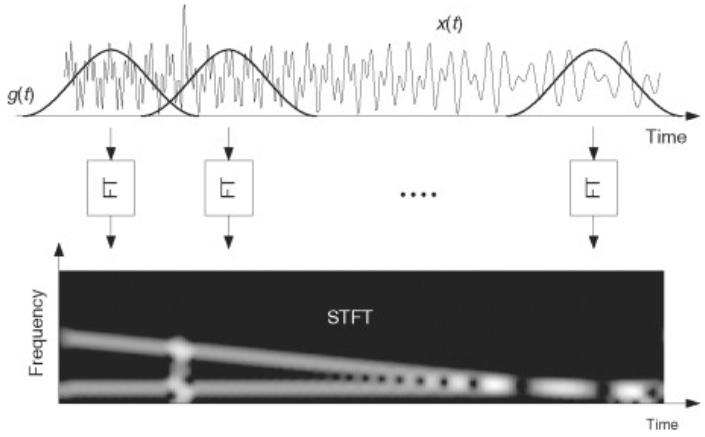


Figure 2.6: Illustration of STFT by taking Fourier transforms of a windowed signal [5].

2.2.3 Mel Scale

Melody is shortly used as "Mel". Mel scale is a way of frequency measurement. It distinguishes the distance between higher and lower frequencies for human beings [19]. For example, the difference between 300hz and 200hz is easily distinguishable by us. But the difference between 800hz and 900hz becomes a difficult task even though the difference is the same as 100hz for both the cases. As a result, even though this distance between the two sets of sounds is the same, our perception of it is not. The logarithmic change in the frequency of a signal is called Mel Scale [19]. The basic notion underlying this conversion is that sounds of equal distance on the Mel Scale are perceived by humans as being of similar distance. Because it reflects the human perception of sound, the Mel Scale is essential in ML applications for audio [19].

The formula of conversion from frequency to Mel scale is:

$$\boldsymbol{M(f) = 1125 \ln\left(1 + \frac{f}{700}\right)} \ldots\ldots\ldots (\mathbf{2.2}) \text{ [1]}$$

2.2.4 Mel-Spectrogram

The feature extraction technique that converts frequencies to the Mel scale is called the Mel spectrogram. It is a mixture of the Mel scale and the spectrogram that depicts a cough sound in both frequency and amplitude by time domains. Colors reflect the amplitude of a certain time. Figure 2.7 shows that brighter hues, from yellow to purple, correspond to increasingly greater amplitudes. The horizontal axis depicts the passage of time from left to right. The frequency is represented by the vertical axis, which ranges from low to high [20].

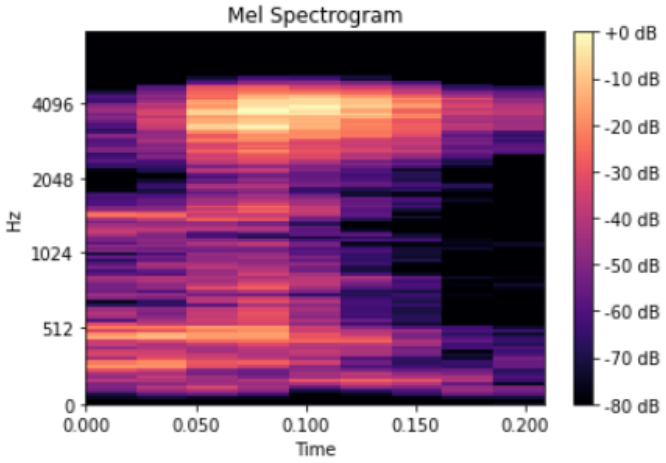


Figure 2.7: Mel-spectrogram of cough sound [20].

2.2.5 Mel-Frequency Cepstral Coefficients (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) is a way to convey spectrum information in a sound. The structure of the vocal tract can determine any type of sound produced by humans. The temporal power spectrum is a good representation of the vocal tract, and MFCC can do it well. Each coefficient represents a sound frame's value [21]. The fluctuations within each coefficient across the sound's frequency range are examined here.

Cepstrum denotes the rate of change of spectral bands. Periodic components provide prominent peaks in the appropriate frequency spectrum generated by FT (Fourier Transform) in terms of time signal analysis. This is seen in the graphic below. We begin by computing the log of the magnitude of the Fourier spectrum, and then we cosine transform the spectrum of this log to obtain cepstrum. In other words, cepstrum is the spectrum of the log of a periodic signal's spectrum.

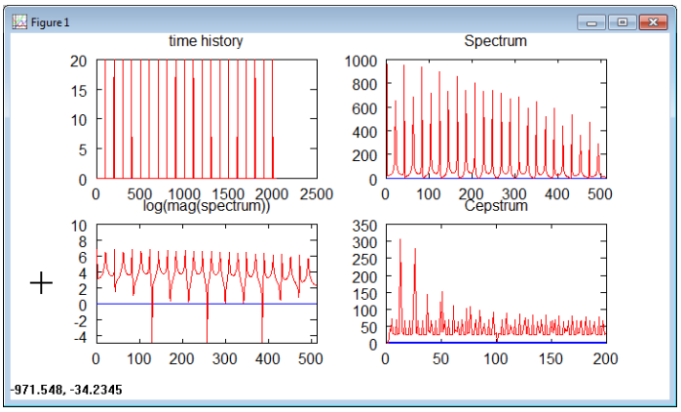


Figure 2.8: Step by step Cepstrum construction [22].

After the Cepstrum is constructed, conversion of the frequency to the Mel scale is needed. Obtaining the MFCCs entails analyzing and processing the sound in the following order [20]:

1. Separate the signal into frames and calculating the amplitude spectrum for each frame.
2. Calculate the log of these spectra also known as cepstrum.
3. Transform to the Mel scale.
4. Use the Discrete Cosine Transform (DCT).

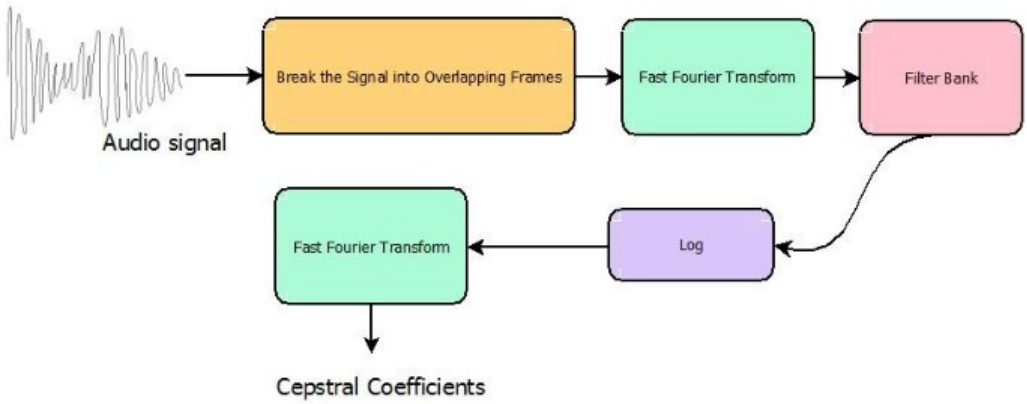


Figure 2.9: Flow diagram of MFCC [22].

Figure 2.9 shows the flow diagram of MFCC. The filter banks in the diagrams are the Mel filters used to convert the frequency to Mel scale and Cepstral coefficients are Mel-Frequency Cepstral Coefficients.

2.2.6 Chromagrams

The twelve separate pitch classes are referred to as chroma features or chromagrams in music. Chroma characteristics, which are resistant to changes in instrumentation, capture the harmonic and melodic aspects of music. Chroma-based feature extraction is an effective method for analyzing music that has been meaningfully categorized in terms of pitch and tuning that approximates the equal-tempered scale. In the extraction of chroma features, STFTs and Constant Q Transforms are applied [2].

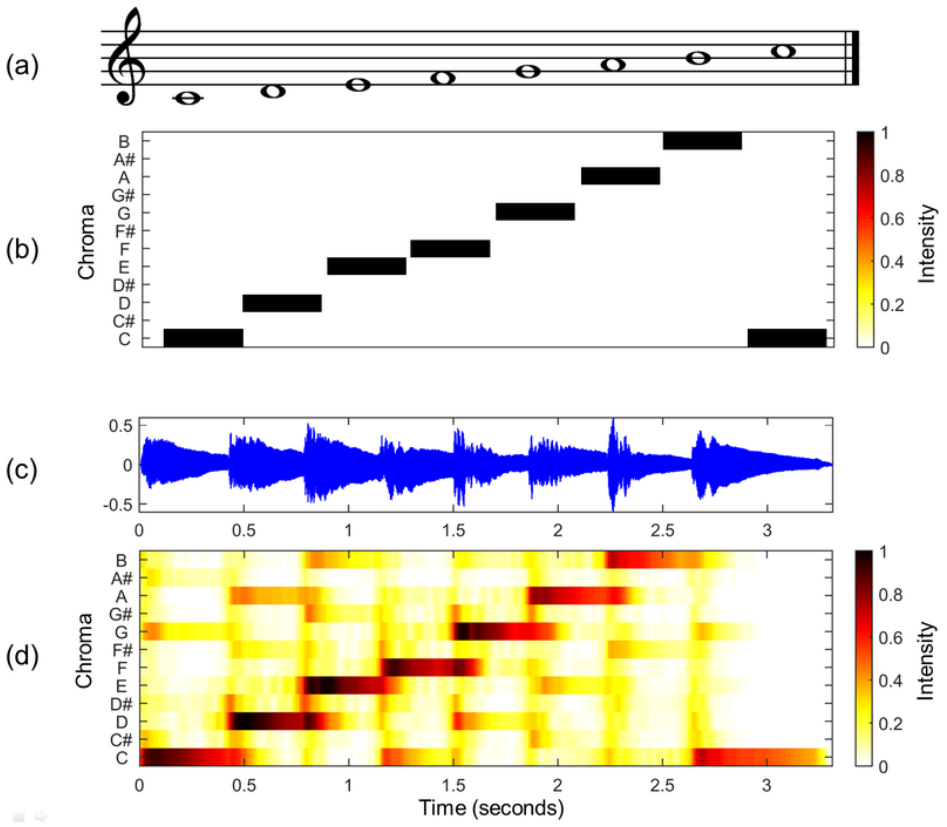


Figure 2.10: (a) The musical score for the C-major scale. (b) Produced chromagram from the musical score. (c) A piano recording of the C-major scale. (d) Audio recording generated chromagram [2].

We briefly reviewed audio signal, respiratory sound, cough sound segmentation, and audio data feature extraction techniques in Section 2.2, which are all relevant and important subjects for this thesis.

2.3 Machine Learning (ML)

Machine learning is a type of data analysis that automates the generation of analytical models [23]. It is a subfield of artificial intelligence (AI) predicated on the concept that information can be learned from given data by a machine. Moreover, ML will be able to detect patterns and make decisions without the need for human intervention. ML is a

fundamental component of data science, a rapidly expanding field. In data mining, statistical approaches are utilized to train algorithms to offer classifications or predictions, exposing valuable insights. Disease categorization and audio signal processing are difficult operations that can be automated with machine learning. Machine learning classifiers are classified into three types: Supervised ML (SML), Unsupervised ML (UML), and Semi-Supervised ML [23]. We applied 14 ML Algorithms in our research, and in this section, we provide a brief overview of supervised learning algorithms because they are an important element of this thesis.

2.3.1 Extra Trees Classifier (ET)

The Extra-Trees approach is used for solving classification and regression tasks. Pierre, Damien, and Louis presented this method in 2006 [24]. This algorithm uses the traditional top-down technique to build unpruned decision or regression trees. The main distinctions between ET and other tree-based algorithms are that it separates nodes by choosing cut-points entirely at random, and generates the trees using the whole learning sample. Aside from accuracy, the derived algorithm's key strength is computational efficiency [24].

This technique builds trees by employing a splitting procedure that takes into account two factors. The first is K, which denotes the number of chosen traits chosen at random at each node. Another factor is nmin, which stands for the minimum sample size randomly selected to divide a node. It is employed numerous times in conjunction with the (whole) original learning sample to construct an ensemble model. The outputs of the trees are combined to provide the final prediction, which is determined by a majority vote or arithmetic average based on the task.

The extra-Trees technique employs explicit randomization of the cut-point and attributes, paired with ensemble averaging, to reduce variation more effectively. To reduce bias, the original (whole) learning sample is utilized rather than bootstrap copies. The computational cost of the tree growth (assuming balanced trees) technique is in order of n log n with regards to learning sample size. However, in terms of the node splitting procedure's simplicity, the authors anticipate that the constant factor will be much smaller than in previous ensemble-based approaches that locally optimize cut-points.

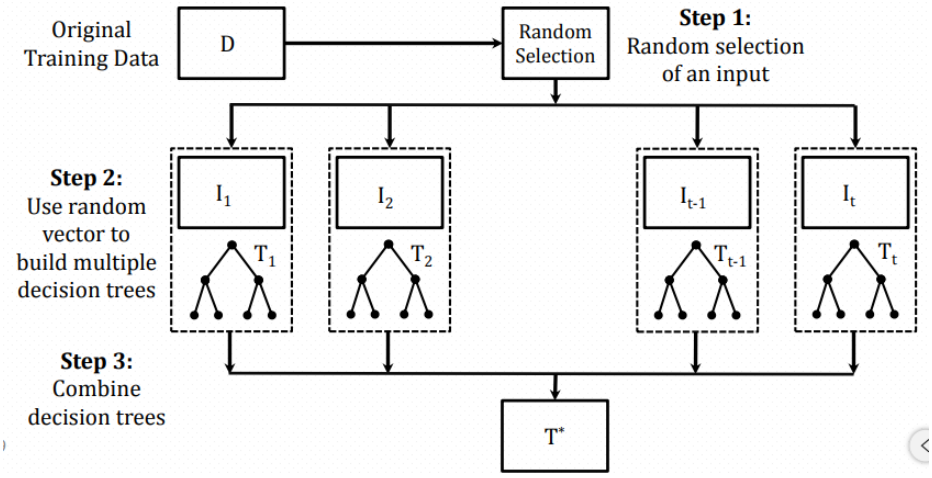


Figure 2.11: Visual representation of Extra Tree Classifier [25].

2.3.2 Random Forest Classifier (RF)

The Random Forest classifier (RF) is a widely used classification approach. It is an SML technique capable of resolving regression and classification problems. Because of its versatility and ease of implementation, RF is widely deployed. An RF is, like a forest, an ensemble of numerous decision trees. This method takes advantage of randomization to improve accuracy and prevent overfitting, a major challenge for advanced algorithms [26].

In RF, decision trees are generated by randomly selecting data samples and gathering predictions from each generated tree. If a dataset has "m" features, the RF method will select a random number of features, let's say "k," which denotes the number of features in the dataset where k<m. After that, root node is selected by calculating and selecting the node with the highest information gain. The method then separates the node into child nodes and continues the process "n" times, resulting in a forest with n trees. Finally, bootstrapping will be conducted, which is the process of merging the outcomes of all the decision trees in your forest [26]. RF has many advantages, such as being robust, fixing the overfitting problem, and being highly versatile.

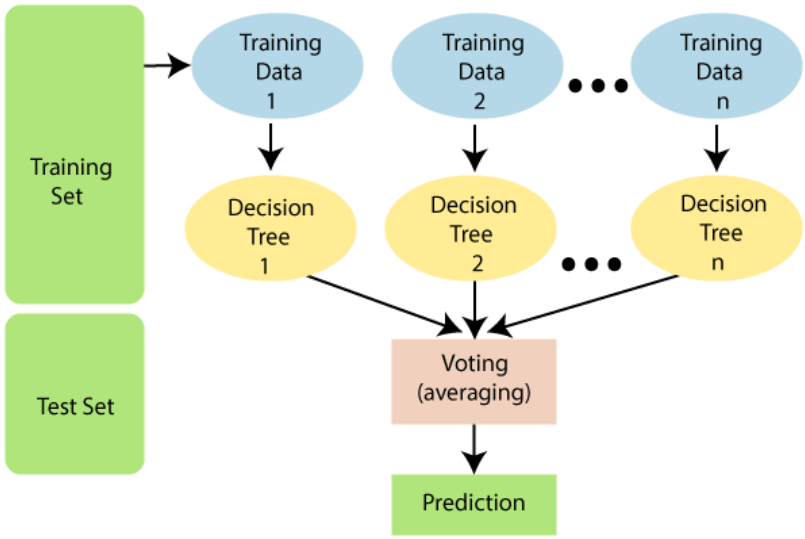


Figure 2.12: Workflow of Random Forest algorithm [27].

2.3.3 Decision Tree Classifier (DT)

Decision Tree is another popular classifier algorithm that we used in our work. DTs are created by aggregating the data on which the model must be trained. This training dataset is constantly being divided into smaller groups of data. This technique is supported by the

construction of a linked tree, which is built progressively as the data is broken down. The decision tree is composed of three major elements: nodes, edges, and leaf nodes [28].

The following are the three elements:

1. Nodes: The value is tested here, the value of a certain attribute is supplied, and the values are verified and tested against the values to make a choice.
2. Edges: Edges are in charge of the outcome of any test results as well as connecting two separate nodes or leaves.
3. Leaf-Nodes: These nodes are known as terminal nodes since they determine the outcome.

The decision tree operates in a few steps, which are as follows:

a. DT starts the branching process by first specifying a test for the root node. While constructing branches, each conceivable outcome of the stated trial is considered.
b. DT divides data instances into smaller subsets. Each branch has a slice that connects to the node.
c. Repeats the operation for each branch, using the branch's instances.
d. When all of the instances belong to the same class, the recursive process comes to an end.

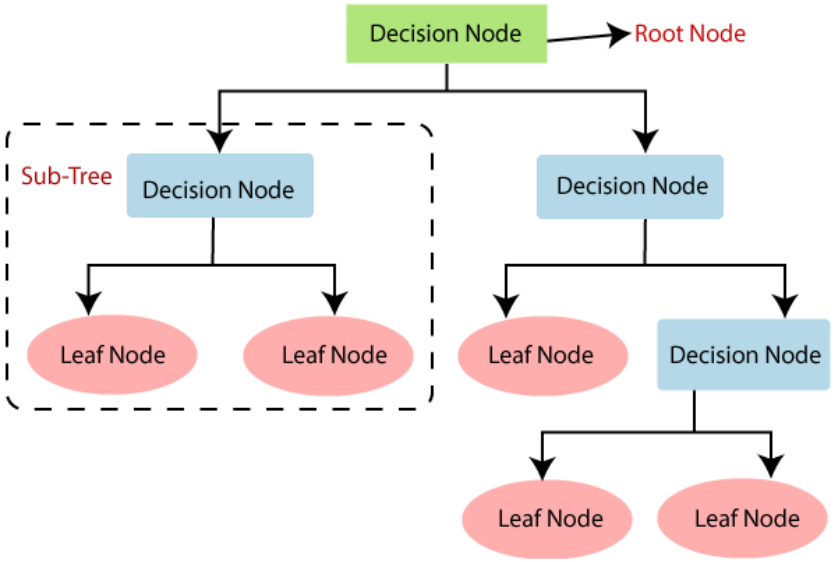


Figure 2.13: Decision Tree Algorithm workflow diagram [29].

After the learning phase, the algorithm finishes building a DT based on the provided training dataset and provides the tree to the user. DT has various advantages, including being cost-effective and efficient, having relatively similar prediction accuracy, and being capable of removing irrelevant characteristics [28].

2.3.4 K-Nearest Neighbor Classification (K-NN)

The K-Nearest Neighbor algorithm (K-NN) is an ML approach that uses the supervised learning paradigm. The K-NN method is based on the idea that equivalent items are close to one another. As a consequence, the K-NN algorithm identifies the values of the new points based on the feature similarity to the points in the training set. In essence, the KNN approach assigns a value to the most recent point based on resemblance with the points in the training set. This algorithm may be used for classification as well as regression applications. K is a key parameter in the KNN algorithm. The value of K is typically calculated using error curves [30].

The following is KNN's workflow [30]:

a. Selects the closest data points as the value of K.
b. Determine the distance between K neighbors. The most often used method for calculating distance is The Euclidean technique.
c. Picks the K nearest neighbor using the Euclidean distance technique and counts the total amount of data points in each category among the nearest K-neighbors.
d. The new data point is assigned to the category that has the most neighbors. Once all of the data points have been classified, the model is complete.

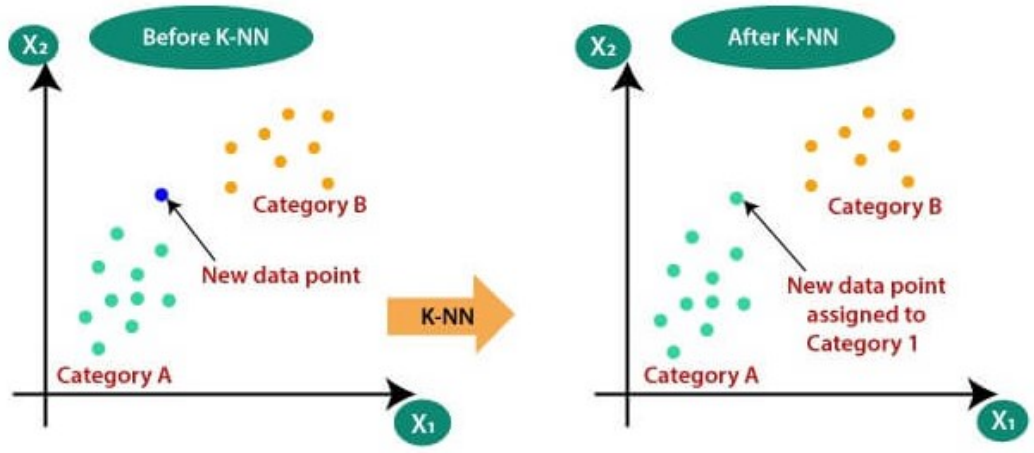


Figure 2.14: The placement of new datapoint before and after applying the KNN algorithm [30].

The benefits of applying KNN are that it is simple to construct and comprehend, it does not require training time, thus it is rapid, it develops in response to new data, and hyperparameter tuning is simple.

2.3.5 Logistic Regression (LR)

In our research, we used a Logistic Regression approach. It is a classification process that is used to assign observations in a dataset to one of a specified number of classes. LR uses the sigmoid function to alter the output and returns a probabilistic value. This algorithm's hypothesis $h_\theta x$ is that the cost function limit is between 0 and 1. As a result, linear functions cannot express the outcome. The potential of receiving a result that is more than 1 but less than 0 can't be achieved by a linear function [31].

Expected Logistic Regression Hypothesis:

$$0 \leq h_\theta x \leq 1 \ldots\ldots\ldots\ldots \text{(2.3) [32]}$$

Conversion of expected values to probabilities is done using the sigmoid function. Any real number can be converted into another value between 0 and 1 by this function. This function is used in ML to translate predictions to probabilities. LR is basically for binary classification but it can be extended for multiclass classification. This is why we employed it in our multiclass problem. The Logistic Regression equation and the Linear Regression model are quite similar. If we consider a model having a single predictor "x" and one Bernoulli response variable "ŷ" where probability p = 1. The linear equation is expressed as follows:

$$\hat{y} = p = b_\theta + b_1 x \ldots\ldots\ldots\ldots\ldots\ldots \text{(2.4) [31]}$$

In equation (2.4), the linear equation is ($b_0 + b_1 x$) and is capable of holding values exceeding the range (0,1). But we know probability will always be in the range of (0,1).

From a linear equation and with the help of the sigmoid function a logistic regression models formed.

Sigmoid function: $\sigma(z) = 1 / (1 + e^{-z})$……...(2.5) [31]

LR model: $\hat{Y} = \sigma(b_0 + b_1x) = 1 / (1 + e^{-(b0 + b1x)})$……..(2.6) [31]

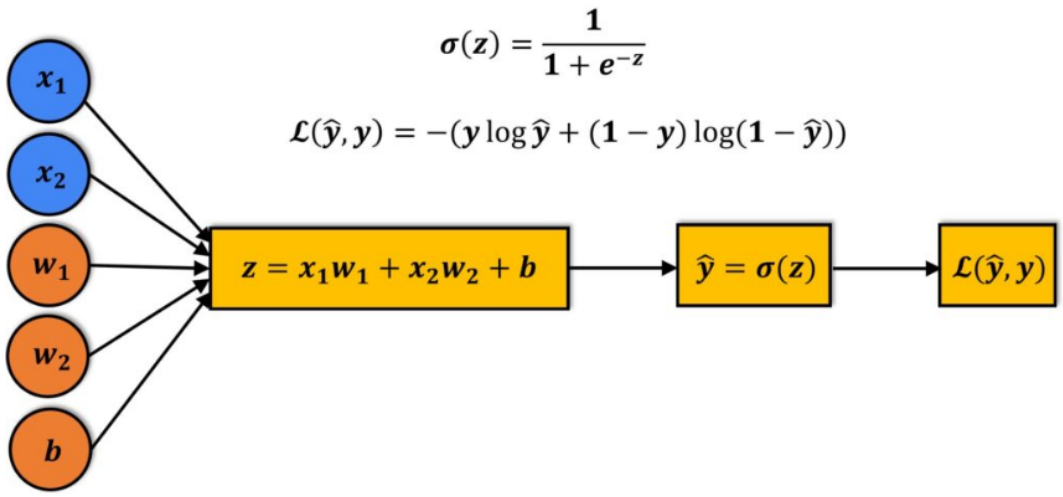
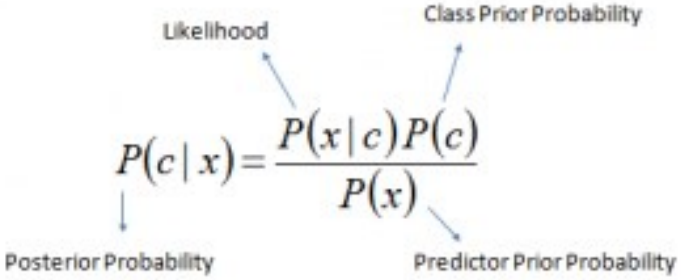


Figure 2.15: Workflow of Logistic Regression [31].

2.3.6 Naive Bayes (NB)

Naive Bayes (NB) is a classification method that is based on the Bayes' Theorem and implies predictor independence [33]. An NB classifier suggests that each feature in a class is independent even when other features are present. The NB model is simple to build and highly effective for very large data sets. Aside from simplicity, this technique has been proved to outperform even the most sophisticated categorization systems. We can use the Bayes theorem to get the posterior probability P(c|x) given P(c), P(x), and P(x|c). Consider the following formula:



$$P(c|x) = P(x_1|c) \times P(x_2|c) \ldots \times P(x_n|c) \times P(c) \quad \ldots..(2.7)$$

Figure 2.16: Bayesian Theorem equation [33].

P(c|x) specifies the posterior probability for target class c in equation 2.7, P(c) and P(x) are the class prior probability and predictor prior probability, respectively. P(x|c) signifies the likelihood, which is the likelihood of a predictor in a specific class. The predictor (x, properties) is given. The NB algorithm technique will be discussed more below [33]:

- Step 1: Create a frequency table from the data collection.
- Step 2: Calculate probabilities and make a Likelihood table
- Step 3: For each class, we will determine the posterior probability using the NB equation. The forecast is the highest posterior probability value of the class.

The NB method makes predicting the class of test data set simple and fast. This approach is also effective in multi-class prediction. When the independence requirement is met, an NB outperforms conventional models like logistic regression and takes less training data [33].

2.3.7 Gradient Boosting Classifier (GB)

The gradient boosting (GB) approach is one of the most effective in machine learning. In ML algorithm errors are widely classified into two categories. The first is bias errors, while the second is variance errors. To reduce the model's bias error, GB is employed as a boosting approach [34].

SL tasks have an output variable y and a vector of input variables x, which are linked by a probabilistic distribution. The basic objective is to identify the function $\hat{F}(x)$ that provides the best approximation for the output variable given the input variables. This is formalized by introducing and minimizing a loss function $L(y, F(x))$:

$$\hat{F} = arg_F \min E_{x,y}[L(y, F(x))] \quad \text{.....(2.8) [35]}$$

The GB algorithm looks for an approximation in form of a weighted sum of functions $h_i(x)$. This works with value y for some H class, which is known as base learners.

$$\hat{F}(x) = \sum_i^M \gamma_i h_i(x) + const. \quad \text{.....(2.9) [35]}$$

A training set of x values and associated y values are provided, such as$\{(x_{1,}y_{1,}), \ldots\ldots, (x_{n,}y_{n,})\}$. Based on the empirical risk minimization concept, this strategy seeks an approximation $\hat{F}(x)$ that minimizes the loss function's average value on the training set, i.e., the empirical risk. This begins with a model with a constant function $F_0(x)$ and gradually expands it in a greedy manner [35]:

$$F_0(x) = arg_\gamma \min \sum_i^M L(y_i, \gamma), \quad \text{.....(2.10) [35]}$$

$$F_m(x) = F_{m-1}(x) + arg_{h_m \in H} \min [\sum_{i=1}^{n} L(y_i, F_{m-1}(x_i)) + h_m(x_i)] \text{.....(2.11) [35]}$$

where $h_m \in H$ is a base learner function. The benefits of employing GB include unrivaled prediction accuracy, a broad range of flexibility (including multiple hyperparameter tuning options), a very flexible function fit, no data pre-processing required, and the ability to manage missing data [36].

2.3.8 Light Gradient Boosting Machine (LGBM)

The LightGBM framework is a GB framework based on the DT algorithm. This approach is known as light because of its incredible categorization speed. LGBM is also noted for its high performance. This algorithm is utilized in a range of machine learning applications such as ranking and classification [37]. Based on DT algorithms, this algorithm splits the tree leaf-wise with the best fit. LGBM is superior to other boosting methods since it divides the tree leaf-wise rather than depth-wise or level-wise. In LGBM, the leaf-wise approach may reduce more loss than the level-wise technique when constructing on the same leaf. As a consequence, gives far more precision than any of the existing boosting strategies can seldom achieve.

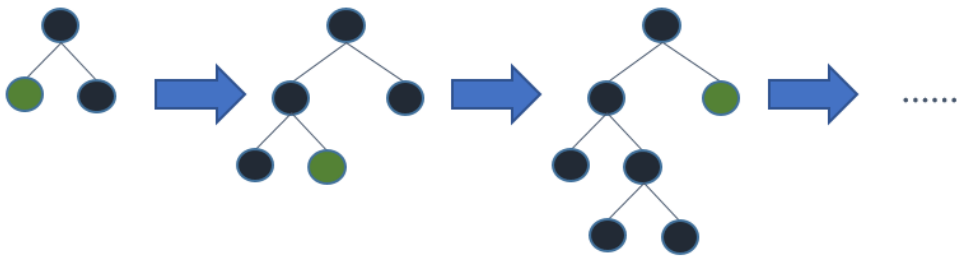


Figure 2.17: Leaf-wise tree growth in LightGBM [37].

Gradient-based LightGBM One Side Sampling Technique (GOSS):

In estimating information gain (IG), various data instances play distinct roles. Larger gradient instances will contribute more to knowledge gain. To maintain the accuracy of IG estimation, GOSS keeps cases with large gradients (e.g., in the top percentiles) and removes instances with small gradients at random. This approach can produce a more accurate gain estimate than uniformly random sampling at the same intended sample rate. This happens especially when the quantity of information acquired varies widely [38].

LGBM provides the advantages of fast training time, compatibility with large datasets, and high efficiency. This approach utilizes less memory and has more accuracy than any other boosting algorithm because of the leaf-wise split technique. LGBM also allows for parallel learning [37].

2.3.9 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a data visualization and classification tool. LDA is also known for its dimension reduction technique. Despite its ease of use, LDA typically produces good, fair, and interpretable classification results. When dealing with real-world classification difficulties, LDA is often used as a benchmarking approach before implementing more complicated and flexible ones.

LDA's development as a supervised classification algorithm: We will examine a general classification problem in which a random variable x with density f_k(x) on R^p belongs to any of the K classes. A discriminant rule separates the data space into K distinct regions starting with R_1 and ending with R_K. This is representative of all classes. In these areas,

discriminant analysis essentially implies that we assign x to j class if x is in the j region. We may use two allocation rules to determine which area the data x belongs to. Rules are:

1. Maximum likelihood rule: Each class has an equal chance of occurrence, so we can assign x to class j.

   $j = argmax_i f_i(x)$……..(2.12) [39]

2. Bayesian rule: If the class prior probabilities given as $\pi_{1,\ldots}\pi_K$, then we can allocate x to class j if

   $j = argmax_i \pi_i f_i(x)$……(2.13) [39]

Now if we consider that the data is from a multivariate Gaussian distribution, i.e. x ~ N(μ, $\sum$), explicit formulations can derive from the aforementioned allocation rules. We classify x to class j using the Bayesian criterion if in equation (2.13), where $\delta_i(x)$ is,

$\delta_i(x) = log f_i(x) + log\pi_i$……..(2.14) [39]

Equation 2.13 is known as the discriminant function. Suppose a set of x given where two discriminant functions have the same value, i.e., $\{x: \delta_k(x) = \delta_l(x)\}$, is the decision border separating any two classes, k, and l. So, any data that falls on the decision border has a high chance that is coming from one of the two groups. For LDA, the assumption is equal covariance across K classes, i.e., $\sum_1 = \sum_2 = \cdots = \sum_k$. The discriminant function will be:

$\delta_k(x) = x^T \sum^{-1} \mu_k - \frac{1}{2}\mu_K^T \sum^{-1} \mu_k + log\pi_k$……..(2.15) [39]

This is a linear function that varies with x. As a result, the decision boundary between any two classes is a linear function of x, giving birth to the phrase "linear discriminant analysis."

The benefits of LDA include its ease of interpretation and application, as well as the classification's robustness and dimension reduction [39].

2.3.10 Quadratic Discriminant Analysis

Quadratic discriminant analysis (QDA) is similar to linear discriminant analysis (LDA), in which measurements from each class are considered to be regularly distributed [40]. QDA does not make assumptions that the covariance of each class is the same. When the normalcy assumption is true, the likelihood ratio test is the most feasible test for the hypothesis that a measurement (given) belongs to a particular class. Assume there are just two groups, each with its own set of means μ_0, μ_1 and covariance matrices $\sum_0$, $\sum_1$ corresponding to y = 0 and y = 1 respectively [40]. For some t-value, the likelihood ratio is given by:

$$\text{Likelihood ratio} = \frac{\sqrt{2\pi|\sum_1|}^{1} \quad \exp\left(-\frac{1}{2}(x-\mu_1)^T \sum_1^{-1}(x-\mu_1)\right)}{\sqrt{2\pi|\sum_0|}^{1} \quad \exp\left(-\frac{1}{2}(x-\mu_0)^T \sum_0^{-1}(x-\mu_{01})\right)} < t \quad \text{.....(2.16) [40]}$$

After some rearrangement, the resultant separation surface between the classes may be shown to be quadratic. The population values will be replaced by sample estimates of the mean vector and variance-covariance matrices in this approach.

2.3.11 AdaBoost Classifier

AdaBoost is an ensemble learning strategy (sometimes known as "meta-learning") designed to improve the performance of classifiers. AdaBoost is an iterative technique to learn from weak classifier errors and turn them into strong ones.

The Mathematics Behind AdaBoost [41]:

We will try to break down AdaBoost step by step and equation by equation to make it understandable. Let us begin by examining a dataset with N points, or rows. At first, all data points will be assigned the same weighted sample w:

$$w = \frac{1}{N} \in [0, 1] \text{.....(2.17) [41]}$$

N denotes the total amount of data points. Each weight has a value between 0 and 1. The sum of the weighted samples is 1. Following that, we employ the following approach to compute the true effect of this classifier in classifying the data points:

$$\alpha_t = \frac{1}{2} ln \frac{1 - TotalError}{TotalError} \text{.....(2.18) [41]}$$

The alpha value shows the impact this stump might have on the final classification. Total Error is just the total amount of misclassifications in the training set divided by its size. By entering multiple Total Error-values ranging from 0 to 1, we can get a graph for Alpha.

If a decision stump performs well means no misclassifications, the error rate is 0 and the alpha value is quite significant and positive. However, if the stump identifies half correctly and half erroneously, the error rate is 0.5, which is not excellent, and the alpha value is 0. Finally, in case the stump continuously keeps producing misclassified results, then the

alpha would be a large negative number. It's time to modify the sample weights, which were previously set to 1/N for each data point, after putting in the actual Total Error-values for each stump. The following formula is used to do this:

$$w_i = w_{i-1} * e^{\pm\alpha} \text{.....(2.19) [41]}$$

As a result, when multiplied by Euler's number, the new sample weight equals the previous sample weight. It will be increased to a value of plus or minus alpha. Although AdaBoost is often used to combine weak base learners (such as decision stumps), it has also been shown that it may effectively combine strong base learners (such as deep decision trees). This results in a more accurate model [42].

AdaBoost offers several advantages, the most important of which is that it is simpler to use, requiring less parameter tuning than methods such as SVM. this algorithm may also be used in conjunction with SVM as an added advantage. Another advantage is, not prone to overfitting in theory, although there is no real evidence for this. This may be since parameters are not simultaneously optimized. Stage-wise estimation slows down the learning process [41].

2.3.12 Ridge Classifier (RC)

Ridge Regression (RR) is a model tuning approach that is used to examine multicollinear data. This method is used for L2 regularization. When there is an issue with multicollinearity, least-squares are not biased and with huge variances, which leads to the predicted values that are far from the actual values [43].

RR's cost function can be written as:

$$\text{Min } (||Y - X(\text{theta})||^2 + \lambda||\text{theta}||^2) \quad \text{.....(2.20) [43]}$$

Lambda is the penalty terminology. The supplied value is specified by an alpha parameter in the ridge function. By changing the value of alpha, we may control the penalty term. Because the values of alpha are proportional to the penalty, the size of the coefficients is lowered, when alpha is high. It reduces the parameters. As a result, it is utilized to avoid multicollinearity. Shrinking the coefficients also minimizes the model's complexity. The standard regression equation serves as the foundation for the regression ML model, which is stated as:

$$Y = XB + e \quad \text{.....(2.21) [43]}$$

In equation (2.21), the dependent and independent variables are denoted by Y and X respectively. The regression coefficients are B and e, respectively represents the errors and residuals.

The RC technique is based on the RR method and converts the label data between the range -1 to 1, before using the regression method. When dealing with multiclass data, the class with the highest prediction value is chosen as the target class, and multi-output regression is applied [44].

2.3.13 SVM - Linear Kernel

A support vector machine (SVM) algorithm is quite popular because of producing high precision while requiring few computer resources. SVM is used for both regression and classification. However, it is commonly employed in classification problems.

The purpose of the SVM method is to find a hyperplane in n-dimensional space. Where n is the total amount of features that classify the data points. Many different hyperplanes might be utilized to divide the two sets of data points. The aim is to find the plane with the largest margin, or the maximum distance between data points from both classes. Increasing the margin distance allows successive data points to be classified with more accuracy [44].

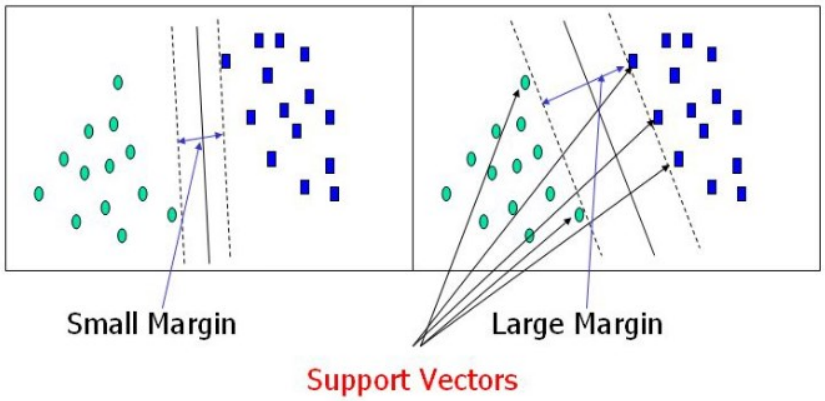


Figure 2.18: SVM algorithm hyperplanes separating classes [44].

The Linear Kernel is used when the data is Linearly separable, that is, it can be split using a single line. It's a commonly used kernel. This is frequently used when a data collection has a significant number of characteristics [45].

SVM offers several advantages, including the ability to perform well on a variety of datasets, being adaptable due to the availability of alternative kernel functions, and working well with both high and low dimensional data [46].

2.3.14 Dummy Classifier

A dummy classifier is a classifier model that predicts without looking for patterns in the data. The default model simply looks at whatever label appears more frequently in the

training dataset and predicts based on that label. However, before we can develop a dummy classifier, we must first understand how to compare the model in hand to the Dummy Classifier.

To really understand and then enhance the performance of our model, we must first build a baseline for the data that we have. It is critical to establish a baseline model against which we can assess the performance of our model, and here is where the Dummy Classifier comes into play [47].

2.4 Deep Learning

Deep learning (DL) may be viewed as a sub-branch of ML. The concept is built on self-learning and improvement through the examination of computer algorithms. DL, as opposed to ML, works with artificial neural networks (ANN), which are supposed to mimic how people think and learn. Until recently, neural networks were restricted in complexity due to computer power constraints. DL, as opposed to ML, employs ANN which is designed to replicate the mechanism of thinking and learning just like humans. Until recently, the complexity of neural networks was limited owing to computer power limits. With improvements in Data analytics, DL has allowed larger, more powerful neural networks, allowing powerful machines, to comprehend, and respond to complex situations faster than humans. Deep learning has aided image classification, language translation, and speech recognition. DL is capable of handling any pattern recognition problem without any human intervention.

Deep learning is the most rapidly increasing machine learning area. A growing number of firms are turning to DL to help them create innovative business models. This section will go through the Dl models that we used in our research.

2.4.1 Sequential Model

The technique of constructing a series of values from a collection of input values is known as sequence modeling. These input values might be time-series data, which shows variables, such as product demand, changes over time. The output might be a forecast of future demand.

The data for sequence models are not samples that are dispersed independently and identically. Furthermore, due to the sequential sequence, the data are dependent on each other. Sequence models are very useful for a wide range of complicated applications, including speech/voice recognition, time series prediction, and natural language processing [48].

2.4.2 Long Short Term Memory (LSTM)

The Long Short Term Memory (LSTM) network is a kind of Recurrent Neural Network (RNN) [49]. The previous step's output works as the input for the upcoming step in RNN. LSTM was developed by Hochreiter and Schmidhuber [49]. This algorithm points out the shortcomings of RNN regarding long-term dependency. RNN is incapable of predicting words stored in long-term memory but capable of making accurate predictions based on the current input. RNN's performance drops when the gap length rises. As a result, the LSTM is constructed in such a way that information may be preserved for an extended period. LSTM is used to process time-series data, predict, and classify it.

The LSTM is designed in a chain structure. The architecture includes four neural networks and multiple cells also known as memory units. The cells are used for storing information and memory modification is done by the gates.

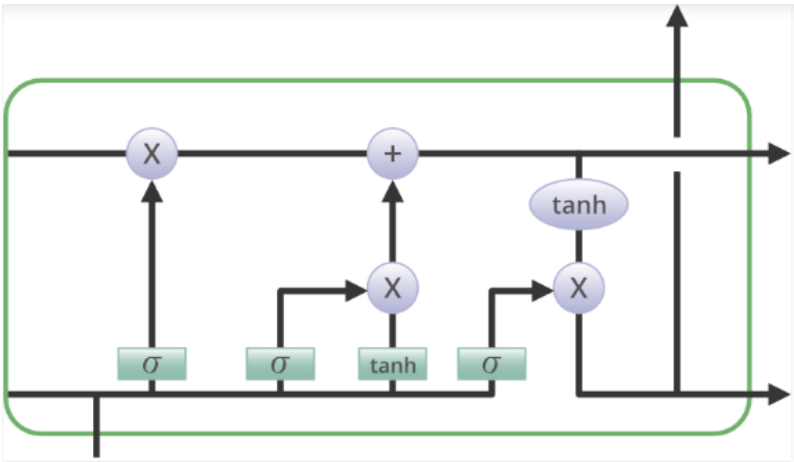


Figure 2.19: LSTM full architecture [49].

This architecture consists of three gates:

1. Forget Gate: The forget gate eliminates extraneous information from the cell state. At the specific moment, x_t and h_t-1are respectively input, and preceding cell output are given into the gate. The inputs are then multiplied by weight matrices, and the bias is added subsequently. The activation function then receives the results and creates a binary output. The output is stored or discarded depending on the cell state value. If the value is 0, the output is lost; if the value is 1, the output is saved.

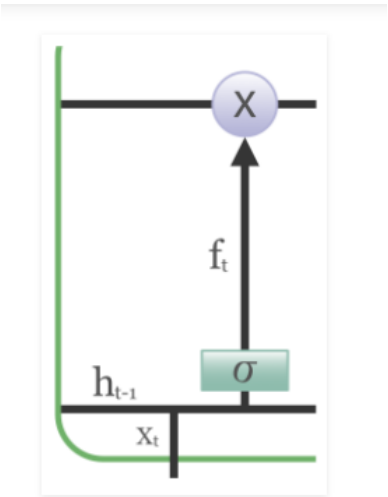


Figure 2.20: Forget Gate of LSTM [49].

2. Input gate: The input gate is responsible for contributing critical information to the cell state. The sigmoid function is used to control the flow of information. Using the h_t - 1 and x_t inputs, this function selects the values that will be saved and remembered. The tanh function is then used to build a vector containing all of the possible values from h_t - 1 and x_t. The range of potential values is -1 to +1. Lastly, the values both vector and regulated ones get multiplied to provide relevant information.

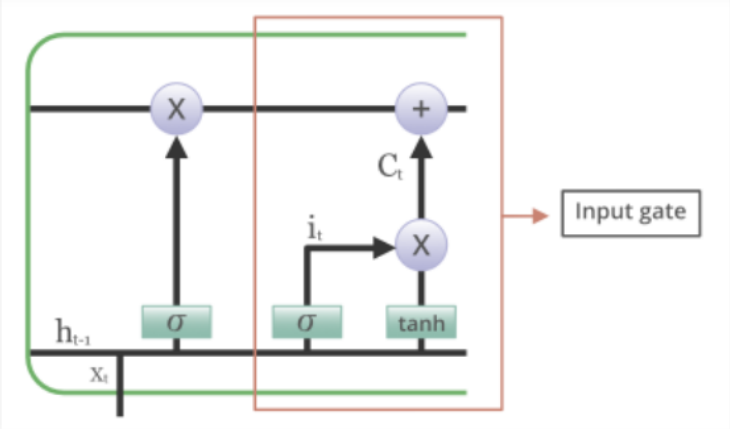


Figure 2.21: Input Gate of LSTM [49].

3. Output gate: The output gate extracts pertinent information from the current cell state and displays it as output. In the first stage, the cell uses the tanh function to generate a vector. The sigmoid function then controls and filters the h_t-1 and x_t input values to be remembered. Finally, the vector and controlled values are multiplied and supplied as output and input to the next cell, respectively.

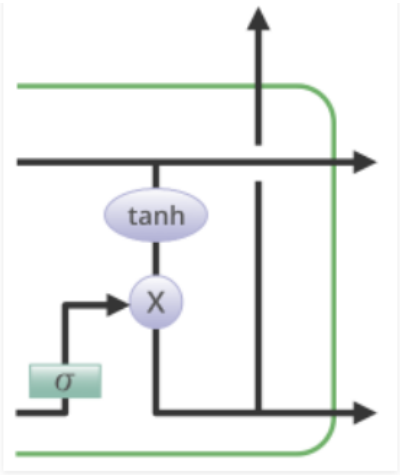


Figure 2.22: Output Gate of LSTM [49].

There are various benefits of utilizing LSTM. LSTM, for example, provides a wide variety of parameters, such as learning rates and input and output biases, no precise changes are required, and the load of updating each weight is decreased.

Chapter 3

3 Related Works

Researchers and medical experts around the globe are working tirelessly to find effective and economically viable COVID-19 screening techniques that can assist public health authorities in curbing the spread of the infection. This chapter will review relevant study activities done on this topic to have a better understanding of the present research's strengths and flaws.

3.1 Healthcare System and COVID-19 Pandemic Overview

As the world becomes more digitized, artificial intelligence is finding widespread application in a variety of fields. AI is quickly being applied in fields such as health monitoring, m-health, enhanced decision-making processes, ailment classification, and diagnosis as a result of the digital revolution in healthcare. AI solutions for respiratory and other illnesses have enormous potential. As the outbreak nears in early 2020, the need for AI-based solutions becomes urgent. In [50], a thorough study is done about chest CT scan for COVID diagnosis. They are suggesting this analysis will help with early diagnosis and find out the severity of the infection. In [51], researchers have analyzed the effectiveness of Chinese Medicine (CM) for other respiratory-caused diseases. Also suggested this will help provide defense against external pathogens. Researchers are looking for a way of early diagnosis and preventive measure using an AI-based method in [52] [53]. In [54], they worked on contact tracing individual patients as well as explored anti-inflammatory and antiviral treatments for COVID as the symptoms show inflammatory signs. Mortality rate projections and epidemic patterns are analyzed and visualized in [55]. The impact of

vaccine and cure is a must at this crucial moment. Researchers and medical professionals are working to come up with a cure. Preventing COVID-19 from spreading is an equally important topic and in [56], authors have worked on preventive measures that can be adopted for preventing COVID-19. Due to geographical and temporal constraints, COVID testing is not always available. The scarcity and high cost of clinical testing are required to meet the large time-sensitive demand in underdeveloped nations such as Africa [8]. In-person visits to health care facilities expose more people to COVID-19. Given recent research indicating how extremely persistent and hence infectious COVID-19 is, this is not a minor issue. Current test turnaround times are several days and have lately increased up to 10 days in certain countries as labs become overburdened [57] [58]. The infection has already spread by the time a patient is diagnosed using present procedures. Current physical testing procedures expose medical personnel, particularly those with insufficient protection, to considerable infection risk. This may result in a severe scarcity of medical services and increasing stress on already overworked medical personnel. The United States Food and Drug Administration (FDA) authorized a rapid test that produces findings in 15 minutes to make testing more accessible [59]. The test functions similarly to the Polymerase Chain Reaction (PCR). The FDA recently authorized another quick molecular-based test that produces test outcomes in 13 minutes [60]. The FDA-approved raid tests have their own set of flaws. The FDA advises that this test has a significant risk of producing false negative results [61], and the patient still needs an in-person visit. Despite being substantially faster, the newly authorized test does not address many of the aforementioned issues. Furthermore, reports of shortages of crucial equipment required to collect patient specimens, including masks and swabs, may hamper its influence on

pandemic control [62] [63]. The FDA has also permitted at-home sample collection to prevent others from potential exposure [64]. Nevertheless, after collecting the nasal sample, the patients must place it in a saline solution and mail it overnight to a designated facility allowed to conduct specified tests on the kit. As a result, this strategy causes delays and may degrade sample quality if the sample is held for an extended period. Furthermore, because people collect the samples rather than qualified doctors or healthcare professionals, it may increase the possibility of inaccuracies during collection.

3.2 COVID-19 Diagnosis Systems and Biomarkers in AI

In paper [65], Wang Yunlu et al. suggested a technique for categorizing substantial screening of COVID-19 infected individuals. This research might be utilized to detect breathing patterns. In addition, a strong RS (Respiratory Simulation) Model has been introduced. This model is expected to close the knowledge gap between training and real-world data. Using bi-directional neural networks (NN) such as the GRU network tool, they found six clinically important breathing patterns (BI at GRU). Jiang Zheng et al. suggested a portable non-contact system in [66]. A thermal imaging camera and an Android device are used in this setup. To categorize respiratory health issues, a BI at GRU function is used for pulmonary illness findings. Brown Chloe et al. offer an ios/android app in [67], for collecting COVID-19 sound data from the crowdsourced sound. Among over 7k unique users, they discovered more than 200 positives for COVID-19. They classified the tasks into three categories and obtained commendable accuracy. In [7], Orlandic Lara et al. used the "COUGHVID" dataset to analyze cough symptoms in COVID-19 patients. Over 27,000 cough recordings (crowdsourced) encompass a wide variety of gender, ages, geographic regions, and COVID-19 status. The data status includes self-reported status factors, healthy

values, COVID sound recordings, and symptomatic sound recordings. It is now the most extensive expert-annotated public dataset. Mohamed Bader et al. presented a significant model for the extraction of samples from non-COVID and COVID utilizing SSP (Speech Signal Processing) and MFCCs [68]. These data reveal a striking connection between various RS and COVID cough sounds in MFCCs. They recorded the sounds of seven healthy individuals (3 female and 4 male) and 7 COVID infected patients (two female, five male). Mahmoud Al Ismail et al. [69] proposed a methodology based on vocal fold oscillation analysis as COVID-19 detection is difficult because the majority of patients have mild to severe respiratory symptoms. The collection comprised recordings of 512 people, however, they only used patients who reported within seven days following a medical check. Only 19 citizens satisfied this condition (ten females and nine males). The effectiveness of logistic regression was 91.20 percent. Laguarta Jord et al. [52] established an AI framework based on cough sound recordings to identify COVID symptoms, which predicted COVID-positive symptoms with 97.1 percent accuracy. A hundred percent accuracy in recognizing asymptomatic symptoms. With SP and Speech modelling approaches, Quatieri Thomas et al. [70] established a framework for identifying symptomatic conditions for COVID. This proposed technique provides a potential solution for COVID warning and monitoring. Jing Han et al. proposed a study of intelligent analysis of COVID-19 voice data using four features in paper [71].

Ref.	Methods	Dataset	Tasks	Advantage	Disadvantage
[72]	SVM	Small dataset collected from YouTube videos	Distinguish between COVID-19 positive and negative Speakers	Promising for sample voice conversations, to detect pandemic risk in a region.	Dataset limitation only 10 COVID positive speakers, 9 COVID negative speakers have been used.

Ref.	Methods	Dataset	Tasks	Advantage	Disadvantage
[71]	SVM	COVID-19 sound data set, Corona voice detect data (private datasets)	Diagnosis of sleep, fatigue and anxiety	Automatic fast and low-cost diagnosis of health status for COVID diagnosed patients.	Small and private dataset. The absence of distinction between healthy individuals and those with various respiratory illnesses.
[69]	LR, NL-SVM, DT, RF, AB	Dataset of COVID-19 positive and negative patients (private dataset).	Characterizing the vocal fold oscillation (VFO) pattern for COVID detection.	The VFO can be helpful to detect COVID-19 among patients.	Small private dataset (total of 19 patients), Effective likely to be on symptomatic patients .
[68]	MFCC	The author collected a dataset from 14 users (7 COVID patients and 7 Non-COVID Patients).	MFCC feature analysis to find correlation among COVID positive and healthy individuals from cough, speech and breathing sound.	Results show the possibility of cough and breathing sound's MFCC properties can be leveraged to develop a diagnostic system.	Due to a lack of data, there is a high correlation between speech data of COVID and non-COVID patients.
[65]	BI-ATGRU (Bi-directional and Attentional Gated Recurrent Unit)	Figshare dataset contains Shear force direction and vertical force direction.	Classifying respiratory patterns	Provides respiratory simulation model for generating training data.	Done by using self-generated data rather than real-world data.
[73]	MFCC + Spectrum + Feature Extraction	Data collected from clinical environments, crowdsourcing, and public media interview extraction	i. Negative cough ii. Positive cough	Generalized crowdsourced and clinical data from two different regions.	Inability to differentiate between COVID symptoms and other respiratory disease symptoms.
[66]	BI_at_GRU mechanism	Private dataset	Abnormal breathing pattern detection	Used a combination of thermal videos and breathing sound and can be used for COVID detection.	Requires thermal camera and specific mask. Small private dataset.
[74]	CNN, RestNEt50 LR, SVM, LDA, KNN, NB, DT, RF, XGB and ANN.	COUGHVID dataset	Dry and wet cough classification	Quick and cheap cough classification model	Dataset crisis and Deficiency of clinical data.

Ref.	Methods	Dataset	Tasks	Advantage	Disadvantage
[52]	CNN, RestNEt50	MIT Open Voice dataset (Private)	COVID-19 detection	Creates an AI speech processing framework. Successfully discriminates against asymptomatic patients.	Private dataset.
[75]	LSTM	Private dataset is collected from 14 users	COVID detection from Speech, Breathing and coughing sounds	Evaluates different acoustic features of cough, breathing and speech sound	Small private dataset. Not verified by a real-time variable large dataset.

Table 3.1: Comparative analysis of the related works.

Extraction of cough sounds from audio samples is critical for establishing useful cough-based biomarkers. Proposed systems must be clinically verified before they can be employed in real-time. Building a robust model is tough, especially when data is limited. The data collection and inclusion procedures take time. Another issue is coping with an uneven dataset. Finally, there is insufficient information to conduct an analysis. Sufficient clinically confirmed data sets are not accessible, especially not in the public domain. To test systems, real-time data must be used. As we can see from the summary, this field has a lot of room to grow. We are using expert-annotated crowdsourced self-reported data that has been thoroughly cleaned. Furthermore, our system addresses multi-class and imbalanced data problems, which are significant constraints in this field of study. We investigated several feature extraction algorithms in order to obtain the optimal collection of features for COVID-19 cough categorization that had not before been investigated.

Chapter 4

4 Proposed Models and Techniques

The suggested multi-class classifier-based models and approaches used to develop the COVID-19 Prescreening System architecture is discussed in this chapter. Section 4.1 gives detail of the overall architecture of the proposed multi-class classifier-based system, including resampling and cost-sensitive ML and DL model analysis. Section 4.2 goes through the dataset and approaches we utilized for data preparation and feature extraction for our models. Section 4.3 looks into the suggested methodologies for system architecture and data flow. Section 4.4 is about hyperparameter tuning of the implemented models. Section 4.5 elaborates the system's evaluation concepts and section 4.6 provides the development environment for our research project.

4.1 System Architecture

In this thesis, our system architecture for Cough analysis for COVID-19 prescreening will be discussed in detail with the techniques and libraries we have used. As we have already discussed, this is multiclass imbalanced disease data. We have employed two different data balancing techniques on Machine learning Algorithms and resampling techniques on Deep learning algorithms.

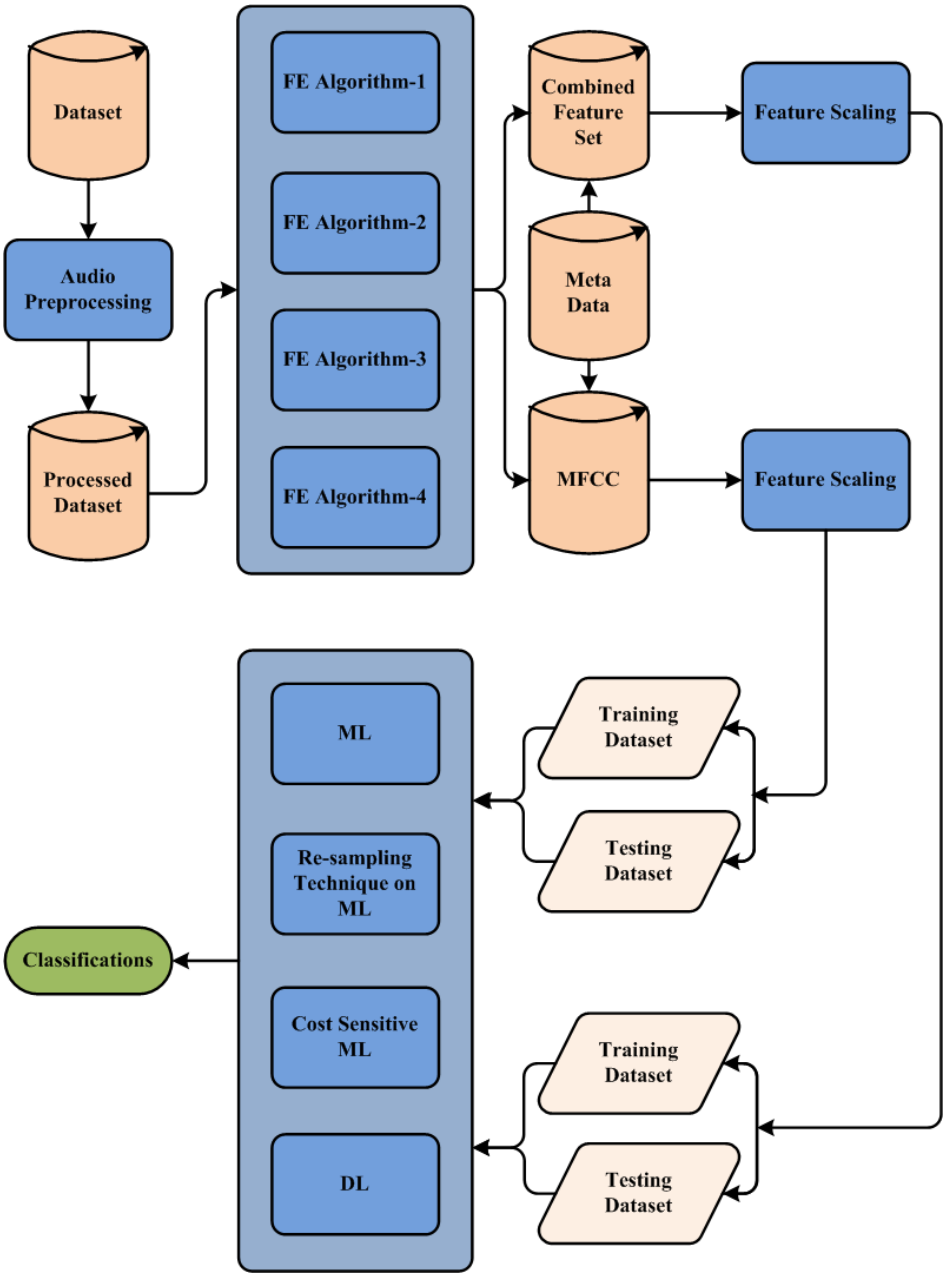


Figure 4.1: COVID-19 Digital Screening System Architecture.

Figure 4.1 depicts the overall design of our system. The crowdsourced dataset is made up of two sets of files, audio sound files and metadata in CSV format. The first step is to thoroughly clean the metadata set and prepare the audio dataset. Second, we extracted the audio signal features using four feature extraction methods, resulting in a combined audio

feature set that included all of the features retrieved by the four FS techniques, as well as an MFCC feature set that solely included MFCC features. The feature sets are then updated by adding metadata with the merged feature set and solely the MFCC feature set. Following that, we will scale the features. The data sets are then fed into machine learning and deep learning algorithms. On ML algorithms, we used the SMOTE-Tomek link resampling approach and Cost-sensitive analysis. We tested COVID classification with and without the resampling approach for DL algorithms. The section will now go over the whole system's architecture step by step. The ML and DL algorithms, as well as the data balance strategies employed, will be discussed first in this procedure. This section will now delve deep into the architecture step by step by addressing data set preparation, preprocessing and feature engineering.

4.1.1 ML Algorithm Using Resampling Technique

The analysis of audio data is a challenging task within itself. Furthermore, our dataset is skewed and multi-classed, which adds to the complexity. SMOTE + Tomek Links [76] were used to cope with the imbalanced multiclass data. This is a hybridization approach that combines both under-sampling and oversampling techniques. This is done to improve the performance of classifier models for samples generated using these strategies. SMOTE-Tomek link is a hybrid approach that tries to clear overlapping data points for each of the classes spread in the sample space. This method was used to train 14 ML algorithms and resulted in considerable improvements across all assessment measures. We used Naive Bayes, Logistic Regression, SVM, DT, K-NN Extra Trees, Random Forest, Gradient Boost, AdaBoost, Light Gradient Boosting Machine, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Dummy Classifier, and Ridge Classifier as models. We

utilized the StratifiedKFold strategy for cross-validation, which is particularly successful for unbalanced data since it maintains the class ratio in each fold with a split set count of 10. The hyperparameters were optimized for the highest performing model Extra Trees method. We used the Imblearn module for the resampling approach and Pycaret for the model implementation and assessment.

4.1.2 ML Algorithm Using Cost-Sensitive Analysis Technique

When training a machine learning model, cost-sensitive learning considers the costs of prediction mistakes (as well as potentially other expenses). It is a branch of research that is closely connected to unbalanced learning and focuses on the classifying of datasets with skewed class distributions. As a result, many of the concepts and approaches developed and utilized for cost-sensitive learning may be used in imbalanced classification situations. Machine learning algorithms that are cost-sensitive, employ the cost matrix directly. These cost-sensitive algorithms employ a class weight attribute based on a specific class weight. Model is penalized for misclassification. We used cost-sensitive multiclass ML algorithms on our imbalanced multiclass COVID dataset. We used the Scikit learn library and the cost-sensitive Extra trees, SVM, Random Forest, and Decision tree algorithms. The stratified fold approach was employed for cross-validation with hyperparameter tuning for improved results. The Extra Trees technique outperformed the other three constructed models.

4.1.3 Deep Learning Algorithms for Multi-Class Imbalanced Data

We have implemented Sequential Model and RNN- LSTM for our multi-classed COVID cough sound data. Sequential DL model has 4 layers. The first three layers are comprised of having activation function ReLU and a dropout function. The final layer has a softmax

activation function. RNN-LSTM algorithm has been deployed for its capability of storing previous inputs and taking a decision based on previous input. Our RNN-LSTM architecture comprises 2 LSTM layers where the first layer is a sequence layer and the second layer is a sequence to a vector layer, a dense layer, and a dropout layer. The final layer is the softmax layer consisting of 3 neurons. We have used the resampling technique for dealing with an imbalanced dataset. For optimization, adam optimizer, and for loss function, sparse categorical cross-entropy has been used for both Sequential model and RNN-LSTM.

Dropout Layer:

Dropout Layer is a prominent regularization approach for reducing overfitting in deep learning models. In models, overfitting is a case that happens due to the shown accuracy of training data rather than testing data. Overfitting occurs when a model uses more of the noise information which increases its performance for known train data and decreases its performance in the case of novel test data. The dropout technique from some of the neurons in hidden or visible layers is dropped or omitted randomly. Doing this dropout technique regularizes the deep learning models and makes the models more robust and avoids overfitting. The dropout layer can be applied to the input layer as well as any or all of the concealed layers, but the dropout layer cannot be applied to the output layer. We have used 0.5 dropout function settings for the sequential model and 0.3 for RNN-LSTM.

Softmax Activation Function:

Softmax is an activation function that transforms a vector of numbers into a vector of relative probabilities. A DL model requires an activation function in the model's output

layer. The softmax function is widely used in multiclass classification, and it is used to generate the output layer of models for multiclass classification problems. This function computes the probability of each class. SoftMax function produces an array with the greatest probability values, which is regarded as the most exact probabilistic label for the sample. For both DL models, we employed the softmax activation function on the output layer.

Rectified Linear Activation Function (ReLU):

The Rectified Linear Activation function, abbreviated ReLU, is a piecewise linear function. If the input is positive, this function returns the same value as the input; otherwise, it returns zero. Because of its ease of implementation, it has become a common activation function for many different types of DL models and neural networks. We have used ReLU as it is easier to train and tune, resulting in improved performance. Advantages include computational simplicity and the ability to generate a real zero-result; optimization is simple because it appears and performs like a linear activation function.

4.2 Dataset Preparation and Feature Engineering

This section will briefly talk about the dataset's description, preprocessing and cleaning techniques, and feature engineering steps.

4.2.1 Dataset Description

We utilized the COUGHVID dataset [7], which is a massive crowdsourced, publicly accessible collection of cough recordings throughout the world. It is the largest publicly available COVID-19 related cough sound collection known to exist. This collection contains 27,000 recordings from all across the world, with 1,155 data instances claiming

to have COVID-19. Crowd-sourced data is supported by Deep Health and the Swiss National Science Foundation (SNSF). Four highly qualified physicians who are experts in this field, analyzed a portion of the information to determine which crowdsourced samples are likely to be from COVID-19 patients, adding an extra degree of confirmation. The COUGHVID collection has over 2,800 recordings which are expert-labeled coughs. Each recording provided a diagnosis, severity level, and indication of the presence of audible health concerns such as dyspnea, wheezing, and nasal congestion present. This dataset, which includes expert classifications as well as participant metadata, can be used to train algorithms that recognize a variety of information from participants that were chosen based on their coughing sounds. This dataset comprises samples from a diverse range of individuals' ages, genders, and geographic locations. The data set also includes individuals' COVID-19 statuses and information on pre-existing respiratory diseases, which may help ML models to generalize well. In a nutshell, this dataset includes the following:

1. This dataset is one of the largest expert physician-labeled cough datasets available for use in a variety of cough audio classification tasks. Four expert professional physicians worked on annotating, each of whom revised 1000 records.
2. Quality, cough type, congestion, diagnosis, severity, and other variables are supplied by experienced annotators.
3. This dataset provides diagnostic categories such as upper infection, lower infection, obstructive disease, COVID-19, and healthy cough.
4. The dataset included 27000 recording files in WEBM and OGG format, as well as metadata spreadsheet files comprising various aspects of the recording, including expert remarks.

4.2.2 Versatile Demographics of The Data

In the data, the male-to-female representation ratio is 65.5:33.8. The average age of the participants and standard deviation was 34.4 years and 12.8 years, respectively. Approximately 77% of the cough recordings were declared healthy, whereas approximately 15.5% were declared symptomatic. The most significant sample, which is COVID positive, accounts for 7.5% of the whole dataset. The demographic variables in the COUGHVID data are seen in Figure 4.2.

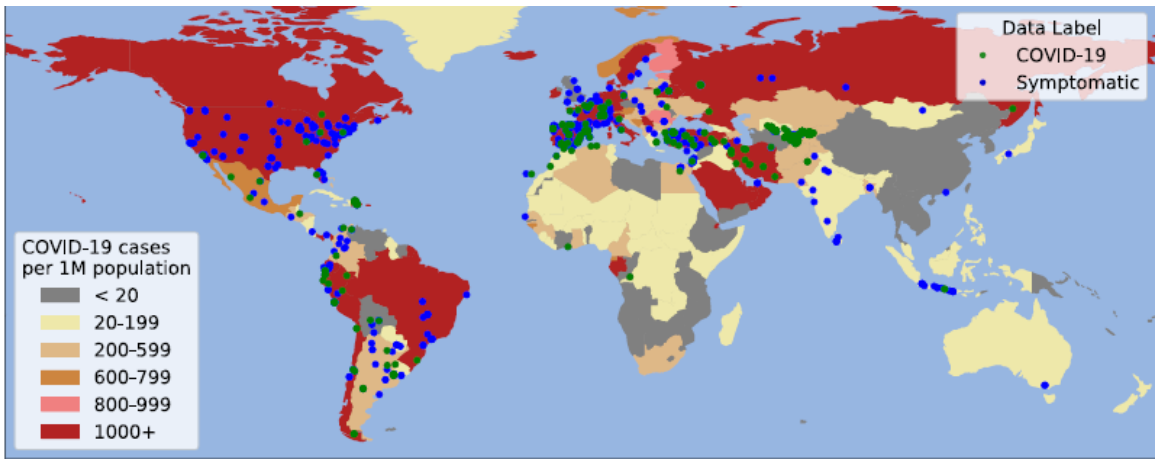


Figure 4.2: Cumulative COVID-19 cases in April and May 2020 per 1 million population, along with the GPS coordinates of the received recordings [7].

4.2.3 Data Cleaning

The data cleaning process in COUGHVID is separated into two phases. Cleaning of information as a CSV file, followed by cleaning of audio data. Data preparation and preprocessing are key processes in deciding the performance of the cough sound analysis as the first stage of developing the prescreening tool. This is because we are currently creating the dataset that will be utilized to train the cough analysis model. As a result, much

effort was expended in comprehending and preparing the data. This stage may be separated into two sections:

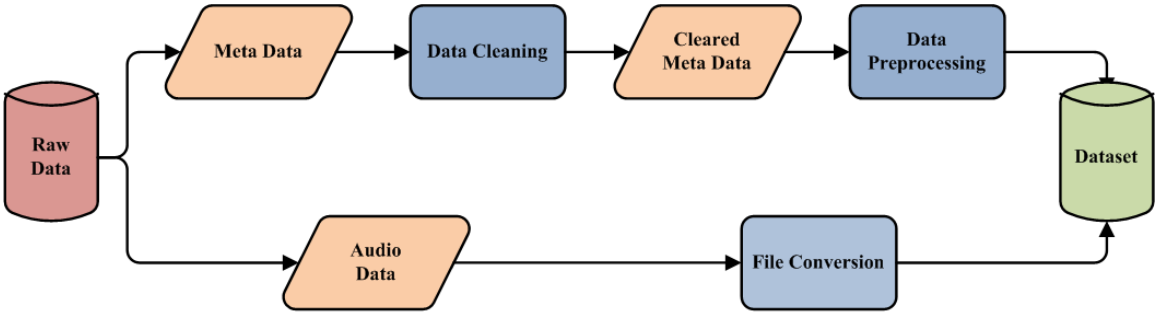


Figure 4.3: Dataset Preparation.

4.2.3.1 Metadata Selection and Cleaning

Metadata selection and cleaning are very important steps. From the 27000 collected data, we have used the subset of COUGHVID crowdsource data that was annotated by 4 experts.

Data Selection:

The initial selection of recordings that are diagnosed by experts is selected based on several criteria. We created the dataset by considering the below factors:

- Recording having a cough detection probability of 0.8 (through using random sampling and pre-filtering with a cough classifier)
- self-reported status variable by the users.
- 100% of the recordings which are self-labelled by users as COVID-19, having a value above 0.8 of feature “cough_detected” and must be labeled by at least one expert.
- Plus, we have selected recordings that were labeled by all 3 experts. It’s about 15% of the whole set.

While selecting the data we only chose the data that are at least diagnosed by one expert and cough detected. Finally, a subset of 2804 data was created.

Meta Data Cleaning:

We dealt with quite a few scenarios while metadata cleaning, such as 1) eliminating the conflicted diagnosis by experts and 2) creating a new feature by merging the diagnosis feature findings based on the majority. The chosen dataset consists of 2804 recordings with various symptoms labeled for the "diagnostic" attribute provided by the four experts. The experts' diagnostic categories include upper infection, lower infection, obstructive sickness, COVID-19, and healthy cough. Any experts' missing values for diagnostic characteristics were replaced with "not annotated." We only preserved the "Coivid-19" and "healthy cough" identified annotations for the expert's diagnosis. We categorized any alternative diagnosis for any of the aforementioned symptoms as "others." The next stage was to incorporate all four experts' diagnoses and establish a "Label." The label is divided into three groups namely COVID-19, healthy cough, and others. We discarded contradictory data and proceeded with the majority. A total of 2748 records are finally in our subset, where the number of “COVID-19” records is 542 (20%), the “healthy cough” record is 617 (22.5 %), and the “others” record is 1589 (57%).

4.2.4 Audio Data File Preparation

We created our cough recording folder by converting, selecting, and cleaning the given audio recordings. COUGHVID audio data recordings were provided in the form of WEBM and OGG files. We used the FFMPEG tool in python to convert the 27000 recordings into WAV files. This tool enables converting audio or video data formats. After that, we created

a separate folder with the 2748 selected recording WAV files from the whole dataset based on the prepared metadata CSV file and created Dataset.

4.2.5 Data Preprocessing

Data preprocessing is a very important step for classification problems. It can make or break the whole classification process. Data preprocessing is done on metadata.csv file and audio data recordings.

4.2.5.1 Metadata Preprocessing

Many algorithms cannot operate with categorical values. Thus, it is necessary to transform them into acceptable data in our instance, numerical values. Because the "Label" column in our metadata CSV file included a categorical value, we used the Label encoding approach using the Label Encoder technique. In Python, the scikit package is used for label encoding. LabelEncoder is an effective tool for transforming category variables into numerical values. LabelEncoder encodes categorical classes with n-1 values. In this situation, n denotes the number of distinct classes. When a label is repeated, the same value is assigned as before.

4.2.5.2 Audio Data Preprocessing

We have normalized the audio signals and passed them through a low pass filter. Then down-sampled the cough samples as well. We have selected only the cough portion of the audio discarding all kinds of noises and silence. Segmented all the cough samples, removed the short segments also resized all the audios to the same size. Lastly rescaled the audio data into a scale of [-1,1] and prepared all the audio files for feature engineering.

4.2.6 Feature Engineering

An audio signal is often expressed in terms of amplitude and time. When the models are unable to understand the audio data that is provided directly, feature extraction is used. Feature extraction is used to convert audio data into an understandable format. Spectrogram, Short-Time Fourier Transform, Mel-Frequency Cepstral Coefficients (MFCC), and Chromagram were the four feature extraction techniques employed. We produced two sets of data in CSV file format after extracting the features. These two datasets are one CSV file that contains all of the features taken from all four approaches and another CSV file that just contains MFCC features. The basic purpose of feature engineering is to get the best performance and accuracy possible when using data to conclude with a model. Datasets after Feature Extraction technique:

- Dataset 1: Created combining all the features extracted from all four feature extraction techniques.
- Dataset 2: Using features extracted using MFCC

4.2.7 Feature Scaling

Feature scaling is a data preparation approach in which we transform all input values to a standard scale before using them in the learning model. If the data is not scaled, characteristics with a wide range of values will have a greater effect on the learning model's output. As a result, other variables that are important but have a narrow range become less useful to the overall conclusions given by the predictive model. To equalize all characteristics, it is necessary to scale the data, which also aids the algorithm in reaching

convergence faster and optimizing the result. We have used the standard scaler technique to scale the data.

Our final data sets consist of three classes which makes it a multiclass classification problem and the dataset is imbalanced so we used the data balancing technique for our imbalanced data.

4.3 Dealing with Imbalanced Dataset

The imbalanced dataset is one in which the target class's observations are unequal. In other words, the allocation of observations to distinct categories or classes is uneven, and the disparity is significant. Imbalanced data often refers to a classification challenge in which the classes are not evenly represented. A class imbalance problem can occur in both two-class classification problems and multi-class classification problems. The majority of approaches may be used for either.

We have explored two techniques that can be used for balancing imbalanced data.

1. Resampling the data set
2. Cost-sensitive algorithms

4.3.1 Resampling The Dataset

Resampling is a frequently used approach for coping with extremely imbalanced datasets. It entails deleting samples from the majority class (under-sampling) and/or including additional instances from the minority class (over-sampling).

4.3.1.1 Under-Sampling

In a classification dataset with a skewed class distribution, the class distribution must be balanced. Under-sampling refers to a set of procedures used to balance the class distribution. The most fundamental under-sampling approach involves randomly selecting samples from the majority class and eliminating them from the training dataset [77].

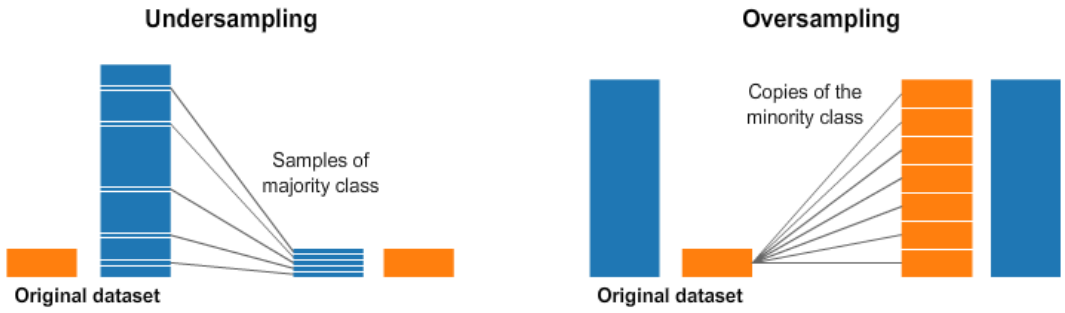


Figure 4.4: Visual representation of under sampling and over sampling technique [77].

4.3.1.2 Over-Sampling

Oversampling is the process of replicating or synthesizing new samples from the class distribution's minority class. Oversampling is used to ensure that the number of observations in the minority class matches the number of observations in the majority class and that the dataset is balanced.

4.3.2 Data Balancing Techniques

1) Resampling Technique:

In a real-world application, classification modeling is frequently confronted with an imbalanced dataset problem, in which the majority class has a substantially larger number of members than the minority class. The SMOTE-Tomek link [76] technique combines

SMOTE's ability to generate synthetic data for the minority class with Tomek Links' ability to erase data from the majority class, both of which are identified as SMOTE-Tomek links. Unlike the original oversampling strategy, this method is effective because the synthetic data produced are relatively close to the feature space on the minority class, hence adding more "information" to the data.

2) Cost-Sensitive Analysis:

Machine learning algorithms that are cost-sensitive employ the cost matrix directly. Machine learning algorithms are rarely designed with cost-sensitive learning in mind. The class weight option on the specialized cost-sensitive classifiers in the Scikit-learn Python machine learning toolkit gives instances of these cost-sensitive enhancements. Each sample in the training dataset has its penalty term. For the minority class, a greater weighting is used, for enabling the margin to be softer, but a lower weighting is used for the majority class, requiring the margin to be firmer and avoiding misclassified cases.

4.4 Hyperparameter Tuning

Hyperparameters are parameters whose values govern the learning process of models. Hyperparameters also dictate the values of model parameters learned by a learning algorithm. We utilized a grid search to fine-tune the hyperparameters for each classical model. The parameters iterated for all techniques up to the maximum number of iterations. We also explored the number of neighbors for the KNN algorithm, kernel and gamma for the SVM Classifier, and maximum depth for the Decision Tree and Random Forest, number of estimators for the Random Forest and Extra Tree classifiers. Some algorithms gave better results with default parameters. The grid search strategy is also used, similar to

the classical models, to discover the ideal hyperparameter values for various DL algorithm settings. We ran several sample tests and discovered that networks with a dropout rate of 0.3 and 0.5 produce the best results. We ran a 5-fold cross-validation on each grid search arrangement. On a separate validation dataset, we trained each model until there was no further improvement in performance.

4.5 Evaluation Criteria

In this part, we will go over the criteria for evaluating our proposed COVID-19 pre-screening Machine Learning and Deep Learning models.

4.5.1 Accuracy of Classification

Classification accuracy is an essential and widely used model evaluation criterion. Predicting a class label's records provided in a problem is what called modeling classification is all about. To begin, a classification model is used to produce a prediction for each case in a test dataset. These predictions are then compared to the known labels for the test set examples. Accuracy is calculated as the ratio of the total number of accurately predicted cases to the total number of predictions made by the model on the test dataset.

$$Accuracy = \frac{Total\ Number\ of\ Correct\ Prediction\ done\ by\ Model}{Total\ Number\ predictions\ of\ the\ Model} \times 100 \quad \text{.....(4.1) [78]}$$

Accuracy, on the other hand, does not always match the model's actual performance. When the misclassification rate for minor classes is high, accuracy suffers. It also disregards the dataset's class imbalance problem. This imbalance occurs when the number of examples in the majority and minority groups vary significantly. As a result, different performance evaluation matrix must be assessed to obtain actual model performance.

4.5.2 Confusion Matrix

A Confusion matrix is another important assessment criterion, particularly for an imbalanced dataset. The performance of a classification model with n target classes is evaluated using this approach. The confusion matrix compares the actual class values to the machine learning model's predictions. This provides us with a clear view of how well the classification model is doing and the sorts of errors it produces.

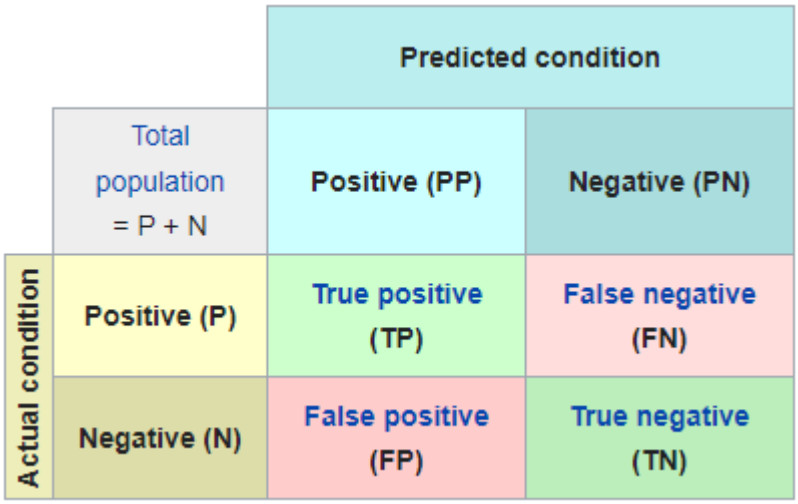


Figure 4.5: A typical Confusion Matrix for binary classification [79].

The output of the classification matrix can be labeled as True Positive, True Negative, False Positive, and False Negative, as briefly stated below:

- **True Positive (TP):** The value of the model shows the number of predicted values that match the actual value. The actual value was positive and the model also predicted a positive value. For our system suppose the model predicted that the data is COVID-19 and the actual label is also COVID-19.

- **True Negative (TN):** The value of the model shows the number of predicted negative values. These values should that match with the actual value which was negative as well. For our system suppose the model predicted that the data is predicted as not COVID positive and the actual label is also not COVID positive.
- **False Positive (FP):** The value of the model shows the number of predicted positive values that do not match with the actual value which is negative. For our system, False Positive will be the model predicting that the instance of data is COVID positive but the actual label is not COVID positive.
- **False Negative (FN):** The value of the model shows the number of predicted negative value that does not match with the actual positive value. For our system, False Negative will be the model predicting that the data is not COVID positive but the actual label is COVID positive. This is also known as type 2 error.

4.5.3 Precision

Another assessment measure is precision, which provides an overview of how many of the accurately anticipated events turned out to be positive. Precision is a helpful statistic when an FP is more concerning than an FN. Precision may be calculated using the following equation:

$$Precision = \frac{TP}{TP + FP} \quad \text{.....(4.2) [78]}$$

4.5.4 Recall

Recall provides an overview of how many real positive situations we were able to anticipate accurately using our model. In circumstances when False Negative outweighs False Positive, recall is a relevant statistic. This strategy is useful when dealing with

medical data, such as ours, where it doesn't matter if we raise a false alert, but true positive instances should not be ignored. The equation for calculating recall is:

$$Recall = \frac{TP}{TP + FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.3)$$ [78]

4.5.5 F1 Score

The F1 Score is calculated by taking the weighted average of Precision and Recall values. This score considers both FP and FN. For an imbalanced dataset, the F1 score is much more informative. When the costs of FP and FN are comparable, accuracy is more beneficial. Precision and Recall should be explored if the cost of FP and FN vary considerably.

$$\text{F1 Score} = 2 \times \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \ldots\ldots\ldots\ldots\ldots\ldots (4.4)$$ [78]

4.5.6 Receiver Operating Characteristic Curve (ROC Curve)

The ROC curve (Receiver Operating Characteristic curve) is a graph that displays the performance of a classification model's overall categorization levels. The True Positive Rate (TPR) and the False Positive Rate (FPR) are shown on this curve.

1. True Positive Rate is a model's sensitivity or Recall parameter that indicates the proportion of positive values that were correctly detected as positive. This may be stated as follows:

$$TPR = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.5)$$ [80]

2. The False Positive Rate, often known as a model's specificity, determines the fraction of values that are negative and were likewise recognized as negative by the model. This may be stated as follows:

$$FPR = \frac{False\ Positive}{True\ Negative + False\ Positive} \ldots\ldots\ldots\ldots\ldots\ldots (4.6)\ [80]$$

The Roc curve is designed for binary classification. As we are dealing with the multiclass task, we are going to use a bit modified version using OvO or also known as the One vs One technique. This technique shows the roc curve for one class vs another. This means we have three classes class1, class2, and class3. It will display the roc curve for class1 vs class2, then class1 vs class3, and so on.

4.5.7 Area Under Curve (AUC Score)

The term "Area Under the ROC Curve" is an acronym for AUC Curve. In other words, AUC evaluates the entire two-dimensional region beneath the entire ROC curve. The AUC ROC Curve (Area Under ROC Curve), also known as the ROC AUC Score, is a matrix used to compare different ROC Curves.

4.6 Development Environment

We utilized a Windows 10 PC Processor Intel(R) Core i7-3537U CPU @ 2.00GHz, 2501 Mhz, 2 Core(s), 4 Logical Processor(s), and 8.00 GB of RAM. We used the Google Colab platform to create our tool and models in Python 3.7 language. CoughVid [77] datasets were kept on Google Drive, separated into distinct folders. The NVIDIA-SMI 495.46 GPU contains 320 Tensor Cores with a clock speed of 585-1590, making model building easier and faster. We used the Panda for data analysis, and numerical analysis was done by using

Numpy, Matplotlib, and Seaborn libraries are used to create graphs for the experimental findings. SciPy and Librosa libraries were utilized for feature engineering and preprocessing. Tensorflow, Keras, and Pycaret were used to implement the models.

Chapter 5

5 Results and Discussion

This chapter will provide our findings and analysis from the tests carried out to develop the COVID-19 prescreening framework. Section 5.1 will present the findings and tests performed on fourteen different ML algorithms with and without using the resampling approach, and Section 5.2 will present the results of the cost-sensitive analysis technique on chosen ML algorithms. Section 5.3 will provide the deep learning model outcomes, and Section 5.4 will provide a comparative result analysis of all the approaches we used in our research. We will also demonstrate how different data balancing techniques affect model performance.

5.1 Findings on ML Algorithms

As discussed earlier in chapter 4, our dataset COUGHVID is multiclass containing three classes and imbalanced. We developed two feature sets after extensive cleaning and preprocessing. One is a combined feature set, which contains features from all four feature extraction approaches used in this study, and the other is an MFCC feature set, which only contains MFCC features. We applied fourteen ML algorithms on both the MFCC feature set and the Combined feature set without applying any data balancing technique.

5.1.1 ML Algorithm without Balancing Technique

As shown in Figure 5.1, the best performing algorithm among these fourteen algorithms is RF, however, accuracy is quite low (53.66%), and other assessment matrix are also low. If

we simply evaluate the AUC score, the ET algorithm performs better, but the rest of the assessment scores are lower than RF.

Model	Accuracy	AUC	Recall	Prec.	F1
Random Forest Classifier	0.5366	0.7399	0.5361	0.5364	0.5357
Light Gradient Boosting Machine	0.5358	0.7348	0.5352	0.5354	0.5350
Extra Trees Classifier	0.5353	0.7432	0.5348	0.5373	0.5356
Quadratic Discriminant Analysis	0.5183	0.7043	0.5182	0.5220	0.5186
K Neighbors Classifier	0.4995	0.6871	0.4987	0.5030	0.4909
Gradient Boosting Classifier	0.4858	0.6717	0.4850	0.4845	0.4837
Decision Tree Classifier	0.4701	0.6036	0.4694	0.4693	0.4689
Linear Discriminant Analysis	0.4342	0.6051	0.4336	0.4331	0.4327
Ridge Classifier	0.4340	0.0000	0.4331	0.4320	0.4312
Logistic Regression	0.4324	0.6064	0.4317	0.4311	0.4308
Ada Boost Classifier	0.4046	0.5936	0.4038	0.4014	0.4023
SVM - Linear Kernel	0.3927	0.0000	0.3924	0.3917	0.3915
Naive Bayes	0.3747	0.5564	0.3723	0.3708	0.3471
Dummy Classifier	0.3391	0.5000	0.3333	0.1150	0.1717

Figure 5.1: Multiclass COVID-19 Cough Sound Classification without balancing technique on combined feature set.

For designing a classification model more robust and accurate, two data balancing technique has been explored for ML algorithms.

5.1.2 ML Algorithm with Balancing Technique

In the process of this thesis, we have used the resampling technique for dealing with the COVID-19 sound analysis problem as it is an imbalanced dataset and multiclass classification problem. The SMOTE-Tomek link resampling technique has been used for

balancing the dataset. We implemented fourteen ML algorithms with and without the resampling technique to see the improvement. These algorithms are Extra Trees (ET), RF, LGBM, DT, KNN, GBC, QDA, AdaBoost, LR, Ridge Classifier, and Dummy classifier. Using the SMOTE-Tomek link has shown a significant improvement in accuracy of about 16.89% and the AUC score improved to 4.14%. We have implemented the SMOTE-Tomek link on the combined features set and MFCC features set.

The SMOTE-Tomek link which is a hybrid data balancing technique is used on the features extracted from MFCC. In Figure 5.2, all fourteen implemented model is shown with respective Accuracy, AUC score, Recall, Precision and F1 score. Among all the models, ET algorithm performs best in terms of AUC score which is 0.79. The second-best model is Random Forest (RF) but its accuracy differs from ET by 0.65% and the AUC score is also lower by 0.35%. The third best model is LGBM providing a COVID-19 analysis AUC score of 72.53% and an accuracy of 62.88% which is respectively 6.47% and 6.83% lower than the ET model.

Model	Accuracy	AUC	Recall	Prec.	F1
Extra Trees Classifier	0.6971	0.7900	0.5594	0.6886	0.6738
Random Forest Classifier	0.6906	0.7865	0.5694	0.6779	0.6735
Light Gradient Boosting Machine	0.6288	0.7253	0.5550	0.6298	0.6286
Decision Tree Classifier	0.5660	0.6428	0.5217	0.5993	0.5778
K Neighbors Classifier	0.5098	0.7827	0.6103	0.6856	0.5108
Gradient Boosting Classifier	0.4944	0.6244	0.4628	0.5425	0.5099
Quadratic Discriminant Analysis	0.4463	0.5849	0.4206	0.5148	0.4667
Ada Boost Classifier	0.4020	0.5533	0.3893	0.4865	0.4247
SVM - Linear Kernel	0.3914	0.0000	0.3840	0.4894	0.4144
Logistic Regression	0.3893	0.5633	0.3958	0.4934	0.4120
Ridge Classifier	0.3886	0.0000	0.3951	0.4934	0.4112
Linear Discriminant Analysis	0.3873	0.5635	0.3910	0.4929	0.4108
Naive Bayes	0.3792	0.5321	0.3625	0.4708	0.4044
Dummy Classifier	0.1708	0.5000	0.3333	0.0292	0.0499

Figure 5.2: The evaluation score of ML algorithms using the SMOTE-Tomek link method using features extracted by MFCC.

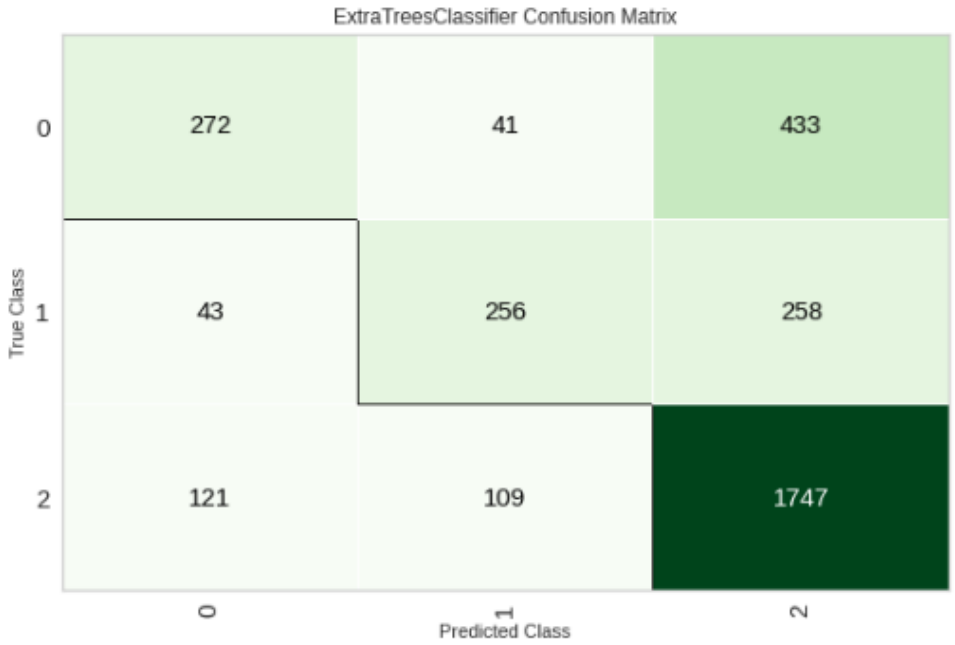


Figure 5.3: The Confusion Matrix for Extra Tree Classifier with SMOTE-Tomek link and MFCC feature set.

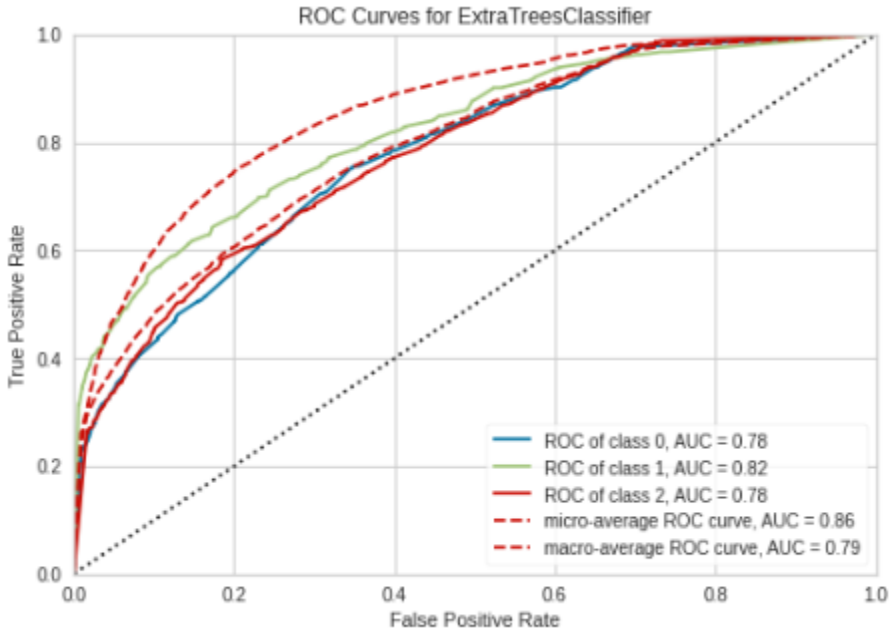


Figure 5.4: ROC Curves for Extra tree classifier where ROC score is 0.79.

ET algorithm outperforms other models and provides better Recall (0.5594), Precision (0.6886) and F1 score (0.6738) (figure 5.2). ROC score for ET is 0.79 and the ROC score for COVID-19 (class 0) against healthy cough (class 1) and others (class 2) is 0.78 (figure 5.4). Figure 5.3 shows Class 0 correctly classified 272 instances.

	Model	Accuracy	AUC	Recall	Prec.	F1
0	Extra Trees Classifier	0.6936	0.7878	0.5693	0.6809	0.672

Figure 5.5: Results from test set for ET on MFCC features.

As shown in Figure 5.5, ET performs well on the train set providing an AUC score of 0.7878. Other evaluation scores are also close to trainset results. ET achieved Recall of 0.5693, Precision 0.6809 and F1 score of 0.672.

We have also implemented the fourteen models using the combined features set. In the case of using combined features, the Extra tree algorithm gives the best AUC score of 0.7877 and an accuracy of 71.23% among all fourteen classification models. The Recall, Precision and F1 score are also higher than other implemented models. The Dummy classifier works as a baseline and the AUC score from the Dummy classifier to ET algorithm has increased by 28.77%.

Model	Accuracy	AUC	Recall	Prec.	F1
Extra Trees Classifier	0.7123	0.7877	0.5666	0.7110	0.6855
Random Forest Classifier	0.6845	0.7753	0.5663	0.6705	0.6683
Light Gradient Boosting Machine	0.6524	0.7360	0.5570	0.6409	0.6442
Quadratic Discriminant Analysis	0.6179	0.6442	0.4672	0.5876	0.5883
Decision Tree Classifier	0.5545	0.6392	0.5242	0.5975	0.5682
Gradient Boosting Classifier	0.5267	0.6434	0.4808	0.5577	0.5380
K Neighbors Classifier	0.4595	0.7623	0.5828	0.6730	0.4487
Ada Boost Classifier	0.4182	0.5631	0.3945	0.4918	0.4400
Logistic Regression	0.4107	0.5852	0.4258	0.5120	0.4311
Linear Discriminant Analysis	0.4087	0.5853	0.4232	0.5093	0.4289
Ridge Classifier	0.3988	0.0000	0.4183	0.5077	0.4194
SVM - Linear Kernel	0.3949	0.0000	0.4165	0.5092	0.4153
Naive Bayes	0.3535	0.5348	0.3712	0.4747	0.3721
Dummy Classifier	0.1743	0.5000	0.3333	0.0304	0.0518

Figure 5.6: The evaluation score of ML algorithms using the SMOTE-Tomek link method using the Combined Feature set.

In Figure 5.6, all fourteen implemented model is shown with respective Accuracy AUC score, Recall Precision and F1 score. Among all the models, ET algorithm performs best in terms of AUC score which is 0.787. The second-best model is Random Forest (RF) but

its accuracy differs from ET by 2.78% and the AUC score is also lower by 1.24%. The third best model is LGBM providing a COVID-19 analysis AUC score of 0.7360 and an accuracy of 65.24% which is respectively 5.17% and 5.99% lower than the ET model.

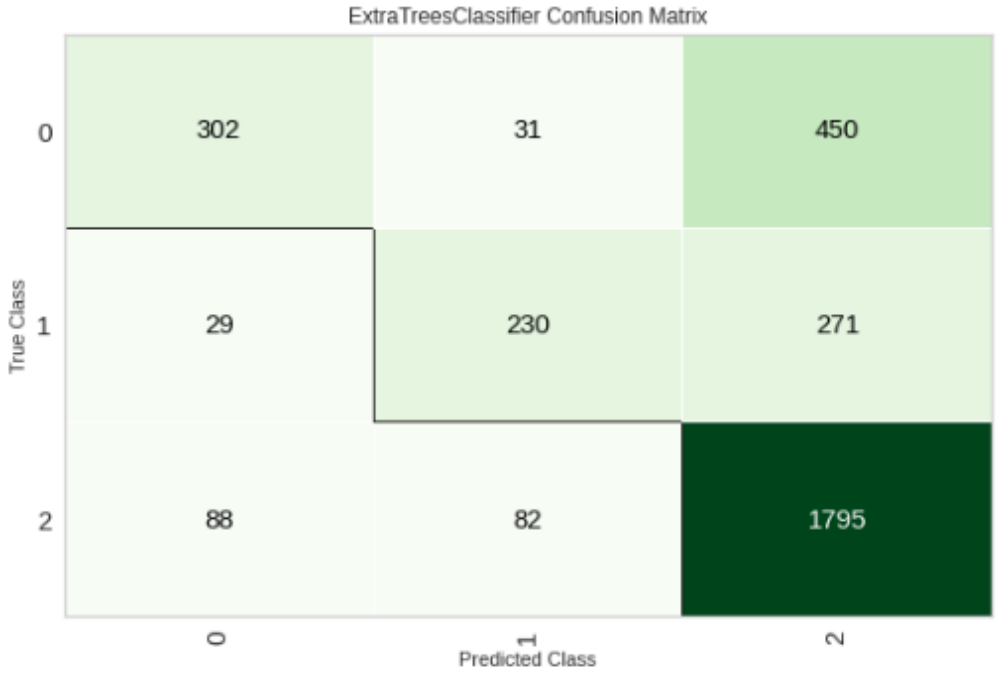


Figure 5.7: The Confusion Matrix for Extra Tree Classifier using SMOTE-Tomek link and combined feature set.

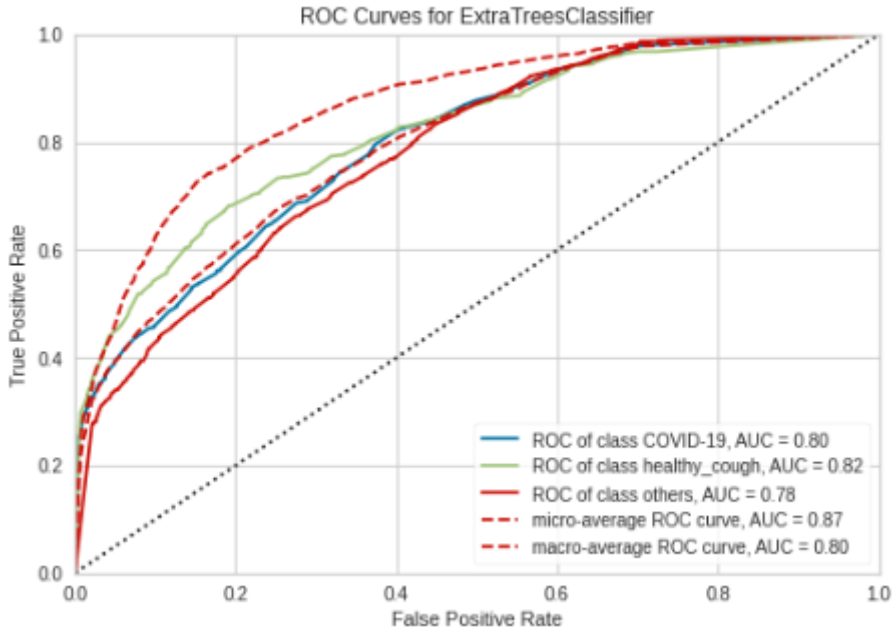


Figure 5.8: ROC Curves for Extra Tree classifier where ROC score is 0.80.

ET algorithm outperforms other models and provides better Recall (0.5666), Precision (0.7110) and F1 score (0.6855) (figure 5.6). ROC score for ET is 0.80 and the ROC score for COVID-19 (class 0) against healthy cough (class 1) and others (class 2) is 0.80 (Figure 5.8).

	Model	Accuracy	AUC	Recall	Prec.	F1
0	Extra Trees Classifier	0.709	0.793	0.5709	0.7028	0.6846

Figure 5.9: Results from test set for ET on the combined feature set.

As shown in figure 5.9, ET performs well on the test set providing an AUC score of 0.793. Other evaluation scores are also close to trainset results. ET achieved Recall of 0.5709, Precision 0.7028 and F1 0.6846.

5.2 Cost-Sensitive ML Classification

Cost-sensitive analysis was used for the COVID-19 cough analysis model. There are plenty of ways to work with imbalanced data, and in our case, we used costs as a penalty for misclassification when the algorithms are trained. Cost-sensitive machine learning algorithms DT, SVM, RF and ET were implemented on COVID crowdsourced data.

ML Algorithm	Accuracy	Precision	Recall	F1 Score	Roc AUC
Extra Trees	0.6920	0.6409	0.5440	0.57	0.7500
Random Forest	0.6910	0.6416	0.5430	0.57	0.7346
SVM	0.6648	0.7084	0.4750	0.47	0.7390
Decision Tree	0.6081	0.5268	0.5405	0.53	0.6540

Table 5.1: The scores of the applied algorithms on MFCC feature set.

Table 5.1 Showing the Accuracy, Precession, Recall, F1 and ROC AUC score of ML algorithms using the cost-sensitive method on MFCC feature set. In our applied classification models, the Extra Trees (ET) algorithm showed the highest accuracy 69.2% and the ROC AUC score of 0.75. Whereas RF showed a ROC AUC score of 0.734, SVM's AUC score is 0.739 and the lowest score is DT's for 0.654.

Figure 5.10 shows the confusion matrix for the classes and Figure 5.11 shows the ROC curve which displays the specificity and sensitivity among the three classes (COVID-19, healthy cough and others)

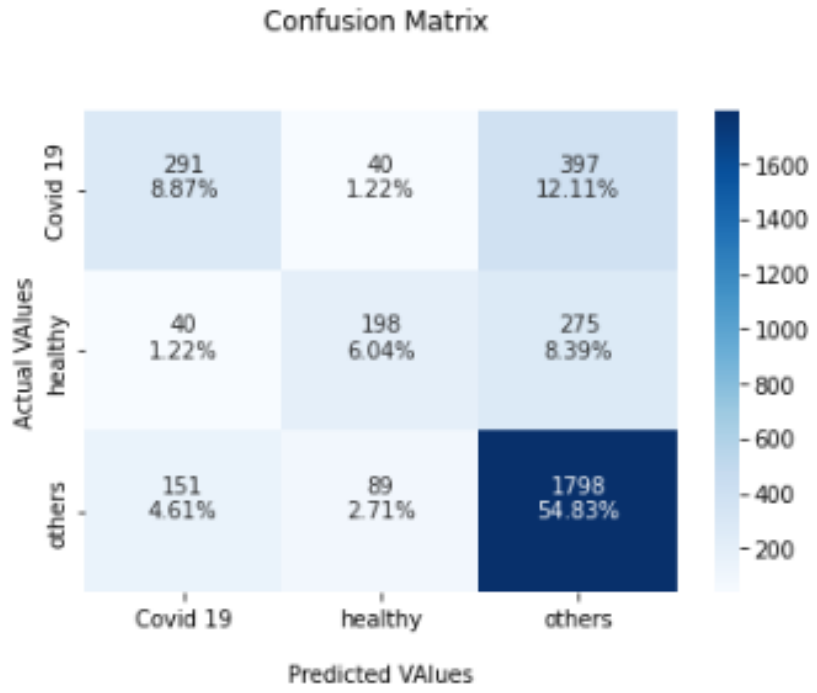


Figure 5.10: The Confusion Matrix of Extra Tree Classifier using cost-sensitive analysis with MFCC feature set.

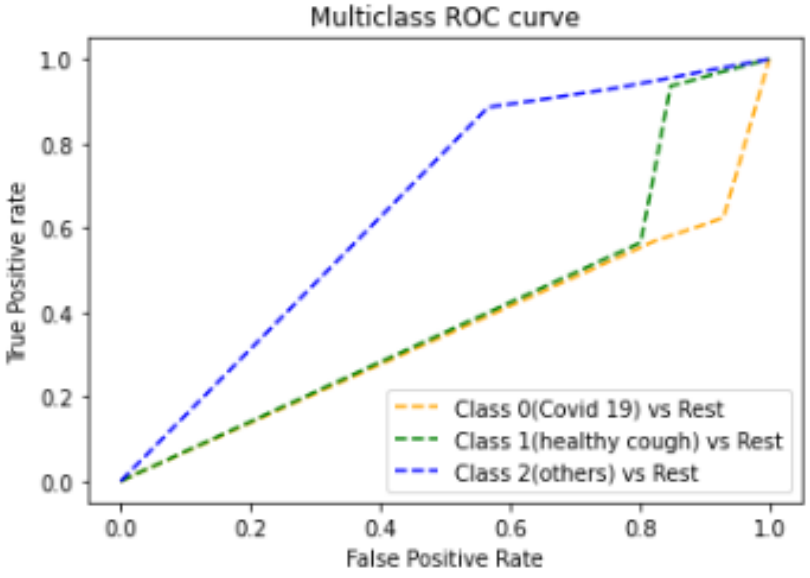


Figure 5.11: ROC Curves for Extra Tree classifier on MFCC feature set.

After applying the cost-sensitive ML algorithms to MFCC features, we applied the same set of algorithms to our combined set. Even with the combined feature set Extra Tree algorithm shows the best result in terms of COVID-19 diagnosis by analyzing cough sound accuracy improved by 9.65% from the DT algorithm. Considering all together ROC AUC score, Precision and Recall ET performs better using cost-sensitive ML algorithms. The improved AUC score for ET is 0.76. Table 5.2 shows all four algorithms with the respective performance matrix result.

ML Algorithm	Accuracy	Precision	Recall	F1 Score	Roc AUC
Extra Trees	0.6930	0.6470	0.550	0.58	0.761
Random Forest	0.6890	0.6310	0.548	0.57	0.741
SVM	0.5499	0.3363	0.413	0.37	0.685
Decision Tree	0.5965	0.5150	0.533	0.52	0.653

Table 5.2: The evaluation score of ML algorithms using the cost-sensitive method on combined feature set.

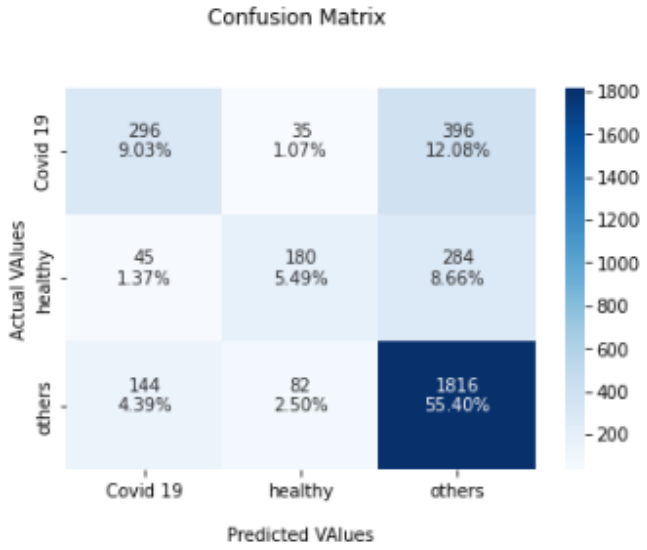


Figure 5.12: The Confusion Matrix for Extra Tree Classifier using cost-sensitive analysis on combined features set.

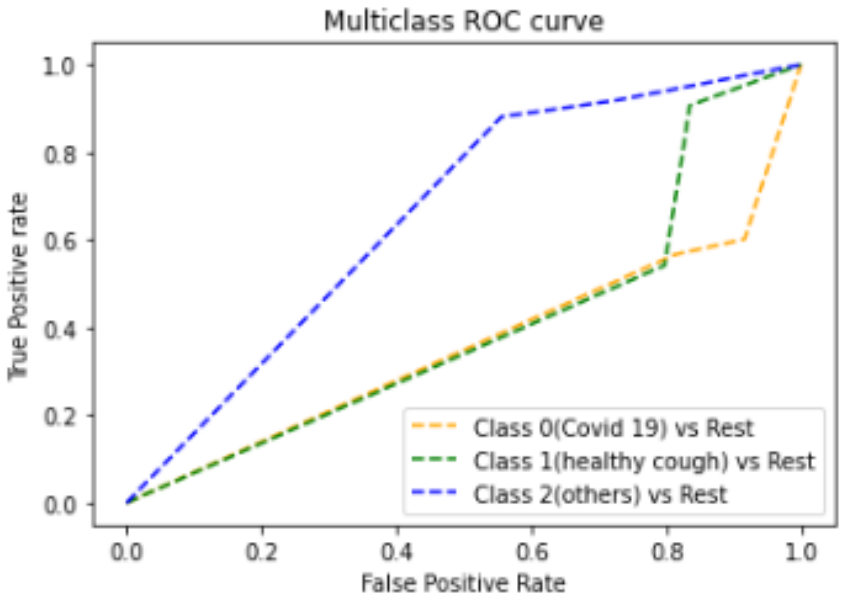


Figure 5.13: ROC Curves for Extra Tree classifier on the combined feature set.

Figure 5.12 shows the confusion matrix for the classes and Figure 5.13 shows the ROC curve which helps to visualize the specificity and sensitivity among the three classes (COVID-19, healthy cough and others).

We applied the SMOTE-Tomek link balancing technique on fourteen ML algorithms and implemented several cost-sensitive ML algorithms on both the MFCC feature set and the Combined feature set. Among all the ML models, Extra Tree Classification on a combined feature set using the SMOTE-Tomek link balancing technique provides the best set of scores for the evaluation matrix.

5.3 Deep Learning Algorithms

This section discusses the deep learning algorithms used in this thesis. We have applied the Sequential Model and RNN-LSTM algorithm. Both of the algorithms have been implemented using the MFCC feature set and Combined Feature set. Data sets have been

divided into training and validation parts. The Sequential Model showed the highest accuracy of 66.35% for the MFCC set with SMOTE-Tomek link and LSTM showed 50.35% accuracy with the MFCC dataset with the SMOTE-Tomek link Balancing technique.

We implemented a sequential model on the MFCC feature set and combined feature set. The sequential model algorithm applied to the MFCC algorithm performs better without the data balancing technique. As shown in table 5.3, MFCC without balancing technique, scored a Precision of 0.56 which is 4.5% more and Recall is 0.1% less than MFCC with the balancing technique. But in the case of the F1 score MFCC with SMOTE-Tomek link has increased up to 2.5%.

Feature Set	Data Balancing technique	Accuracy	Recall	Precision	F1 Score
MFCC	None	59.38%	0.530	0.560	0.520
MFCC	SMOTE-Tomek Link	52.85%	0.529	0.515	0.545
Combined Feature set	None	63.85%	0.580	0.550	0.550
Combined Feature set	SMOTE-Tomek Link	66.35%	0.554	0.590	0.568

Table 5.3: Summary of the evaluation score for Sequential Model algorithm with different constraints.

Now if we check it for combined feature set then with SMOTE-Tomek link it improved more as the accuracy, Precision and F1 score has improved respectively by 2.5%, 4% and 1.8%. As sequential model showed the best result with a combined feature set, so we

showed the ROC curve and confusion matrix for the combined set with the SMOTE-Tomek Link.

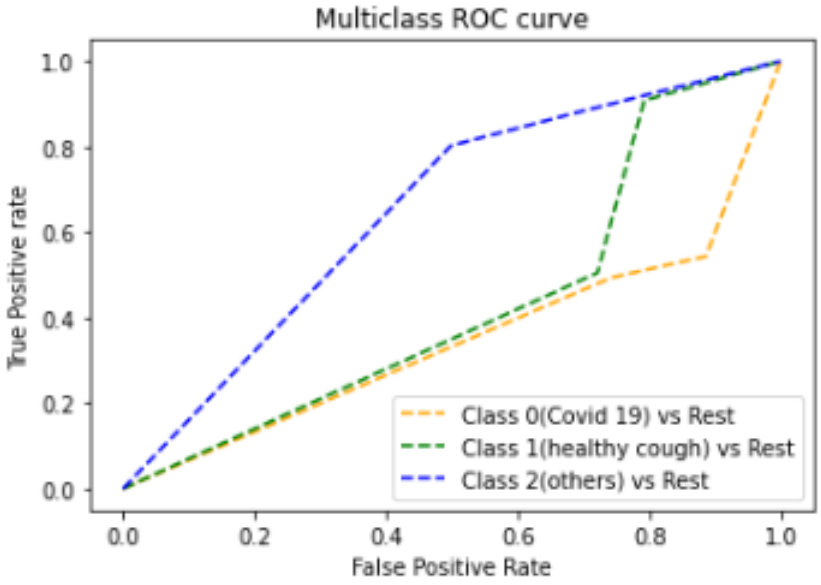


Figure 5.14: The Roc curve showing True Positive Rate (TPR) against False Positive Rate (FPR) at various threshold.

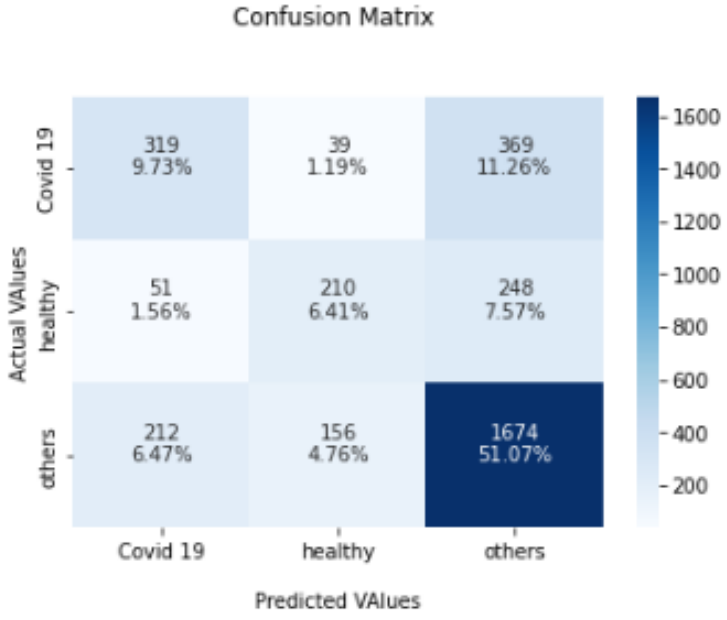


Figure 5.15: The Confusion Matrix for Sequential Model.

Figure 5.15 shows the confusion matrix for a sequential model which shows that the COVID-19 class is correctly classified 319 times and healthy cough and other category gets correctly classified 39 and 369 times. The confusion matrix helps us to visualize the recall, precision ad F1 score.

We implemented an RNN-LSTM on the MFCC feature set and combined feature set. Table 5.4 summarizes the evaluation result and shows even though RNN LSTM showed more accuracy without balancing technique for both MFCC and Combined sets, respectively 60.59% and 61.05%, the overall evaluation score is low. RNN LSTM shows the low score for Recall, precision and F1 score and all of them are below 0.5. In terms of evaluation metrics scores, the Sequential model outperforms the RNN-LSTM method among DL algorithms.

Feature Set	Data Balancing Technique	Accuracy	Recall	Precision	F1 Score
MFCC	None	60.59%	0.333	0.202	0.252
MFCC	SMOTE-Tomek Link	48.74%	0.366	0.371	0.367
Combined Feature Set	None	61.05%	0.330	0.204	0.253
Combined Feature Set	SMOTE-Tomek Link	31.66%	0.377	0.368	0.313

Table 5.4: Summary of the evaluation of RNN-LSTM algorithm.

5.4 Overall Results and Findings

Cough classification by sound analysis is a challenging classification problem in and of itself. We utilized the largest crowdsourced dataset COUGHVID, which is freely available

and annotated by healthcare experts. The first difficulties were that this dataset was multiclass and unbalanced. To address these issues, we investigated several data balancing and multiclass classification algorithms.

Method	Feature Set	Best Model	Evaluation Score
ML Algorithm without Balancing Technique	Combined Feature Set	Random Forest	Accuracy : 53.66% Recall : 0.53 Precision : 0.53 F1 Score : 0.53 AUC : 0.73
ML Algorithm with SMOTE-Tomek Link	Combined Feature Set	Extra Trees	Accuracy : 71.23% Recall : 0.5660 Precision : 0.7110 F1 Score : 0.6855 AUC : 0.7877
Cost-Sensitive ML Algorithm	Combined Feature Set	Extra Trees	Accuracy : 69.3% Recall : 0.6470 Precision : 0.5500 F1 Score : 0.5800 AUC : 0.7610
Deep Learning	Combined Feature Set	Sequential Model	Accuracy : 66.35% Recall : 0.5540 Precision : 0.5900 F1 Score : 0.5680

Table 5.5: Summary of all methods with their respective best models and evaluation Score.

Table 5.5 presents the model results for each classification strategy. ML algorithms with resampling data balancing strategies outperformed all other approaches. The Extra Trees method used for the combined feature set produced the best model, with an accuracy of 71.23%. In the event of an unbalanced dataset, the AUC score, Recall, Precision, and F1 score are more trustworthy and taken into account. The ET method has the highest AUC score by 0.7877. We may infer that our top performing COVID-19 Cough classification model is the ET algorithm with the combined feature set. To demonstrate the statistical significance of our results, we performed statistical analysis (ANOVA test [81]) on our data where H_0 denotes hypothesis where samples are equal and H_1denotes hypothesis

where samples are independent. Our derived p-value $\leq$ α (0.05), which nullifies the hypothesis H_0, and supports the validity of our results.

Chapter 6

6 Discussion and Conclusion

6.1 Summary and Discussion

As COVID-19 spread around the planet, everyone felt the impact in some manner. Researchers, medical industry specialists, and biological scientists are all working hard to discover a cure for the condition. AI is one approach that might assist in continuing this effort to evaluate academics and scientists. We discovered that it is crucial to detect COVID-19 positive illness using human respiratory sounds. As a result, AI-based approaches for diagnosing COVID-19 illness from respiratory sounds are relevant, trustworthy, and effective. By classifying cough sounds, we want to automate COVID-19 prescreening. Identifying infected persons for isolation before the physical COVID-19 test will alleviate the pressure on medical professionals. Furthermore, because patients will be able to prescreen and obtain results, the hassle of having a swab test will be reduced. As a consequence, people will be able to take the test more effortlessly. In this research, we properly cleaned both metadata and audio data. Feature extraction for COVID detection was also performed utilizing a variety of audio signal feature extraction approaches. Our metadata is annotated by four experienced physicians, and we choose the data with the majority voted diagnosis. Conflicting data were deleted to make metadata cleaner and more accurate. COUGHVID data was imbalanced and multiclass, both of which are challenges for ML classification, and we addressed these concerns using the resampling technique, a cost-sensitive classification strategy based on state-of-the-art ML algorithms, and Deep Learning. We fine-tuned the models' hyperparameters to attain the best results. We

classified the data into three categories which are COVID-19, healthy cough data and others category (means symptoms of other respiratory diseases). The extra trees method produced the best results using the feature resampling technique and combined feature set, with AUC score of 78.77%, accuracy scores of 71.23%, recall of 56.60%, and F1 score of 68.55%. As the Covid19 test is expensive, prescreening relieves patients of the stress of having to take a swab test by allowing them to prescreen and determine the outcome. This prescreening tool will ease the current wait time for physical testing and make it more accessible to everyone including the underprivileged regions. People in such areas have little if any, opportunity to test themselves and competent healthcare facilities for Covid-19 are unavailable [8].

6.2 Limitations and Future Work

As the performance of the model depends on the dataset, the lack of availability of sufficient data is considered a limitation for training our AI algorithms. As part of future work, we intend to employ a larger scale of high-quality cough data from across the world to reflect a diverse variety and community-specific phonological distinctions. Another research limitation is the quality of the training and testing data. Experts worked hard to guarantee that the data in the COUGHVID dataset was appropriately annotated. However, any inaccuracy in data labeling that escaped detection is likely to influence reported performance. Such an influence can be more evident when the data is small. The lack of available clinical datasets in the public domain is another problem. It is critical to conduct large-scale trial-based validation to verify generalization capabilities. A large-scale medically supervised covid-19 test data would be the ideal dataset for our developed models. As part of the ongoing study, our proposed framework will be integrated with a

mobile app to give a full-fledged prescreening tool to users. We plan to expand our model to include data from real-time clinical COVID-19 diagnosed patients rather than only physician-screened data. In order to improve the performance of our ML algorithms, we are going to explore ensemble techniques as well. Also, we believe that this tool has the potential to contribute to developing digital screening techniques to classify other respiratory diseases.

Bibliography

[1] "Practical Cryptography," [Online]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/. [Accessed 29 04 2022].

[2] A. Shah, M. Kattel, A. Nepal and D. Shrestha, "Chroma Feature Extraction," in *Chroma Feature Extraction using Fourier Transform*, 2019.

[3] L. Lu and A. Hanjalic, "Audio Segmentation," in *Encyclopedia of Database Systems*, Boston, MA, Springer US, 2009, pp. 167-172.

[4] Y. Zeng, H. Mao, D. Peng and Z. Yi, "Spectrogram based multi-task audio classification," *Multimedia Tools and Applications,* vol. 78, no. 3, pp. 3705-3722, 2019.

[5] "Short-Time Fourier Transform - an overview | ScienceDirect Topics," [Online]. Available: https://www.sciencedirect.com/topics/engineering/short-time-fourier-transform. [Accessed 29 04 2022].

[6] "The Short-Time Fourier Transform | Spectral Audio Signal Processing," [Online]. Available: https://www.dsprelated.com/freebooks/sasp/Short_Time_Fourier_Transform.html. [Accessed 29 04 2022].

[7] L. Orlandic, T. Teijeiro and D. Atienza, "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms," *Scientific Data,* p. 156, 2021-6-23.

[8] "In Africa at-home COVID tests are scarce and expensive, help may not come until next year," [Online]. Available: https://www.pbs.org/newshour/world/in-africa-at-home-covid-tests-are-scarce-and-expensive-help-may-not-come-until-next-year.

[9] M. Kumar, Ayesha Fatma and Nalin Bharti, "Access to Medicines and Medical Equipment during COVID-19: Searching Compatibility between the WTO and the WHO," *India Quarterly,* vol. 78, no. 1, pp. 68-78, 2022.

[10] Cascella M, Rajnik M, Cuomo A and Dulebohn SC, "Features, Evaluation, and Treatment of Coronavirus (COVID-19)," 2022.

[11] "Breath sounds: MedlinePlus Medical Encyclopedia," [Online]. Available: https://medlineplus.gov/ency/article/007535.htm. [Accessed 29 04 2022].

[12] N. Sengupta, M. Sahidullah and G. Saha, "Lung sound classification using cepstral-based statistical features," *Computers in Biology and Medicine,* vol. 75, pp. 118-129, 2016.

[13] Jpetiot, "Audio Segmentation," [Online]. Available: https://www.irit.fr/SAMOVA/site/research/analysis/audio-segmentation/. [Accessed 29 04 2022].

[14] "Audio Signal Processing- Understanding Digital & Analog Audio Signal Processing," [Online]. Available: https://www.pathpartnertech.com/audio-signal-processing-understanding-digital-analog-audio-signal-processing/. [Accessed 29 04 2022].

[15] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 25, no. 3, pp. 235-238, 1977.

[16] J. B. Allen and L. R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," in *IEEE, vol. 65*, 1977.

[17] J. B. Allen, "Application of the short-time Fourier transform to speech processing and spectral analysis," in *IEEE ICASSP-82*, 1982.

[18] P. Theodor, "Time-frequency analysis, by L. Cohen, Prentice Hall Signal Processing Series, Prentice Hall, Englewood Cliffs, New Jersey, 1995 - Book review," *Control Engineering Practice,* vol. 5, p. 292–294, 1997.

[19] mlearnere, "Learning from Audio: The Mel Scale, Mel Spectrograms, and Mel Frequency Cepstral Coefficients," 2021. [Online]. Available: https://towardsdatascience.com/learning-from-audio-the-mel-scale-mel-spectrograms-and-mel-frequency-cepstral-coefficients-f5752b6324a8. [Accessed 29 04 2022].

[20] T. Tran and J. Lundgren, "Drill Fault Diagnosis Based on the Scalogram and Mel Spectrogram of Sound Signals Using Artificial Intelligence," *IEEE Access,* vol. 8, pp. 203655-203666, 2020.

[21] R. Loughran, J. Walker, M. O'Neill and M. O'Farrell, "The Use of Mel-frequency Cepstral Coefficients in Musical Instrument Identification," 2008.

[22] P. Nair, "The dummy's guide to MFCC," 2018. [Online]. Available: https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd. [Accessed 29 04 2022].

[23] "What is Machine Learning?," 2021. [Online]. Available: https://www.ibm.com/cloud/learn/machine-learning. [Accessed 29 04 2022].

[24] P. Geurts, D. Ernst and L. Wehenkel, "Extremely randomized trees," *Machine Learning,* vol. 63, no. 1, pp. 3-42, 2006.

[25] N. Chakrabarty and S. Biswas, "Navo Minority Over-sampling Technique (NMOTe): A Consistent Performance Booster on Imbalanced Datasets," *Journal of Electronics and Informatics,* vol. 2, pp. 96-136, 2020.

[26] "Random Forest Classifier: Overview, How Does it Work, Pros & Cons," 2021. [Online]. Available: https://www.upgrad.com/blog/random-forest-classifier/. [Accessed 29 04 2022].

[27] "Machine Learning Random Forest Algorithm - Javatpoint," [Online]. Available: https://www.javatpoint.com/machine-learning-random-forest-algorithm. [Accessed 29 04 2022].

[28] "Decision Tree Classification: Everything You Need to Know," [Online]. Available: https://www.upgrad.com/blog/decision-tree-classification-everything-you-need-to-know/. [Accessed 29 04 2022].

[29] "Machine Learning Decision Tree Classification Algorithm - Javatpoint," [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. [Accessed 29 04 2022].

[30] "KNN Classifier For Machine Learning: Everything You Need to Know," 2021. [Online]. Available: https://www.upgrad.com/blog/knn-classifier-for-machine-learning/. [Accessed 29 04 2022].

[31] "#005 PyTorch - Logistic Regression in PyTorch," [Online]. Available: https://datahacker.rs/005-pytorch-logistic-regression-in-pytorch/. [Accessed 29 04 2022].

[32] "An Introduction to Logistic Regression," [Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/an-introduction-to-logistic-regression/. [Accessed 29 04 2022].

[33] "Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples," 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/. [Accessed 29 04 2022].

[34] "Gradient Boosting Algorithm: A Complete Guide for Beginners," [Online]. Available: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/. [Accessed 29 04 2022].

[35] "Gradient boosting," [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gradient_boosting&oldid=1082655359. [Accessed 29 04 2022].

[36] "Gradient Boosting for Classification," [Online]. Available: https://blog.paperspace.com/gradient-boosting-for-classification/. [Accessed 29 04 2022].

[37] "Light GBM vs XGBOOST: Which algorithm takes the crown," 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/. [Accessed 29 04 2022].

[38] "LightGBM (Light Gradient Boosting Machine)," 2020. [Online]. Available: https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/. [Accessed 29 04 2022].

[39] "Linear discriminant analysis, explained · Xiaozhou's Notes," [Online]. Available: https://yangxiaozhou.github.io/data/2019/10/02/linear-discriminant-analysis.html. [Accessed 29 04 2021].

[40] "Quadratic Discriminant Analysis," [Online]. Available: https://www.geeksforgeeks.org/quadratic-discriminant-analysis/. [Accessed July 2022].

[41] "A Guide To Understanding AdaBoost," [Online]. Available: https://blog.paperspace.com/adaboost-optimizer/. [Accessed 29 04 2022].

[42] A. J. Wyner, M. Olson, J. Bleich and D. Mease, "Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers," *Journal of Machine Learning Research,* vol. 18, no. 48, pp. 1-33, 2017.

[43] DataTechNotes, "Classification Example with Ridge Classifier in Python," [Online]. Available: https://www.datatechnotes.com/2020/07/classification-example-with-ridge-classifier-in-python.html. [Accessed 29 04 2022].

[44] "Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science," [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed 29 04 2022].

[45] "Creating linear kernel SVM in Python," [Online]. Available: https://www.geeksforgeeks.org/creating-linear-kernel-svm-in-python/. [Accessed 29 04 2022].

[46] "Introduction to Support Vector Machines (SVM) - GeeksforGeeks," [Online]. Available: https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/. [Accessed 29 04 2022].

[47] B. Tezcan, "Why Using a Dummy Classifier is a Smart Move," 2021. [Online]. Available: https://towardsdatascience.com/why-using-a-dummy-classifier-is-a-smart-move-4a55080e3549. [Accessed 29 04 2022].

[48] "A Tutorial on Sequential Machine Learning," [Online]. Available: https://analyticsindiamag.com/a-tutorial-on-sequential-machine-learning/. [Accessed 29 04 2022].

[49] "Deep Learning | Introduction to Long Short Term Memory," [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/. [Accessed 29 04 2022].

[50] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun and L. Xia, "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases," *Radiology,* pp. E32-E40, 2020.

[51] H. Luo, Q.-L. Tang, Y.-X. Shang, S.-B. Liang, M. Yang, N. Robinson and J.-P. Liu, "Can Chinese Medicine Be Used for Prevention of Corona Virus Disease 2019 (COVID-19)? A Review of Historical Classics, Research Evidence and Current Prevention Programs," *Chinese Journal of Integrative Medicine,* 2020-02-17.

[52] J. Laguarta, F. Hueto and B. Subirana, "COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings," *IEEE Open Journal of Engineering in Medicine and Biology,* pp. 275-281, 2020.

[53] I. A. and P. I, "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app - PMC," *Informatics in Medicine,* 2020.

[54] P. Wang, . X. Zheng and G. Ai, "Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran - PMC," *Chaos, Solitons & Fractals,* vol. 140, pp. 110214,, 2020.

[55] D. Ho, "Addressing COVID-19 Drug Development with Artificial Intelligence - Ho - 2020 - Advanced Intelligent Systems - Wiley Online Library," *Advanced Intelligent System,* 2020.

[56] A. S. Adly, A. S. Adly and M. S. Adly, "Approaches Based on Artificial Intelligence and the Internet of Intelligent Things to Prevent the Spread of COVID-19: Scoping Review," *Journal of Medical Internet Research,* 2020-8-10.

[57] Shannon Najmabadi and Jay, "Coronavirus test results in Texas are taking up to 10 days," 2020. [Online]. Available: https://www.texastribune.org/2020/03/28/coronavirus-test-results-texas-are-taking-10-days/.

[58] "Post Washington. Hospitals are overwhelmed because of the coronavirus. Accessed on: Mar. 31, 2020.," [Online]. Available: https://www.washingtonpost.com/opinio ns/2020/03/15/hospitals-are-overwhelmed-because-coronavirus-heres-how-help/; 2020..

[59] "CNN. FDA authorizes 15-minute coronavirus test. Accessed on: Mar. 31, 2020.[Online].," [Online]. Available: https://www.cnn.com/2020/03/27/us/15-minute-coronaviru s-test/index.html; 2020..

[60] " Abott. Detect COVID-19 in as little as 5 minutes. Accessed on: May. 30, 2020.[Online]. Available,," [Online]. Available: https://www.abbott.com/corpnewsroom/product-and-inno vation/detect-covid-19-in-as-little-as-5-minutes.html; 2020..

[61] " STAT. FDA says Abbott's 5-minute COVID-19 test may miss infected patients [Online]. Available,," [Online]. Available: https://www.statnews.com/2020/05/15/fda-saysabbotts- 5-minute-covid-19-test-may-miss-infected-patients/.

[62] "Coronavirus Test Obstacles: A Shortage of Face Masks and Swabs - The New York Times," [Online]. Available: https://www.nytimes.com/2020/03/18/health/coronavirus-test-shortages-face-masks-swabs.html.

[63] "shortages-face-masks-cotton-swabs-basic-supplies-pose-new-challenge-coronavirus-testing," [Online]. Available: https://www.washingtonpost.com/climate-environment/2020/03/18/shortages-face-masks-cotton-swabs-basic-supplies-pose-new-challenge-coronavirus-testing/.

[64] "Coronavirus (COVID-19) Update: FDA Authorizes First Diagnostic Test Using At-Home Collection of Saliva Specimens," [Online]. Available: https://www.fda.gov/news-events/press-announcements/coronavirus-covid-19-update-fda-authorizes-first-diagnostic-test-using-home-collection-saliva.

[65] Y. Wang, M. Hu, Q. Li, X.-P. Zhang, G. Zhai and N. Yao, "Abnormal respiratory patterns classifier may contribute to large-scale screening of people infected with COVID-19 in an accurate and unobtrusive manner," *arXiv:2002.05534 [cs, eess],* 2020-12-20.

[66] Z. Jiang, M. Hu, Y. Pan, W. Tang, G. Zhai and Y. Lu, "Combining Visible Light and Infrared Imaging for Efficient Detection of Respiratory Infections such as COVID-19 on Portable Device," *arXiv:2004.06912 [cs, eess],* 2020-04-15.

[67] C. Brown, J. Chauhan, A. Grammenos and J. Han, "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* pp. 3474-3484, 2020-08-23.

[68] M. Bader , I. Shahin and A. Hassan , "Studying the Similarity of COVID-19 Sounds based on Correlation Analysis of MFCC | Semantic Scholar," *arXiv: 2010.08770.,* 2020.

[69] M. A. Ismail, S. Deshmukh and R. Singh, "Detection of COVID-19 through the analysis of vocal fold oscillations," *arXiv:2010.10707 [cs, eess],* 2020-10-20.

[70] T. F. Quatieri, T. Talkar and J. S. Palmer, "A Framework for Biomarkers of COVID-19 Based on Coordination of Speech-Production Subsystems," *IEEE Open Journal of Engineering in Medicine and Biology,* pp. 203-206, 2020.

[71] J. Han, K. Qian, M. Song and M. Song, "An Early Study on Intelligent Analysis of Speech under COVID-19: Severity, Sleep Quality, Fatigue, and Anxiety," *arXiv:2005.00096 [cs, eess],* 2020-05-14.

[72] Ritwik, K. V. Sai, Shareef Babu Kalluri and Deepu Vijayasenan, "COVID-19 Patient Detection from Telephone Quality Speech Data," 2020.

[73] G. Chaudhari, X. Jiang, A. Fakhry, A. Han, S. Shen and A. Khanzada, "Virufy: Global Applicability of Crowdsourced and Clinical Datasets for AI Detection of COVID-19 from Cough," *arXiv:2011.13320 [cs, eess],* 2021-01-09.

[74] Celik, Devrim, Nina Mainusch and Xavier Oliva i Jürgens, "The COUGHVID crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms," 2020. [Online]. Available: https://zenodo.org/record/4048312.

[75] Hassan A, Shahin I and Alsabek MB, "COVID-19 Detection System using Recurrent Neural Networks," 2020.

[76] R. A. A. Viadinugroho, "Imbalanced Classification in Python: SMOTE-Tomek Links Method," 2021. [Online]. Available: https://towardsdatascience.com/imbalanced-classification-in-python-smote-tomek-links-method-6e48dfe69bbc. [Accessed 29 04 2022].

[77] J. Brownlee, "Undersampling Algorithms for Imbalanced Classification," 2020. [Online]. Available: https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/. [Accessed 29 04 2022].

[78] "Confusion Matrix for Machine Learning," [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/. [Accessed 29 04 2022].

[79] "Confusion matrix," [Online]. Available: https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=1058352752. [Accessed 29 04 2022].

[80] V. Trevisan, "Interpreting ROC Curve and ROC AUC for Classification Evaluation," 2022. [Online]. Available: https://towardsdatascience.com/interpreting-roc-curve-and-roc-auc-for-classification-evaluation-28ec3983f077. [Accessed 29 04 2022].

[81] B. Jason, "17 Statistical Hypothesis Tests in Python (Cheat Sheet)," Machine Learning Mastery, 2018. [Online]. Available: https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/.

Curriculum Vitae

Name	:	Annita Tahsin Priyoti
Post-Secondary Education and Degrees	:	Military Institute of Science and Technology Dhaka, Bangladesh 2013-2016 B.Sc The University of Western Ontario London, Ontario, Canada 2020-2022 M.Sc
Honours and Awards	:	Western Graduate Research Scholarship (WGRS) 2020
Related Work Experience	:	Teaching Assistant The University of Western Ontario 2020-2022
Publications	:	1. Sajal Saha, Annita Tahsin Priyoti, Aakriti Sharma, and Anwar Haque. "Towards an Optimal Feature Selection Method for AI-Based DDoS Detection System." In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 425-428. IEEE, 2022. 2. Saikat Das, Sajal Saha, Annita Tahsin Priyoti, Etee Kawna Roy, Frederick T. Sheldon, Anwar Haque, and Sajjan Shiva. "Network Intrusion Detection and Comparative Analysis using Ensemble Machine Learning and Feature Selection." *IEEE Transactions on Network and Service Management* (2021).