Deep Learning Anomaly Detection methods to passively detect COVID-19 from Audio

by

Shreesha Narasimha Murthy

A thesis submitted in partial fulfilment for the degree of
Master of Science
in
Data Science
May 10, 2021

APPROVED:

Professor Emmanuel O. Agu, Advisor

Professor Elke A. Rundensteiner, Reader

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own. I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at https://www.wpi.edu/about/policies/academic-integrity.

Signed: ____________________ Date: ____________________

Abstract

The world has been severely affected by COVID-19, an infectious disease caused by the SARS-Cov-2 coronavirus. COVID-19 incubates in a patient for 7 days before symptoms manifest. During this incubation period, affected individuals, unknowingly, transmit the virus through respiratory droplets released when the individual coughs or sneezes, which has resulted in a record number of daily cases around the world. The identification of the presence of COVID-19 is challenging as its symptoms are similar to influenza symptoms such as cough, cold, runny nose and chills. COVID-19 affects human speech sub-systems involved in respiration, phonation, and articulation. This master thesis proposes a deep anomaly detection framework for passive, speech-based detection of COVID-related anomalies in voice samples of COVID-19 affected individuals. The low percentage of positive cases and extreme imbalance in available COVID audio datasets present a challenge to machine learning classifiers but creates an opportunity to utilize anomaly detection techniques. This thesis investigates COVID detection from audio using various types of deep anomaly detectors and autoencoders. Contrastive loss methods are also explored to force our models to learn the discrepancies between COVID and non-COVID cough data representations. In rigorous evaluation, the variational autoencoder with the elliptic envelope as the anomaly detector analyzing Mel Filterbanks audio representations performed best with an AUC of 65.7, outperforming the state of the art.

Acknowledgements

This thesis wouldn't have seen the light if not for the support from some wonderful people. Firstly, I would like to thank my advisor Professor Dr. Emmanuel O. Agu for his constant support and guidance throughout the duration of my research. Whenever I digressed, missed my track, or hit a creative block, the professor, with his calming influence and creative inputs always steered me back on track. I am taking with me his knack for simplifying things, not just in research but in life. I am grateful for the support I received from Professor Dr. Elke A. Rundensteiner. The support I received from DARPA was immensely helpful in focusing only on conducting successful research. Besides my advisors, I would also like to thank Apiwat Dittapron for his valuable inputs from time to time.

Lastly, undeterred support from my family and friends helped me reach my goals successfully. My wife, sister, parents, and in-laws have stood by me in all my decisions in the past two years. A sincere heartfelt thank you to everyone for being a part of this journey.

This material is based on research sponsored by DARPA under agreement number FA8750-18-2-0077. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government

Contents

List of Figures

List of Tables

1 | Introduction

1.1 Motivation

As of today, confirmed COVID-19 cases across the world have exceeded 61 million. This alarming number of cases has led to increased testing, diagnosis, and screening of individuals on a large scale. As assessed by the department of mental health and substance abuse at WHO, individuals are at high risk of falling into depression, inflicting self-harm, experiencing loneliness, and are at elevated risk of stress and anxiety due to quarantine and lockdown. Physicians need to spend long working hours with such individuals on treatments and cures. The World Health Organization(WHO)[1] has declared that effective solutions need to be developed at a rapid pace to curb the virus and further control its spread. They have also come up with a list of symptoms that can be used to identify the presence of the COVID-19 virus in the human body. Breathing difficulties, cough, muscle pain, chills, loss of taste, sore throat, and high body temperature are among the symptoms that commonly appear when the virus takes over an individual's immune system. However, the uniqueness of the virus makes it difficult to ascertain the list of symptoms as the effects of the virus on individuals vary a great deal. The symptoms appear only after an incubation period of 6 to 7 days. During this time, the virus-affected individual stays asymptomatic.

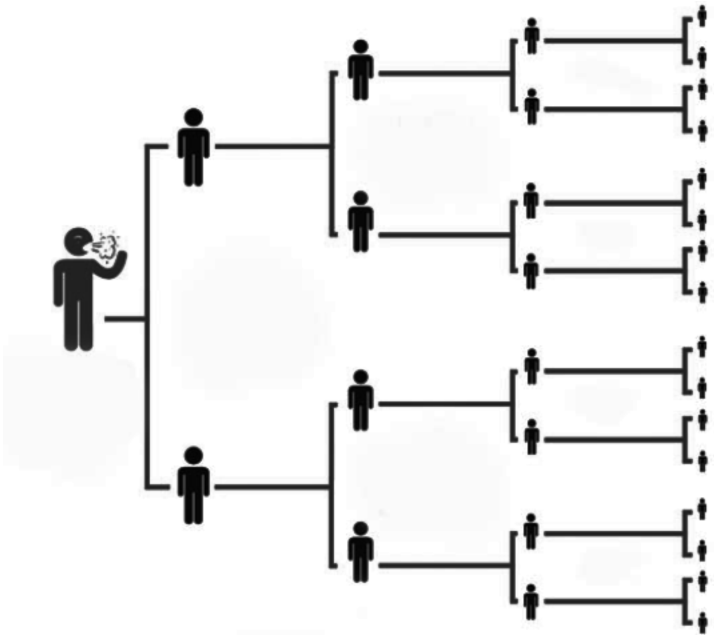


Figure 1.1: COVID-19 proliferating effect

1.1.1 How COVID-19 spreads?

As shown in Fig. 1.1, COVID-19 transmits from person to person who is in close contact within 6 meters through respiratory droplets when the infected person talks, sings, breathes, or coughs. Other less common ways of spread are airborne transmission, transmission via contaminated surfaces, and through people in enclosed spaces with poor ventilation. Since infected people are initially asymptomatic and the virus spreads from person to person, it is lethal and needs immediate effective ways to control the spread and treat its symptoms. These uncertainties and adverse effects create the need for effective ways to identify individuals who are afflicted with the coronavirus and stay asymptomatic during the virus' incubation period. Artificial Intelligence has shown some positive signs in detecting such asymptomatic individuals and helping isolate them from healthy individuals thus helping curb the spread of the virus. [2]. In the next section we will introduce in general how AI is used in Healthcare, and discuss data types(Audio, Time series data) used in passive health monitoring systems. Finally we will discuss how AI specifically helps in COVID-19 detection.

1.1.2 AI in Healthcare

Artificial Intelligence predict outcomes well in advance if it is shown enough data. Collecting data has been a challenge in Healthcare systems due to a plethora of challenges like privacy issues, dearth in technology to passively collect data etc. These challenges gave rise to devices called Wearables, [3] a segment of commercially available products used in passive data collection. They are made in the form of smartwatches [4], smart shirts, etc. that sit on a human body and track certain primitive yet prominent metrics like Heart Beat, muscle movements, fall tracking, etc. These devices are usually not widely adopted mainly due to the extra cost associated with them. This, on the other hand, motivates researchers to innovate solutions with resources that are currently available with almost every individual, a smartphone. The penetration level of smartphones and the sensors embedded in them, along with efficient wireless communication technologies have made continuous passive health monitoring easier than before with negligible additional costs [5]. More specifically, smartphones have been used to predict the onset of FLU symptoms [6] a day before the person turns symptomatic. Authors of this work effectively show that continuously feeding data from proximity sensors, WiFi access points, masked SMS, and call records, for obvious privacy reasons, can help predict if the user of the smartphone will show signs of FLU infection. In another similar work [7], authors use smartphone mobility features to detect depression in individuals using a Deep Anomaly detection technique. They essentially formulate a deep learning system to tackle the imbalance in the dataset while still achieving an AUC score of 0.92. In the section we discuss more on work that uses audio in passive health monitoring systems.

1.1.3 Audio in Healthcare

Furthermore, audio as a modality has been widely used for continuous health monitoring in a variety of tasks. In one such work [8] authors use voice samples to screen depression in patients affected by Parkinson’s disease. In another interesting problem, researchers use human voice to identify if the user is intoxicated from alcohol [9]. In a similar study [10], human voice is used to detect individual's sleepiness, a fatal state to be in for individuals working in critical systems. The technique used in the paper achieves an accuracy of 75% in detecting sleepiness. With such large scale adoption of human audio for detecting series of diseases, researchers from MIT Lincoln Laboratory proposed [11] showed in their paper the effects of COVID on human vocal system. This motivated the audio community to utilize learning methods to analyze human audio to find discrepancies in voice signals, ultimately leading to detection of COVID-19.

1.1.4 COVID-19 detection from Audio and AI

There is rapid growth in research towards the identification of COVID-19 using human voice. This paper[11] showed how COVID-19 affects human voice. The voice is generally modelled in three phases: Respiration, Phonation and Articulation. These are the essential vocal subsystems involved in speech production. Fig 1.2, taken from the paper, shows the progression of respiration(Top row) and fundamental frequency(Bottom row). Readings of Individual subjects on the right and their mean values on the left. It is evident that respiration and fundamental frequency show a downward pattern with the onset of COVID. Adding to that, many research papers published recently provide elegant solutions which are successful in identifying the presence of COVID-19 in an individual through human sounds like Cough and Breathing.[12, 13, 14, 15]. In chapter 2 we discuss in more detail on the performance of these proposed methods.

1.2 Thesis Goal

Inspired by the recent breakthroughs in Deep learning, feature extraction from highly unstructured and high dimensional data like text, image and audio has been easy. With the extracted features, many tasks such a time-series forecasting, anomaly de-

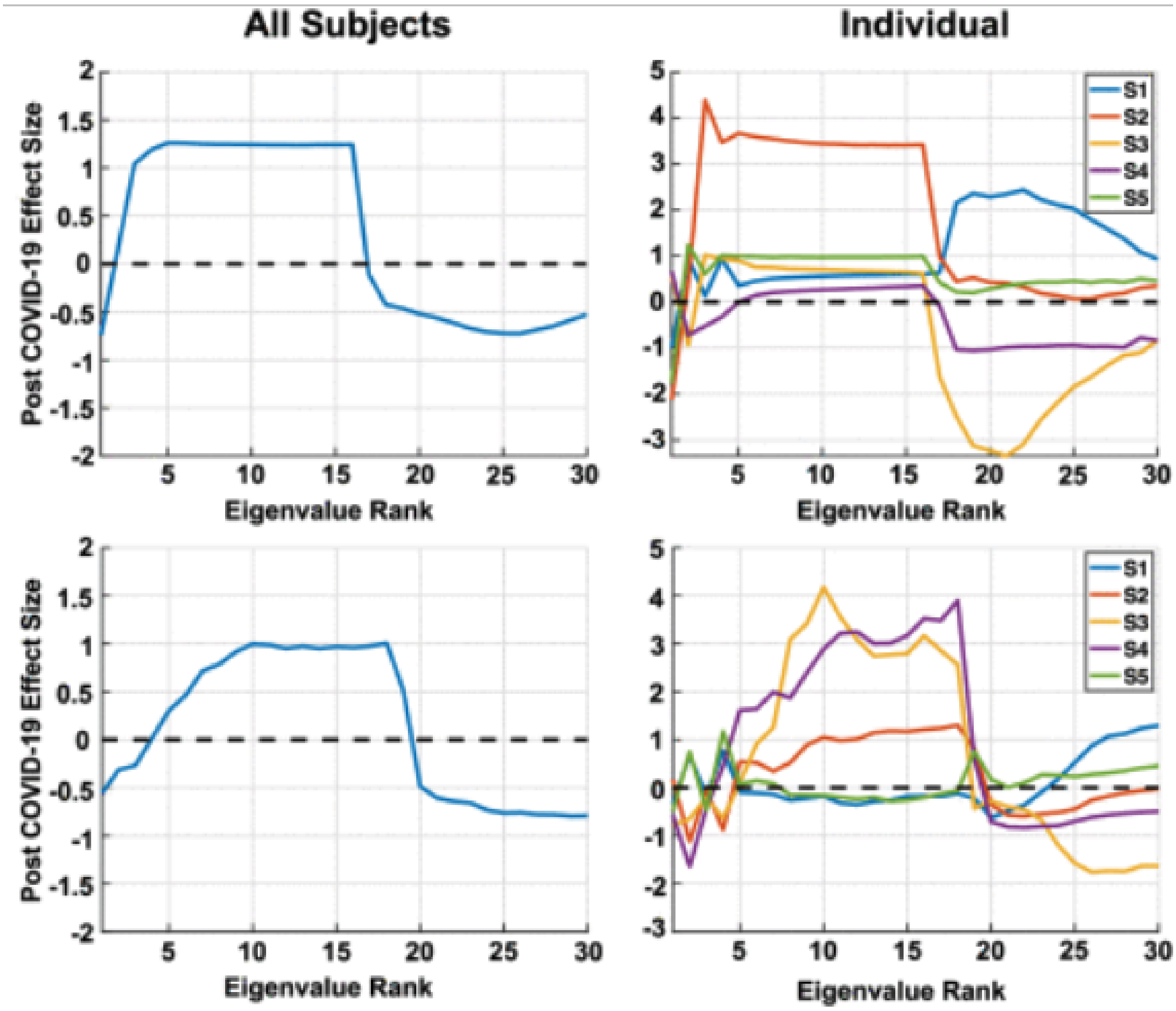


Figure 1.2: Affect of COVID on speech subsystems.
On top: Respiration intensities. On Bottom: Fundamental Frequency

tection, natural language understanding, classification tasks with complex decision boundaries have shown superior performance.

Our work explores different types of Deep Autoencoder architectures to extract meaningful features from a high dimensional spaio-temporal space that MFCCs and Mel Filters encode, and use the extracted features train traditional anomaly detectors to find COVID-19 cases(anomalies). More specifically, we address two main research questions:

- Are there significant statistical differences found in the voice of a COVID-19 affected individual to a healthy individual?
- How likely can this statistical difference be used to classify between these individuals with the application of modern deep learning techniques?

Answers to these questions were broken down into these practical steps:

- Convert the raw audio signals into Mel coefficients, namely MFCCs and Mel Filterbanks
- Use a Convolutional Autoencoder or one of its variant to compress the audio representation into a one dimensional array, the Bottleneck, which is then suitable to use in Traditional Anomaly detectors.
- Use appropriate metrics to measure the model's performance

This work has been built on top of certain underlying assumptions. Below is a comprehensive list:

1. We rely on research from [11] that scientifically shows the differences in voice reproduction in the vocal cords of a COVID-19 affected individual
2. Data recorded by MIT and IISC are good representatives of the population, and capture maximum voice variations, human demographics and different stages of COVID-19 affected individuals.

COVID-19 Questionnaire
Do you have cough?
Have you been sneezing recently?
Have you been tested for COVID-19 by a medical professional?
Do you smoke?
Do you have any lung ailments?
Do you have any neurological problems?
Do you have any neuro-muscular problems?

Table 1.1: Symptom Survey. All questions were Yes/No/Skip responses

1.3 COVID-19 Audio Dataset

These parts about the specific dataset and exploratory data analyses should be part of your methodology section.

With the onset of COVID-19, the need for rapid analysis and testing of COVID-affected patients has been at the forefront of health research. Massachussetts Institute of Technology has created a web interface[16] for the general public to upload their voice samples and answer a questionnaire as shown in Table 1.1. This helps self-label the recorded audio sample into COVID-19 and non-COVID-19 categories. The Indian Institute of Science(IISC) also maintains a repository of COVID-19 and non-COVID-19 voices called Coswara. In this work, we explore in detail MIT and Coswara datasets. We thoroughly analyze these two datasets answering questions discussed in section 1.2. These datasets capture human audio in multiple acoustic forms like breathing, cough, alphabets, vowel pronunciation. For the COVID detection task we mainly focus on using cough audio data. Prior studies show that [17, 18, 19] respiratory ailments with cough as one of its symptoms have distinct underlying features, and can be extracted using appropriate signal processing techniques like Mel Spectrograms and low level descriptors.

1.4 Proposed Solution

As seen in the previous section, MIT and IISC COVID-19 datasets are highly imbalanced. Data imbalance is a major factor to consider while developing a reliable and robust deep learning system. Factoring this concern, treating this as a binary classification problem would lead to a highly skewed model, considering penalizing cost functions and employing dropouts. As a result, we approach solving this problem through Anomaly detection methods, treating the minority class samples as Anomalies. Anomaly detection is a field in data mining that detects data points in any dataset that display unusual behavior i.e. data points that deviate from normal behavior. In our case, we treat COVID negative observations as normal behavior. Further in this study, we show the statistical difference visually between the majority and minority classes of the audio samples, which backs our approach to solve the problem using Anomaly detection.

1.4.1 Proposed System Architecture

In the previous sections, we established the need for continuous health monitoring and how it can easily be done because of the high penetration levels of smartphones. In this section, we brief the entire system architecture used in this research for continuous health monitoring. Fig 1.3 gives a high-level view of all the components involved in this research, right from the source of data to the final decision made. In an ideal real-world scenario, audio signals from the smartphone's mic are picked up and fed to an audio preprocessor which converts the raw audio signals into Mel spectrogram features and generates low-level descriptors like fundamental frequency, shimmer, jitter, zero-crossing rate, and pitch. All the processed audio features are then fed to the Deep Anomaly detector which consists of an Autoencoder for dimensionality reduction and an Anomaly detector like EllicticEnvelope, OneClass-SVM, etc. Based on the output from this component, we decide if it is necessary to notify the user or continue monitoring passively. We explain each component in the Audio preprocessor and Deep

Anomaly detector in greater detail in the chapter 4

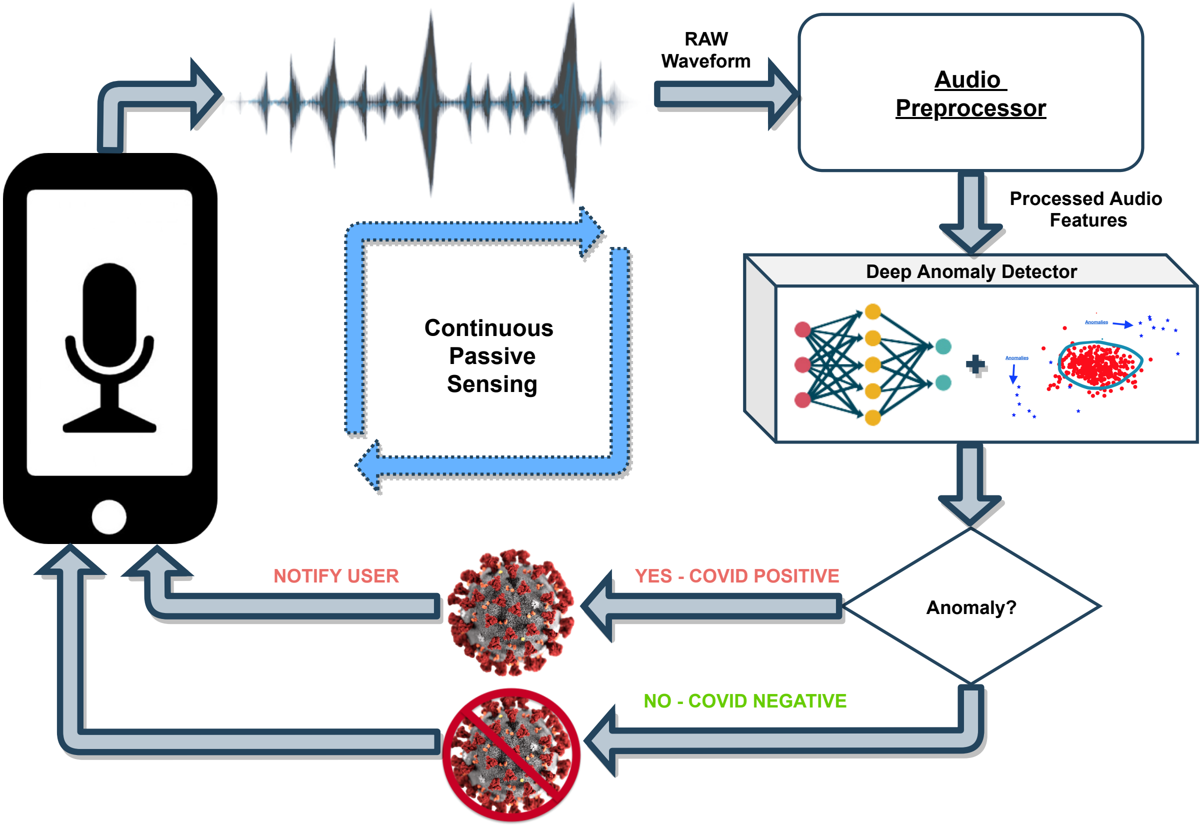


Figure 1.3: Passive COVID-19 detection network

1.4.2 Contributions

The main contributions of this research are as follows:

1. Analyze voice samples in COVID-19 datasets sourced from smartphones, IISC[20] and MIT[16]

2. Investigate the effect of COVID-19 on human voice across various signal encoding strategies. Listed below are the ones explored in this study

   (a) Mel Frequency Cepstral Coefficients, Mel Filter Banks [21, 22]
   Motivation: Previous research[17, 18] shows cough audio has more energy in lower frequencies. Mel scale, with its frequency band provides higher resolotuin in lower frequencies and vice versa. This makes Mel coefficients a suitable choice for cough audio representation

    (b) Acoustic low level descriptors such as Pitch, Fundamental frequency(f_0), Zero crossing rate, Shimmer and Jitter[23]

        Motivation: Low level descriptors mentioned above capture statistical, structural, temporal and frequency based attributes helping models understand the overall structure of the underlying audio.

3. Explore Autoencoders and similar variants for feature extraction and Dimensionality reduction. More specifically, explore

    (a) Convolutional Autoencoders(AE)

        Motivation: As discussed above, for the inherent capability it possesses to extract high dimensional spatio-temporal features.

    (b) Convolutional Variational Autoencoders (VAE)

        Motivation: Along with the above discussed advantages of a Convolutional AE, VAEs have the ability to encode the latent representation.

4. Employ traditional Anomaly detectors for the final classification step. Below are the ones explored in this study.

    (a) One class SVM

    (b) Isolation Forest

    (c) Elliptic Envelope

1.5 Challenges in Detecting COVID using Audio

Considering the risk factors when solving a critical system use case where False negative predictions have serious impacts, this study faces some inherent challenges. Some of them are discussed below

1. **Overlapping symptoms**

   Main symptoms in COVID are cough, cold, sneezing, fatigue, loss of taste and smell, headache, respiratory issues like shortness of breath and congestion during breathing. Some of these symptoms overlap heavily with other general problems like non COVID fever, fatigue due to other health reasons, respiratory issues due to Asthma or Bronchitis. In addition to this, it is possible for people having underlying respiratory disorders to get infected with COVID. With the scenarios discussed above, it is necessary to build systems that detect COVID not just based on presence of overlapping symptoms rather based on more deeper voice variations.

2. **Imbalanced datasets**

   As of now, 150 million people out a 7 billion have been affected with COVID i.e 2% of the entire population has been affected with COVID-19. As a result both MIT and IISC Coswara datasets have a high class imbalance ratio. This problem has been researched thoroughly[24]. Models built on top of such datasets are not only biased towards the majority class, but suffer from poor predictive performance, often resulting in all predictions falling under the majority class. To counter this issue, we consider solving this as an Anomaly detection problem instead of a plain binary classification, treating minority class samples as Anomalies.

3. **Difficulty with generating synthetic audio data to mitigate data imbalance**

   Since the datasets we are dealing with are related to Human health, and resulting actions from model predictions are highly sensitive. Sampling techniques like SMOTE[25] that generate samples from data distribution of a class synthetically, although fall in the same distribution, does not guarantee patterns of a COVID-19 audio sample.

4. **Variable quality of audio recorded by different equipment and situations**

   Both MIT and Coswara datasets ask subjects to upload audio through a website. Since each individual uploads data independently, factors like device used for recording, quality of recording, ambience, data loss play an important factor in receiving reliable audio for analysis. However, utmost care is taken by both dataset providers to weed out unreliable data. But it is unsafe to assume this will be the case when models are used in real-time as we use audio directly from an individual's smartphone where we do not have control over audio quality, ambience etc.

5. **Privacy concerns**

   Another issue to consider when proposing to use this model in real-time is to acquire the consent of the user before recording audio, either from personal devices or recording studios.

2 | Related Work

As described in the motivation section, the widespread adoption of smartphones resulted in a cheap, reliable and accessible route to continuously monitor human health and act in case of emergencies. Smartphones and other consumer technology devices have long been used by researchers to sense and monitor various illnesses, mental health issues. An individual's constant interaction with their personal devices like smartphones makes it even more favourable to collect enormous amounts of data sufficient for medical professionals to draw conclusions and act on it. Another main advantage of a smartphone is that it eliminates the need to carry a separate camera, mic, GPS or speakers, as smartphones come with all these built-in making it easier for researchers to build apps that once installed in a smartphone pulls data from inbuilt sensors to suit the needs of any study. In 2012, a NASA post [26] revealed that a consumer smartphone had 100 times the compute power of an average NASA satellite.

2.1 Smartphones for Mental Health Screening

Smartphones are playing a major role in modernization of mental healthcare. A study conducted to examine the feasibility, acceptability, and utility of behavioral sensing in individuals with schizophrenia [27]. Researchers used data from accelerometers, Bluetooth, WiFi, microphone, and GPS to collect behavioral and contextual data to assist schizophrenia patients in tracking their condition and improve their cognitive ability. 95% of the times subjects were comfortable receiving assistance from the smart-

Paper	Task	Data	Accuracy
Ben-Zeev et al.	Feasibility, acceptability of behavioral sensing of users with Schizophrenia	Data from Accelerometers, Bluetooth, WiFi, microphone, and GPS	95% of users accepted passive sensing
Saeb et al.	Estimate a semantic location of a patient with anxiety and depression	Data from GPS signals, light, movement and sound	AUC - 0.88
Gerych et al.	Depression Detection	Mobility features like location, speed, total distance travelled	AUC - 0.92

Table 2.1: Summary of work in mental health screening using smartphones

phone app. Another study uses [28] sensor data like GPS signals, light, movement and sound from smartphones to estimate a semantic location of a patient with anxiety and depression issues to find correlations between triggers of anxiety to location. Data from 208 participants was assessed for over 6 weeks. The research predicts the semantic location of a subject with an AUC of 0.88 given the 4 features. However, they conclude that nature of places visited explains only a small part of variation in anxiety and depression symptoms. Another study [29] uses StudentLife dataset [30], a smartphone sensor dataset to identify depression in individuals. The dataset is from a single classroom of 48 students across a 10 week term at Dartmouth College using Android phones. Priliminary findings from the dataset owners reveals a significant correlation in smartphone sensor data, mental health and educational outcomes of a student body. The study utilizes these correlations by training a Deep Autoencoder to first reduce the dimensions, and then use a one class SVM to detect depressed individuals as anomalous or outliers and not depressed individuals as inliers, with an AUC of 0.92. This work significantly outforms traditional machine learning approaches. In summary, table 2.1 gives an overview of some of the smartphone based mental health screening work we refer to during our thesis.

2.2 Smartphones for influenza sensing

With the sudden onset of COVID-19 virus, the need to passively sense influenza symptoms even before it takes over the human immune system is more important now that it ever has been. influenza patterns can be analyzed from multiple data sources avail-

Paper	Task	Data	Accuracy
Gianni et al.	Detect Influenza in an Individual	Mobility features like daily displacement, movement radius, unique places visited, number of people met	F1 - 0.73
Murthy et al.		Data from proximity sensors, WiFi access points and SMS and call records	AUC - 0.76

Table 2.2: Summary of work in FLU sensing using smartphones

able on the internet. WHO FluNet [31], weekly influenza report from CDC [32], and Google Flu Trends [33]. More specifically, prediction of influenza before its manifestation has been one of the most important research fields. Gianni et al. [34] used sensors from a subject's smartphone to extract mobility features such as total daily displacement, subject's movement radius, unique places visited per day, number of people met per day to predict the onset of an Influenza before its manifestation on the human body. The research used machine learning models, more specifically Gradient Boosted Trees to predict the occurance of symptoms a day in advance, using historical feature data listed above. They achieved an F1 score of 0.73. In a similar study, the authors from MIT orchestrated a dataset [35] comprising of data collected from 70 subjects over an entire academic year. The data consisted of a constant feed from proximity sensors, WiFi access points and SMS and call records of a subject's smartphone, along with a questionnaire that subjects self reported related to their diets, exercise routines, eating habits and finally if they had any influenza symptoms, which acted as the dependent feature. The research attempted to predict influenza symptoms a day in advance using LSTM Autoencoders was made in 2021 [6]. The work uses all the sensor data mentioned above along with the questionnaire to create behavior specific clusters and built LSTM Autoencoders for subjects in each cluster to eliminate ambiguity in human behavior, with an AUC of 0.76. In summary, table 2.2 gives an overview of some of the smartphone based influenza sensing work we refer to during our thesis.

2.3 Human Audio in Healthcare

Human sounds have long been used by researchers to diagnose human health and behavior. Respiratory system auscultation has been used to diagnose irregularities in lungs [36]. Audio recordings from Smartphones, External Mic systems are used to collect, study, and measure the changes in articulation and enunciation of Human sounds. Diagnosing these sounds and drawing meaningful conclusions and decisions from these sounds often require expertise with clinical backgrounds. However, recent developments have shown that analytics around audio has gained significant popularity and can effectively be used as a potential alternative for diagnosing and detecting abnormalities in Human sounds. Recently, smartphones have been a mainstream source for audio-related diagnosing for two major reasons (1)Data is readily available without the requirement of external hardware. (2) A large population uses smartphones making data less scarce. This also guarantees data sourced is real.

In this study,[37] smartphone audio is used to understand users' ambiance by aggregating information from audio to make up city ambiance. In another study named Emotionsense [38], the phones' microphone is used as a sensor for detecting users' emotion in-the-wild, through Gaussian mixture models. In [39] authors analyze sounds emitted while the user is sleeping, to identify sleep apnea episodes. Similar works have also used sound to detect asthma and wheezing [40]. Using Convolutional Neural Networks, many deep learning methods have been built to recognize and classify sounds of coughs and different respiratory diseases[41] within ambient audio, especially to recognize three potential illnesses (bronchitis, bronchiolitis, and pertussis) based on their unique audio characteristics. Another recent study uses a google audio set that extracts audio from 1.8 million youtube videos and combines it with a free sound audio database[42] to perform tuberculosis cough detection. They convert raw audio into MFCCs and Mel filter banks and feed it to CNNs, achieving an AUC of 94% [43]. Speech during situations like car accidents, domestic violence, or situations close to death has been used for stress detection tasks using low-level feature descrip-

Paper	Task	Data	Performance
Chon et al.	Place tagging to corresponding categories (store, restaurant)	Data from proximity sensors, WiFi access points, GPS, and smartphone Audio	Accuracy - 69%
Rachuri et al.	Emotion Recognition	Data from GPS, Bluetooth, Accelerometer, audio from Microphone	AUC - 0.76
Nandakumar et al.	Detection of Sleep Apnea	Frequency modulated sound signals	R^2 - 0.995
Oletic et al.	Asthma Monitoring	Respiratory sounds	Accuracy - 80%
Bales et al.	Respiratory illness detection from Cough	Raw Cough sounds	F1- 0.85
Miranda et al.	Tuberculosis Detection from Cough	Raw Cough sounds	AUC - 0.94
Partila et al.	Stress detection from speech	Speech from emergency phone calls	87.9%

Table 2.3: Summary of work in health monitoring using Audio

tors extracted from OPENSMILE [44]. This work reports an accuracy of 87.9% with Support Vector Machines and 87.5 with traditional CNNs in classifying stress from neutral speech. Table 2.3 summarizes the literature we referred in our thesis.

3 | Background

3.1 Audio Pre-processing and Representations

3.1.1 Mel Frequency Cepstral Coefficients(MFCC) and Mel Filter Banks

Humans do a good job at identifying small changes in lower frequencies than the changes in higher frequencies. MFCCs and Filter banks specialize in mimicking the human way of percieving an audio signal. They take into account human perception for sensitivity at appropriate frequencies by converting the conventional frequency to Mel Scale, and are thus suitable for speech analysis tasks quite well. Generally 12-13 Mel Frequency coefficients are taken into consideration as features when training models. The process of converting raw audio into MFCC or Mel Filter bank is the same. Below, we briefly explain the steps involved:

1. Break the raw audio signal into short-time frames by applying a windowing function at fixed time intervals. The frequencies of a signal change over-time, additionally, it is observed that signals in short frame(typically 20ms) are stationary, hence working on short frames instead of the entire signal helps capture the changes in frequency of a signal efficiently.

2. Apply Discrete Fourier Transform on each of frame. This step helps calculate the frequency spectrum of the short-time frame

3. Apply a set of 40 triangular filters on the frequency spectrum to extract frequency bands. The positions of these filters are equally spaced along the Mel Fre-

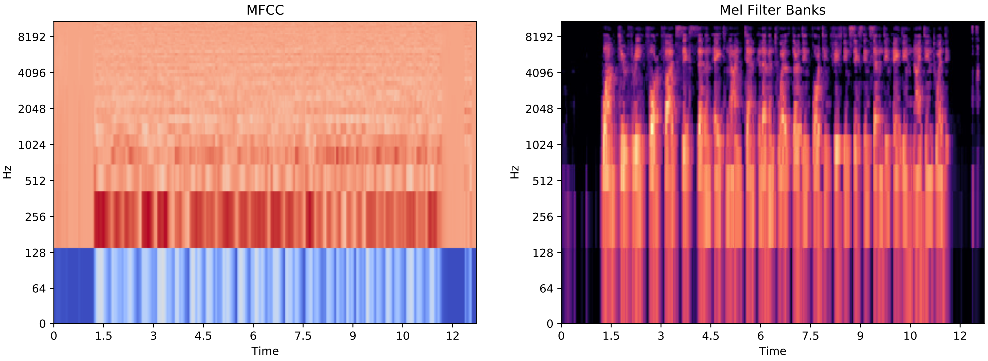


Figure 3.1: MFCCs and Mel Filter Banks

quency. Mel-scale mimics the way humans perceive sound by paying less emphasis at higher frequencies and more emphasis at lower frequencies. Frequencies(f) can be converted into Mel(m) using eq 1

$$m = 2595 * \log_{10}(1 + \frac{f}{700}) \tag{1}$$

4. At this point we have calculated the Mel Filter bank representation. However, filter bank coefficients are highly correlated, which might be problematic for some machine learning algorithms.

5. Apply Discrete Cosine Transform(DCT) to decorrelate the cofficients, resulting in Mel Frequency Cepstral Coefficients.

Fig 3.1 shows the visual difference between MFCCs and Filter Banks

3.1.2 Low Level Descriptors

Pitch & Fundamental Frequency: Pitch of a human voice is defined as the rate of vibration of the vocal folds. As rate of vibrations change the sound of human voice changes. On the other hand, fundamental frequency(f_o) refers to the approximate frequency of the periodic structure of human voice. When airflow in the vocal folds is suitably tensed, an oscillation originates giving rise to f_o.

Zero-Crossing rate: The number of times in a given time interval the amplitude of

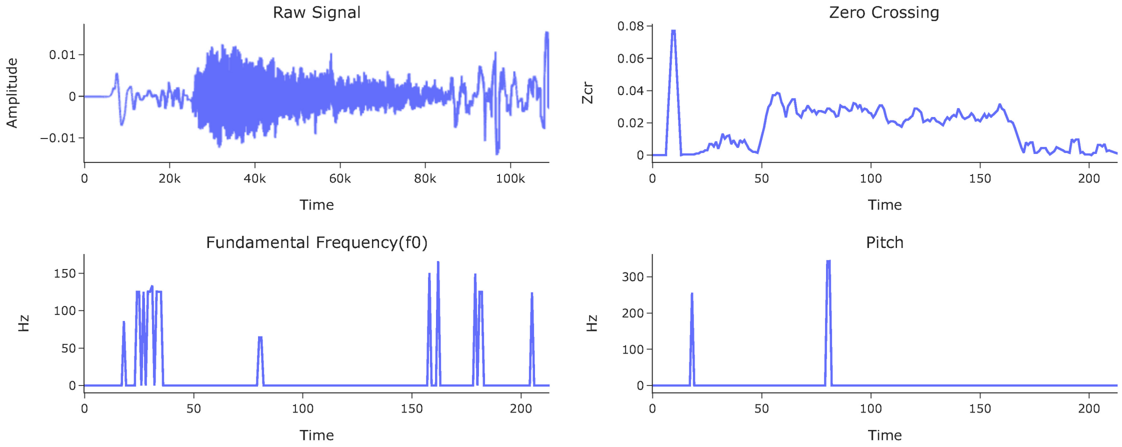


Figure 3.2: Low Level Descriptors

human voice passes through the value of zero is termed as Zero-crossing rate.

Fig 3.2 shows the visual representation of Zero-Crossing rate, Pitch and Fundamental Frequency.

Shimmer and Jitter: The cycle-to-cycle fluctuations in pitch is referred to as Jitter. It can be extracted by measuring the fundamental frequency(f_o) of each cycle of vibration and deducting it from the previous f_o values. Shimmer, on the other hand, can be extracted by measuring the variability in the signal's peak to peak amplitude.

Pitch, Fundamental Frequency, Shimmer and Jitter have been used widely in detecting Depression, Intoxication and similar pathological changes in voice.[45]. Initial results have shown that these acoustic low level descriptors of a human voice is affected when the person is infected with COVID-19 [46]

3.1.3 Raw Audio Signals

Raw audio waveforms have been used in speech recognition systems to account for the information loss in MFCC, GFCC and Mel filters. End to end trainable Deep learning methods have proved to be an efficient way to perform speech recognition using Raw audio signals[47]

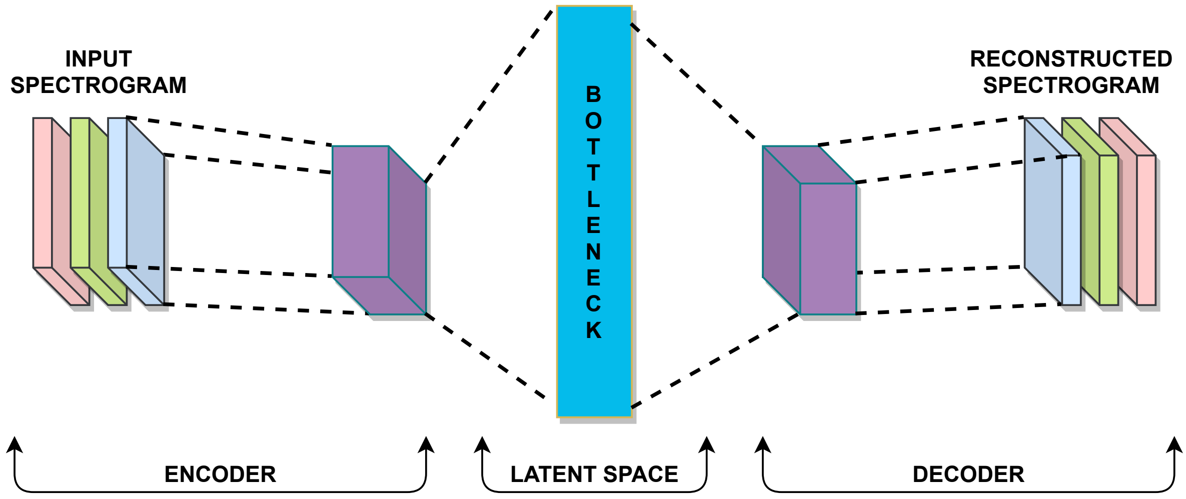


Figure 3.3: Convolutional Autoencoder

3.2 Autoencoders

Autoencoder is an unsupervised artificial neural network mainly used for dimensionality reduction in non-linear high dimensional datasets. Mainly, it consists of two components - An Encoder and a Decoder. The Encoder compresses an input representation into a lower dimensional space, called as Bottleneck. The decoder then reconstructs the input representation back using the Bottleneck. Learning to perform reconstruction is what Autoencoders specialize in[48]. They learn to ignore the noise in input data, forming the so called Bottleneck. Fig. 3.3 shows the structure of an Autoencoder. Stochastic Gradient Descent[49] and Back Propagation[50] is used in training the Autoencoder to minimize the loss function, called the reconstruction loss. The type of Neural Networks used for encoder and the decoder varies based on the end goal. Feed forward Neural Network architectures are used for one dimensional data where each row corresponds to one input sample. On the other hand, to extract spatial features from a image data, Convolutional Neural Networks are employed. Our work explores around exploiting the potential capabilities of using Convolutional Neural Networks for encoder and decoder components as described in Fig3.3.

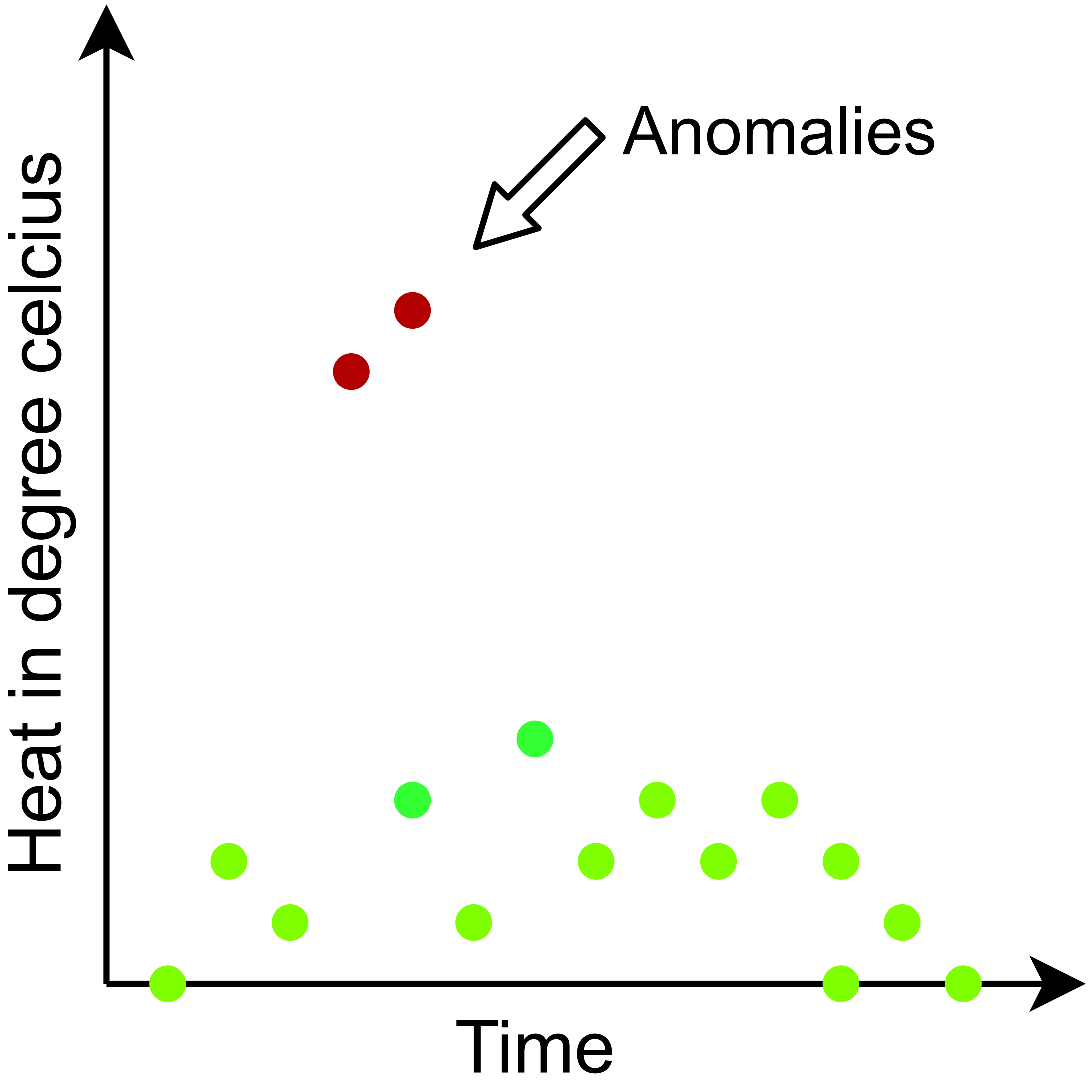


Figure 3.4: Visualization of anomalies in simple 2-d datasets

3.3 Traditional Anomaly detectors

Anomaly detection is the identification of observations that tend to display abnormal characteristics compared to majority of observations in a dataset. Typically, anomalous observations look almost similar to normal observations but they display unrealistic behavior at random intervals. Malicious activities, security breakdowns, hardware failures are some reasons why data can contain anomalies. Critical systems such as hardware installed in Hospitals, powerplants, aircrafts etc. need to be identified and warned well before failure to avoid catastrophic outcomes.

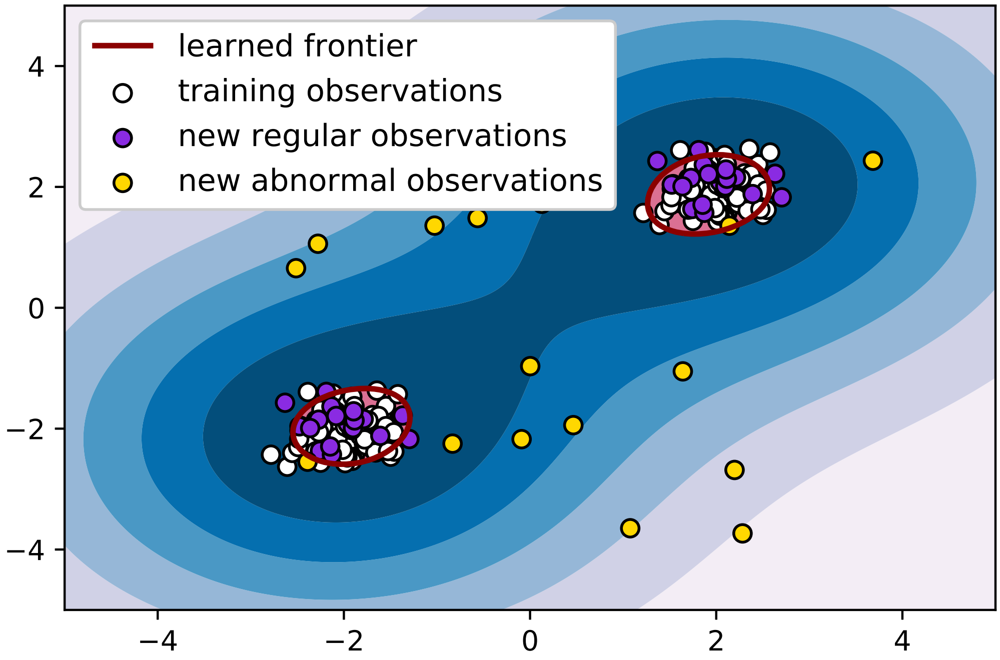


Figure 3.5: Demonstration of decision boundaries in One class SVM

In simpler datasets, a mere visualization of the data gives enough proof to identify anomalies as shown in fig 3.4 However, in higher dimensional datasets with hundreds of variables visualization is not a practical solution. Research in the field of Machine learning and Statistics have given birth to many amazing Anomaly detectors [51]. Specifically, in this study we will explore two anomaly detectors.

3.3.1 One Class Support Vector Machines

Support vector machine is a well known machine learning algorithm for classification and regression problems. In case of a binary classification, Support vector machines find a line or hyperplane based on the input data observations to differentiate between two classes. However, highly imbalanced datasets produce extremely biased models. To combat this issue, One Class Support Vector Machines [52] were introduced. OC-SVMs take only one class of data for training. The basic analogy is to separate the input data observations from a reference origin point and draw a hyperplane that maximizes the distance between input data points and the origin. This essentially creates regions in the input space where the probability density of the input data exists. Any data observation not falling within this probability density is marked an anomaly. Fig 3.5, sourced from scikit-learn [53], shows decision boundaries that One class SVM learns based on the dimensionality of underlying data points. It performs anomaly detection reasonably well on high dimensional data. One of the main reasons for this are Kernel functions used to train these models. Kernel functions are a set of mathematical methods that transform data and its dimensions so that linear decision boundaries can be drawn to non-linear data. The most basic kernel function are linear kernels, which is just a dot product of feature vectors. However, with the assumption that our the latent space representation is not linearly separable, we explore two kernel functions that are robust against non-linearity:

1. Polynomial Kernel

   These kernels are used to project similarity present in input data in terms of polynomials of original variables, allowing to learn non-linear complexities. They are defined as shown in eq 2

   $$K(x, y) = (x^T y + c)^d \tag{2}$$

   x and y are any two vectors from input space. $c \geq 0$ is a bias coefficient. d is the degree of polynomial space

2. Radial Basis Function

   RBF kernels project input space in higher dimensions based on the distance of a data point from the origin. This function is defined as shown in eq 3

   $$K(x, y) = e^{-\gamma \|x-y\|^2} \tag{3}$$

   $\|x - y\|^2$ is the squared Euclidean distance between two input vectors. γ is the kernel coefficient.

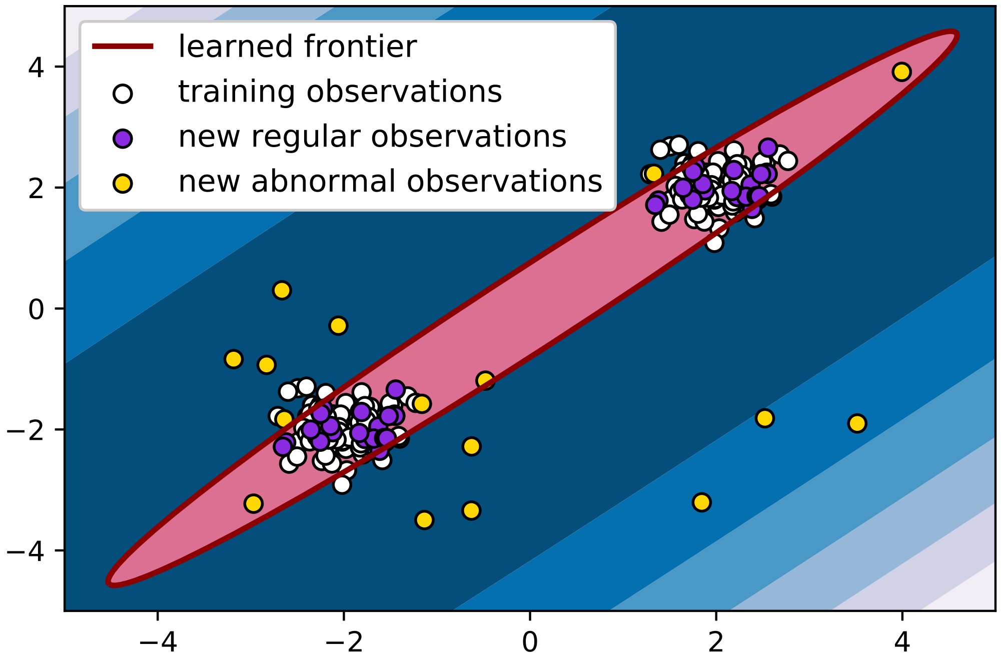


Figure 3.6: Demonstration of decision boundaries in Elliptic Envelope

3.3.2 Elliptic Envelope

Datasets in real-world are not always normally distributed. But in cases where we know a dataset is normally distributed, it becomes easy to detect abnormalities. However, primitive techniques like visualization as shown in Fig 3.4 might still not be a practical solution due to higher dimensionality in data. But techniques like Elliptic Envelope when the underlying data is normally distributed because the algorithm is built on an assumption that input data is uni-modal, specifically with a zero mean and unit variance. It works by modelling the input data as a high dimensional Gaussian and draws an elliptical boundary around the modelled data. Any data observation falling outside of elliptical boundary is marked as an anomaly. Fig 3.6,, sourced from scikit-learn [53], in general demonstrates the decision boundaries created by EllipticEnvelope with the underlying assumption that input data is normally distributed.

4 | Methodology

4.1 Overview

The predefined goal of this thesis is to formulate a continuous health monitoring system to detect COVID-19 infections within the time frame of it entering the Human body to showing initial symptoms i.e. the incubation period of 4 to 7 days. We leverage the MIT, IISC Coswara datasets, and Convolutional Autoencoders with Anomaly detectors to achieve this goal. Further, we thoroughly explore MIT and Coswara datasets, and later jump into our experimental architectures.

Column name	**Range or Categories**
Individual's Age	10 years to 89 years
Gender	Male, Female, Other
Height	Value in centimeters
Country	Israel, United States, India, Canada United Kingdom
Smoking Habits	I currently smoke, I used to smoke, I have never smoked
COVID-19 Diagnosis(target)	**Yes, No**

Table 4.1: **MIT dataset demographics**

4.2 Exploratory Data Analysis

4.2.1 MIT dataset

Massachusetts Institute of Technology sourced an audio dataset[16] containing voice samples of patients affected with COVID-19 and of individuals who had no history of COVID-19. Table 4.1 gives an overview of the demographics of the participants representing the data. Out of the 1877 participants, 28 individuals had been affected with COVID-19 at the time of recording voice samples, making it a highly imbalanced dataset for any binary classification task. Each participant has recorded their voice in 5 different variations, namely,

1. Cough samples
2. Pronunciation of all English alphabets
3. Prolonged pronunciation of Vowels - A, E, O

Each subject in the dataset was instructed to record specific variations of their voice. Fig 4.1 displays the number of audio samples that are COVID-19 positive and COVID-19 negative. Fig 4.2 gives an insight of how each demographic feature is distributed in the dataset. More than half of our subjects of the dataset are within the age of 20-40 years. Around 300 audio samples are from subjects above the age of 60, which CDC categorizes this age group as high risk individuals, especially the ones with pre-existing health conditions. On the other hand, there is a 2:1 ratio in the number of voice samples when it comes to Gender. A study on 3,111,714 globally reported COVID-19 cases shows that there is no concrete evidence on which gender is more suspectible to COVID-19. But, amongst the COVID-19 affected individuals, male patients have almost three times the odds of requiring intensive treatment unit (ITU) or admission to a medical facility to contain the damage on lungs and respiratory tract. [54]. Tobacco intake through smoking has highly been associated with higher risks of COVID-19. MIT data contains information about smokers, however as seen in Fig 4.2 more than half

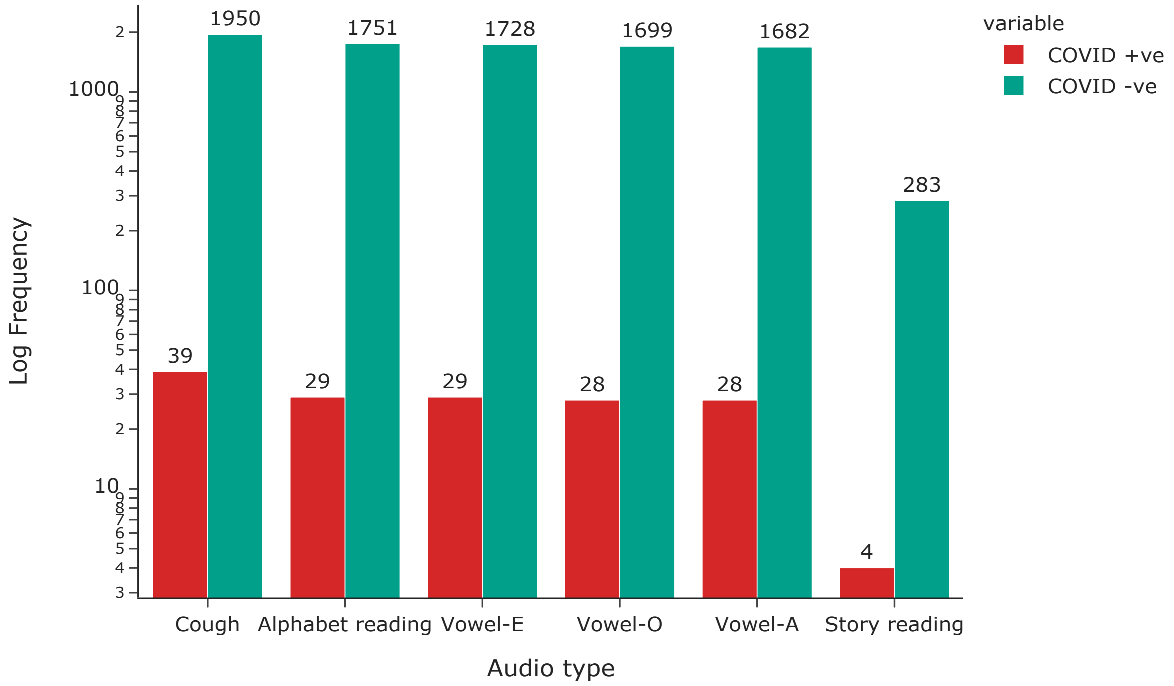


Figure 4.1: MIT COVID-19 diagnosis for each voice variation

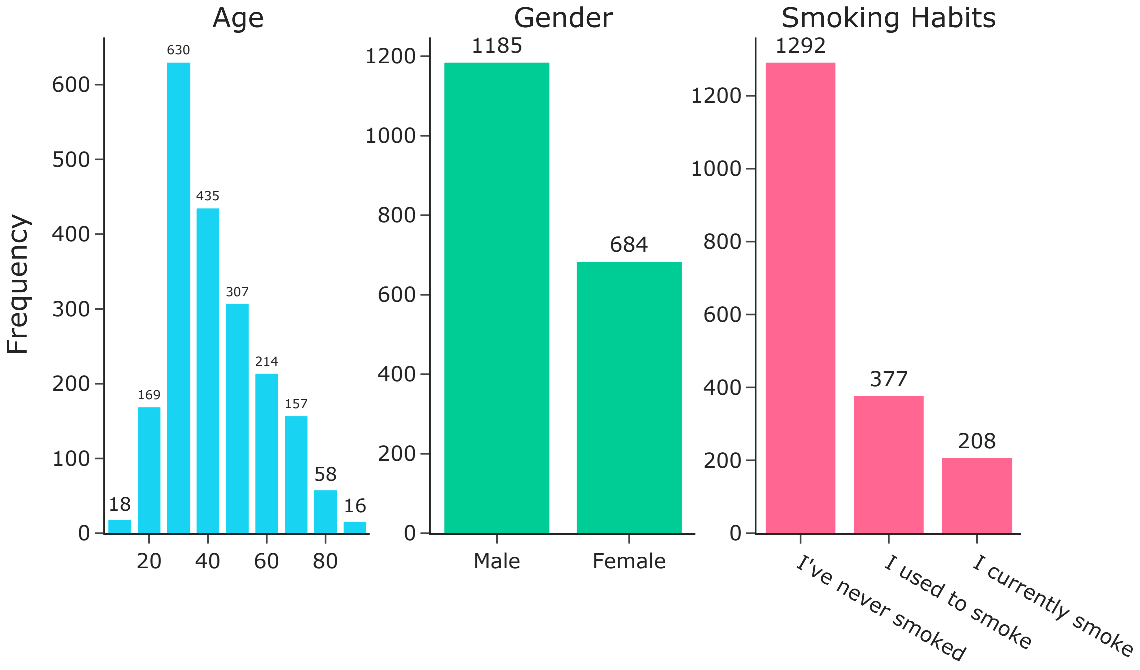


Figure 4.2: MIT dataset demographics

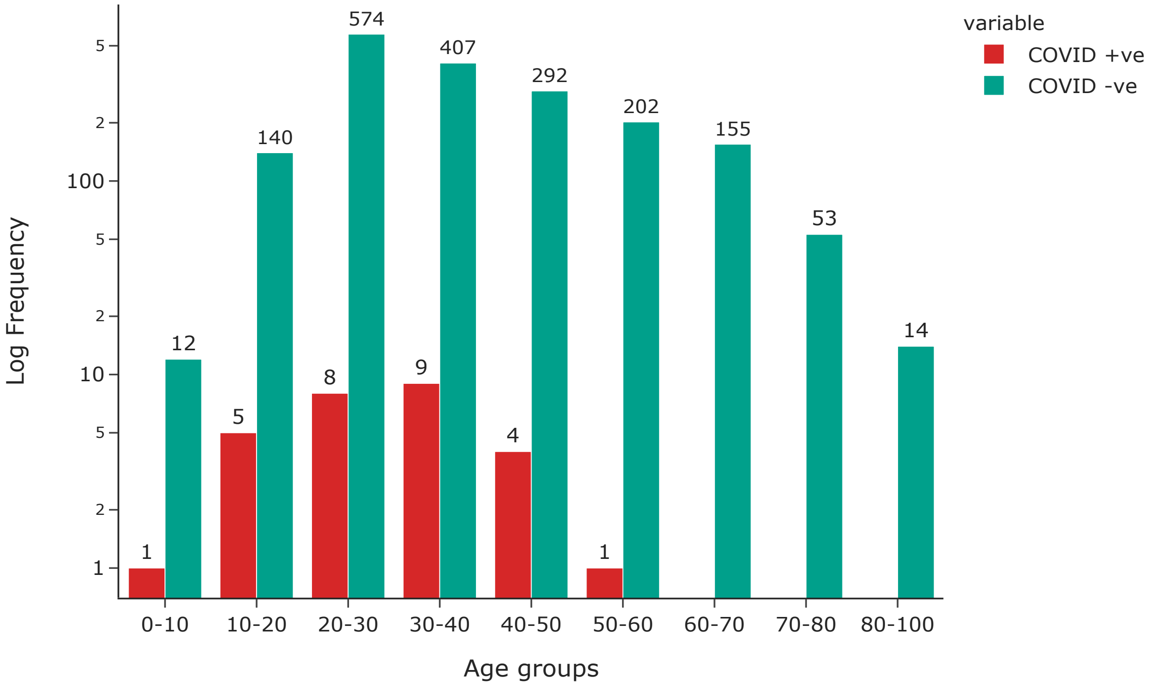


Figure 4.3: MIT COVID-19 diagnosis across all age groups

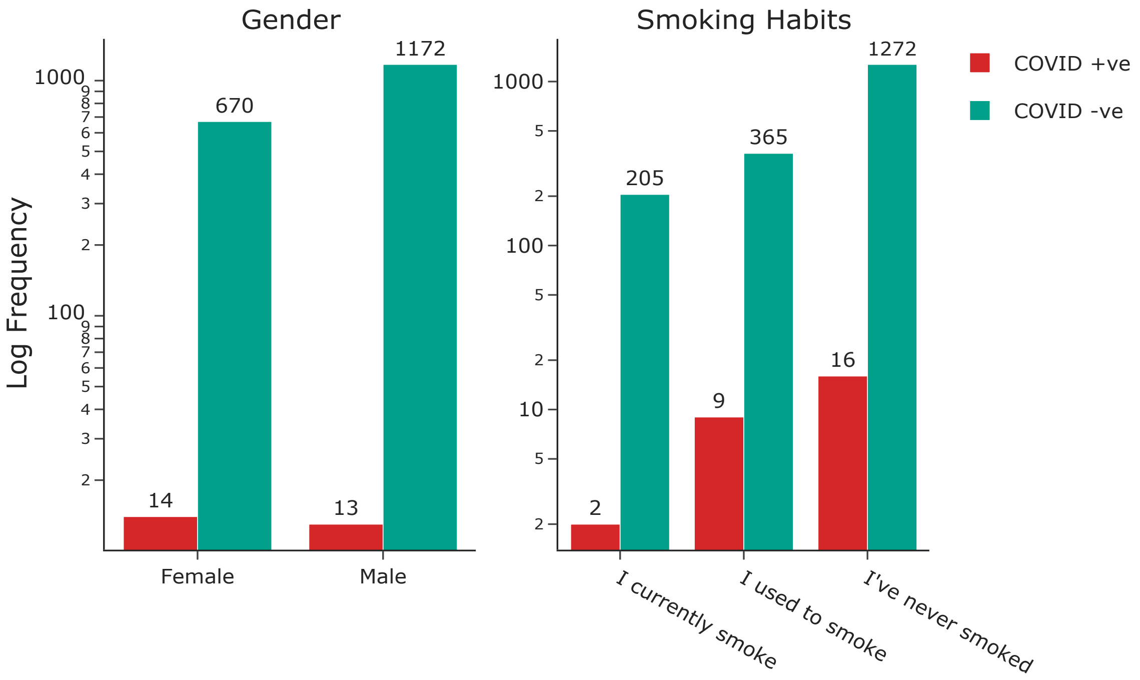


Figure 4.4: MIT COVID-19 diagnosis across gender and smoking habits

Column name	**Range or Categories**
Individual's Age	18 years to 80 years
Gender	Male, Female, Other
Proficient in English	Yes, No
Country	India
Asthma	Yes, No
Cough	Yes, No
Smoker	Yes, No
COVID-19 status(target)	**Healthy, COVID-19, Other respiratory illness**

Table 4.2: **IISC Coswara dataset demographics**

of the subjects are non smokers and only 13% of the subjects were actively smoking at the time of data recording.

Further, the dataset is divided based on the subject's COVID-19 diagnosis. Fig 4.3 and Fig 4.4 portray the total number of COVID-19 positive to COVID-19 negative samples across different age groups, gender and smoking Habits. Clearly, we can notice that there is a high imbalance between positive and negative cases.

4.2.2 IISC Coswara dataset

Indian Institute of Science, Bengaluru[55] put together a similar audio dataset[20] having 1459 participants, out of which 122 individuals had contracted COVID-19 at the time of recording. Table 4.2 shows the demographics of the dataset. Each participant recorded multiple voice variations. Below is a list of recorded variations:

1. Breathing - Deep and Shallow
2. Count - Fast and Normal
3. Cough - Heavy and Shallow
4. Vowel - A, E, O

Fig 4.7 displays counts across each voice variation. Unlike in the MIT dataset, the variations here are more uniformly distributed with each variation having around

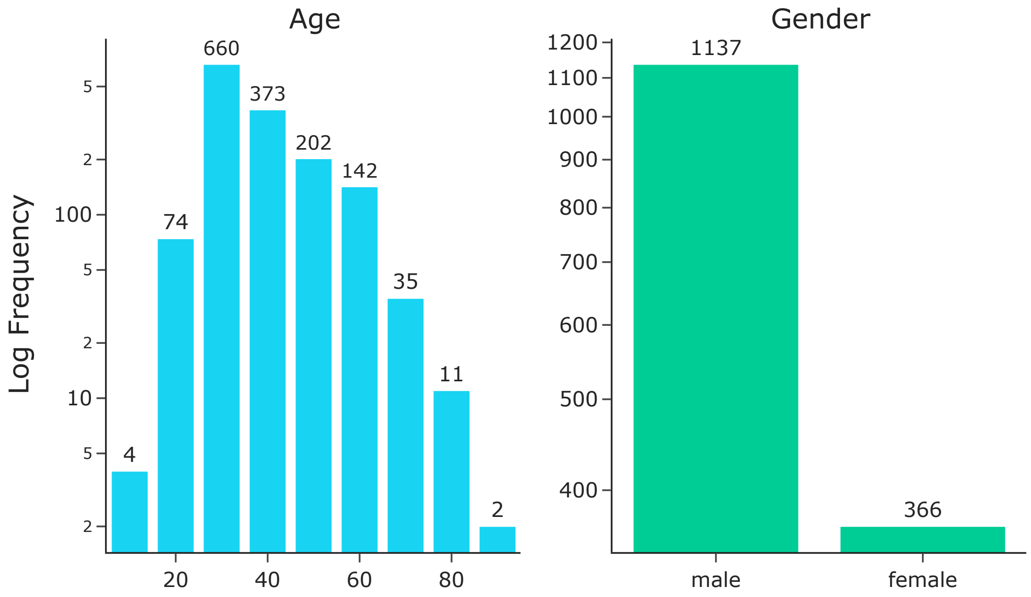


Figure 4.5: Coswara dataset demographics

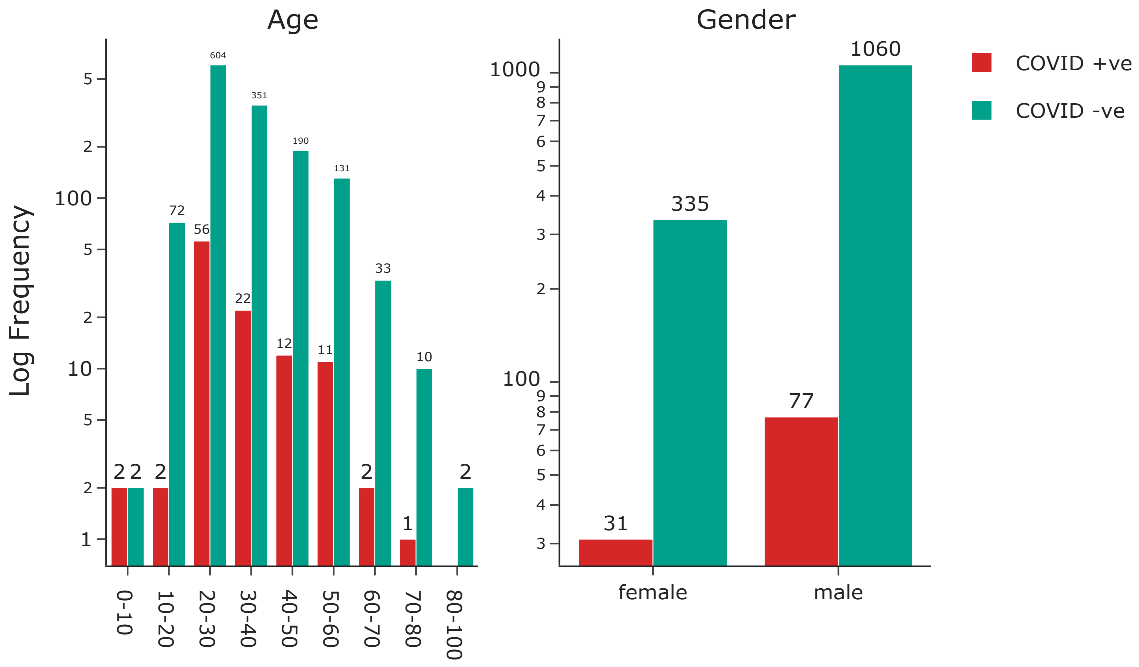


Figure 4.6: Coswara COVID-19 diagnosis across age groups and gender

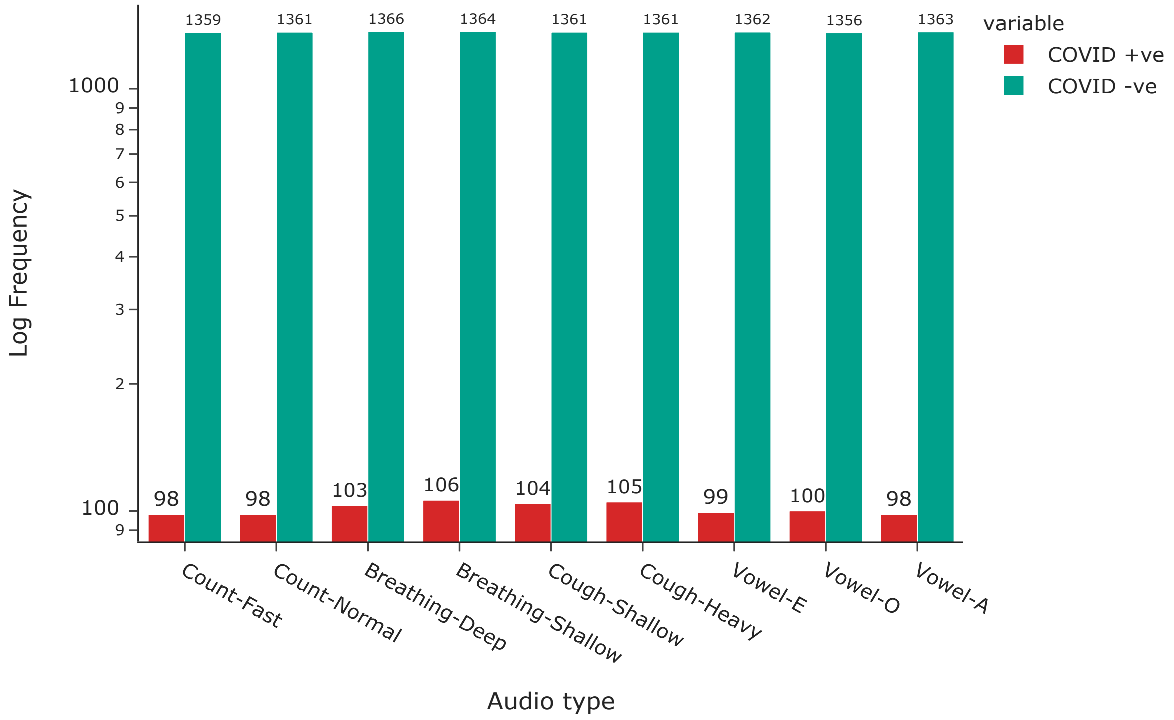


Figure 4.7: Coswara COVID-19 diagnosis for each voice variation

95 to 105 audio samples for COVID-19 positive case, and around 1350 to 1365 audio samples for the COVID-19 negative case.

Fig 4.5 gives a look and feel of the age groups and gender of the subjects who participated in this study. More specifically, fig 4.6 goes a step further to break the categories of age groups and gender down into COVID-19 positive and negative samples. Similar to the MIT dataset, coswara also highly suffers from class imbalance. In summary, both datasets have subjects in the age group of 20 to 45, which can be due to the nature of data collection through installing applications on smartphones or by visiting websites to record audio. Further, we can notice that female infection numbers is twice compared to males in MIT dataset, however, this effect is not seen in Coswara, hence it is not possible to conclusively state that females are at high risk compared to males. Table 4.3 gives an overall class split in both MIT and Coswara datasets.

Dataset	COVID +ve Samples	COVID -ve Samples	Class ratio
MIT	28	1849	66:1
Coswara	108	1395	13:1

Table 4.3: Dataset Statistics

4.3 Proposed Machine Learning Architecture

The main components of the proposed architecture are described in Fig. 1.3. In this section, we detail each component and walk through the specifics of implementation. Briefly, the flow of control is as described - Data feed(audio signals) from a Smartphone's microphone is recorded continuously with the owner's consent. Collected audio data is passed onto the audio preprocessor stage for background noise removal, extraction of MFCC/Filterbanks, and Low-level descriptors such as Zero crossing rate, Fundamental Frequency, Pitch, Shimmer, and Jitter. These extracted audio features are passed on to the anomaly detection block which outputs a decision - Anomalous or Normal. If Anomalous, notify the user of the irregularity found in voice and suggest an RT-PCR test, if not, continue to passively monitor voice in a loop. In the next section, we go over each component in this architecture and explain it in detail.

1. **Audio Preprocessor**

   Audio preprocessor receives raw audio and converts it into features that are used to model the problem. Fig 4.9 and chapter 3 illustrates this mechanism step by step. So the audio processor converts raw audio into Mel Frequency Cepstral Coeffiencients, Mel Filter Banks and Low level descriptors. We use a Spectrogram to visualize the Mel-scaled features.

   (a) Spectrograms

   Spectrograms are used to represent the strength, loudness of an audio signal visually across various frequencies present in the waveform over time. Fig. 4.8 shows the visual of a typical spectrogram. In our work, we use spectrgrams to represent Mel coefficients, more specifically, MFCCs and Mel

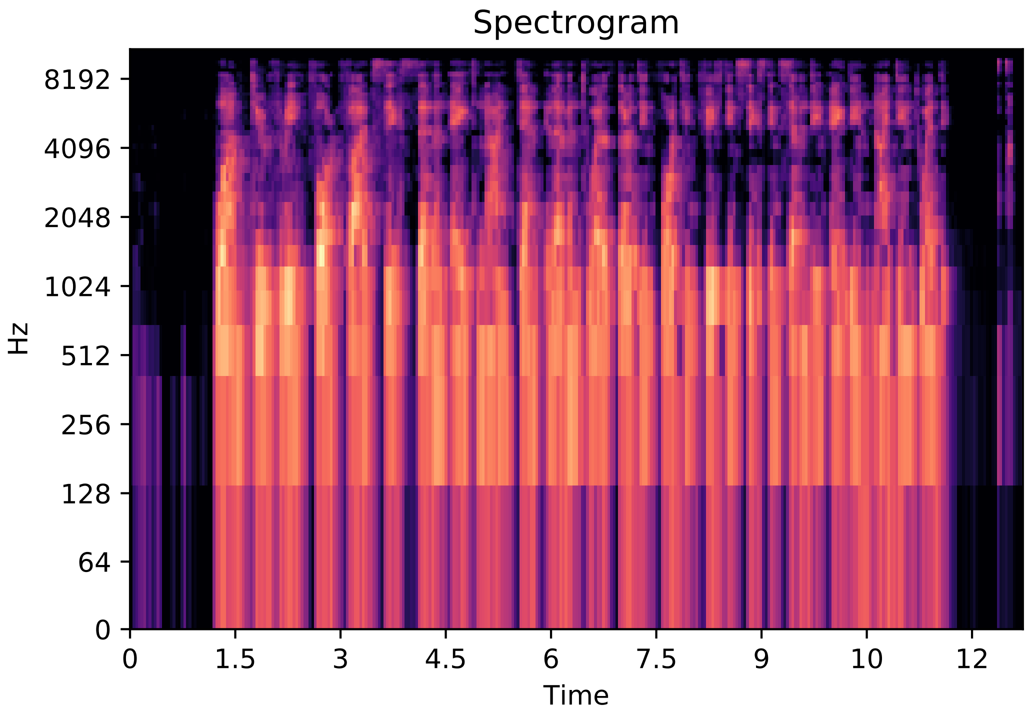


Figure 4.8: Spectrogram Representation

Filterbanks. Fig. 3.1 shows the visual difference between the two variations of spectrograms used.

(b) Low-Level Descriptors

Pitch, Fundamental Frequency, Zero-crossing rate, Shimmer, and Jitter are the set of low-level descriptors used in this work. Fig 3.2 is a 1-d representation of all the descriptors. Pitch and Fundamental Frequency

2. **Deep Anomaly Detector**

The Deep Anomaly detector comprises of a Convolutional Autoencoder for dimensionality reduction and a traditional Anomaly detector to flag anomalous data observations as shown in Fig 4.10. Below we discuss the role of each component in the architecture in more detail.

(a) Convolutional Encoder

Convolutional Neural Networks are long proven for their spatial feature extraction capabilities [56]. In this study we use a Convolutional Neural

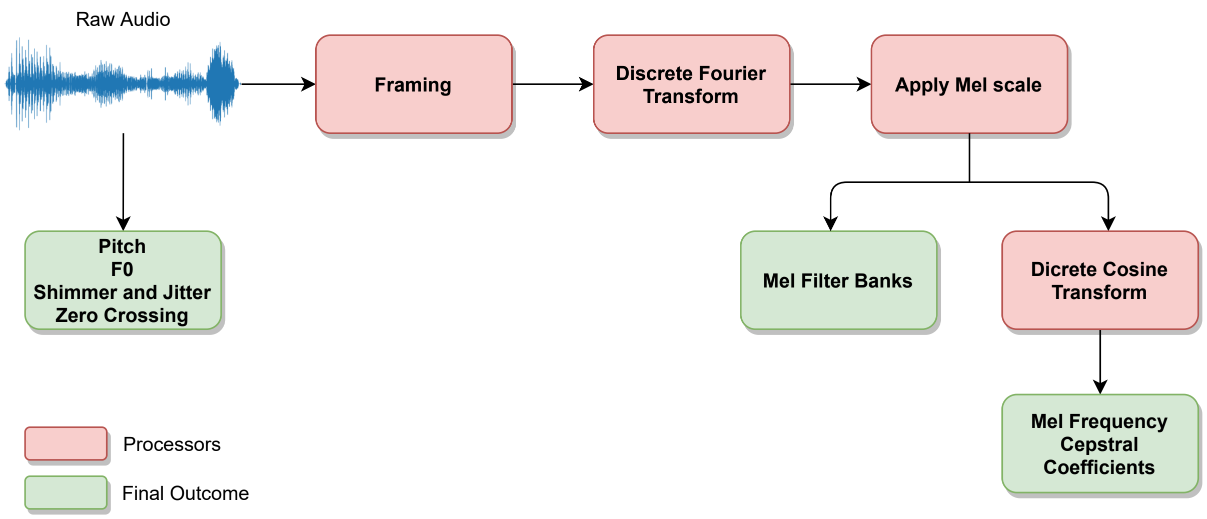


Figure 4.9: Audio Preprocessor

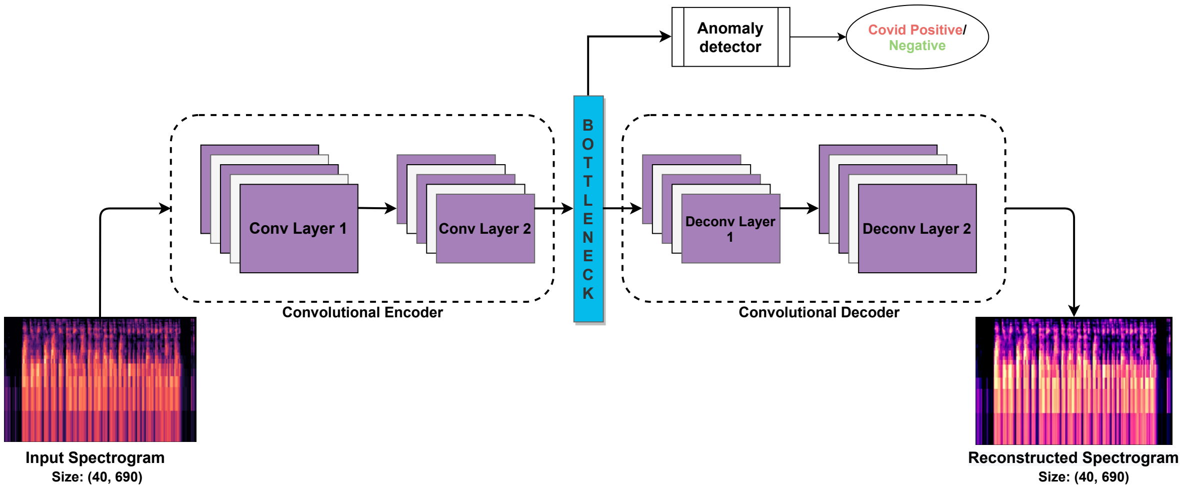


Figure 4.10: Deep Anomaly Detector

Network to be the encoder component of the Autoencoder. As described in fig 4.10 an input spectrogram of shape (40,690) is convolved over with multiple layers of convolution, batch normalization and max-pooling. The intuition behind this approach is that kernels in each convolution layer specialize in extracting specific features from the spectrogram. The output of the final convolution layer is flattened and passed on to the Convolutional Decoder for further processing.

(b) Bottleneck/ Latent space representation

The flattened output of the final layer of the Convolutional Encoder forms the Bottleneck, also referred to as Latent space representation. They are an array of values which are one-dimensional in nature. During the model training stage, the encoder ensures to tune values in bottleneck to be the most prominent set of features needed to represent the input in the lowest dimension possible. Encoder throws away redundant information and noise from the input and retains a compressed representation of the input. In our work, Bottleneck is a one-dimensional array of floating-point values representing activations from the final layer of the encoder.

(c) Convolutional Decoder

The main role of this component is to reconstruct [57] the input spectrogram back from the Bottleneck features as shown in fig 4.10. Again, we use a CNN to carry out this task. This component adds dimensions to the Bottleneck values in every layer, initially filling random values but ultimately learning the lost redundant spatial representation as the training progresses. This process is called Deconvolution or Upsampling [57]. The reconstructed images can be visualized to get first hand insights of how well the Autoencoder is trained. Further, the network performance can be measured statistically using Mean Squared Error and other evaluation metrics which we discuss in detail in chapter 5

(d) Anomaly detector

The one-dimensional output from Convolutional Encoder is used as input to the anomaly detector. In this study, we use One-Class Support Vector Machines and Elliptic Envelope to identify irregularities present in the input spectrogram. In chapter we discuss in more detail on how each component has been implemented along with all the hyperparameters used.

5 | Machine Learning to Detect COVID from Audio

5.1 Technology Stack

In programming our machine learning models, we use Python as the main language. More specifically, below are the libraries used along with their versions. Table 5.1 enumerates all the libraries used during the study.

Library	Version	Purpose
Pytorch[58]	1.5.0	Training Deep learning models
Scikit-Learn[53]	0.23.1	Leveraging Anomaly detector Implementations
Pandas[59]	1.0.5	Handling CSV data
Numpy[60]	1.19.1	Handling multidimensional arrays
Librosa[61]	0.8.0	Reading and Processing audio files
Plotly[62]	4.14.3	Plotting and Viz

Table 5.1: Libraries used in the thesis

The Linux platform utilized in this research was Red Hat Enterprise Linux Server release 7.3 (Maipo). The Deep Learning models were run on NVIDIA V100 Tensor Core GPU along with 32GB on the system memory.

In this section, we discuss the technical details of the implementation of the machine learning modules described in chapter 4, especially the Deep Anomaly detector. We also discuss the experiments conducted and present hypothesis behind each experiment. Finally, in the result section we compare how each experiment performed and provide evidence to support our conclusions. Table 5.2 lists all the data representations used in this study.

Feature Dimensions	Extracted features	Description
One Dimensional	Zero Crossing Rate	Rate at which amplitude of a signal passes through zero
	Pitch	Rate of vibrations of vocal folds
	Fundamental Frequency(f0)	Approximate frequency of the periodic structure of human voice
	Jitter	Cycle to Cycle fluctuations in Pitch
	Shimmer	Variability in the signal's peak-to-peak amplitude
Two Dimensional	MFCC	Mel scaled features followed by a discrete cosine transform to remove high correlation represented on a spectrogram
	Mel Filter Banks	Mel scaled features represented on a spectrogram

Table 5.2: Processed features and data representations

Author	Problem Type	Deep Network for feature extraction	Anomaly Detectors	Resulting Experiments
Self	Anomaly detection	Convolutional Autoencoder	OC SVM, Local Outlier Factor, Elliptic Evelope	Convolutional Anomaly Detector - 5.2
		Variational Autoencoder	OC SVM, Local Outlier Factor, Elliptic Evelope	Variational Anomaly Detector - 5.3
		Contrastive Learning	OC SVM, Local Outlier Factor, Elliptic Evelope	Contrastive Anomaly Detector - 5.4
Brown et al.	Classification	Vggish	-	-

Table 5.3: Experiments resulting from the combination of feature extractors and anomaly detectors

On the other hand, table 5.3 lists all the experiments conducted in our thesis work. We also implement Brown et al. [63] experimental methods on our dataset and compare the results against our methods. Below, we will list each experiment conducted with specific implementation details, and in the next chapter we will go over the results observed from each of these experiments.

5.2 Anomaly detection with Convolutional Autoencoders

Input spectrograms of COVID negative class are used for training the network. The intuition behind this approach is to teach the autoencoder network to learn to reconstruct from latent space representations of COVID negative class only. Therefore the latent space of a COVID positive class will not be in the same distribution of a COVID negative class, which helps us find input samples that do not statistically behave like the majority of the population. We use a 5 layer Convolutional Neural Network(CNN) for the encoder component of the Autoencoder. Each layer contains 64, 128, 256, 128, 32 kernels or activation maps respectively. The convolution operation is carried out with a kernel size of 3 and a stride window of (1,2). We have placed a max-pooling layer after every 2 layers of convolution, with a kernel size of 4 and a stride window of 1. To handle bias-variance trade-off, we have a dropout after the first pool layer. All layers have reLu activation function. Activations from each convolution layer are passed through a layer of Batch normalization [64]. Deep Networks usually contain many layers, activations of layer $n-1$ is passed as input to layer n and so on. In this process, the data distribution of activations suffer from Internal Covariate shift, meaning data distribution of the activations change from layer to layer making it hard for the neural networks to generalize well in a shorter period of time. Adding a Batch normalization layer after every convolution layer mitigates this issue resulting in better generalization and faster training times. Input specrogram of size (40,690) reduces to 1532 values called as Bottleneck. The Convolutional decoder uses the Bottleneck values to reconstruct the input spectrogram. First layer of the decoder matches the configuration of the last layer of encoder, second layer of the decoder matches second layer from last of the encoder and so on. This ensures that each layer in the decoder learns the representation that was thrown away by its corresponding encoder layer. This operation continues until an image of size of input spectrogram is obtained. Mean squared error between the reconstructed image and the input image is calculated. Back propagation calculates gradient updates for parameters of each layer to minimize the loss function. The network is trained for 300 epochs, learning rate of 0.0001. We exponentially decay the learning rate as shown in eq 1 to adapt to the gradient curve and smoothen the training.

$$lr_n = lr_{n-1} * e^{-\gamma \, * \, t} \tag{1}$$

Here γ is the decay factor i.e. amount by which the value of learning rate has to be decayed. t is the timestep(increments after every gradient update).

We use Stochastic Gradient Descent[49] and Back Propagation[50] for minimizing the loss of the Autoencoder. Eq 2 describes the components of the loss function. We use Mean square error to measure the error in reconstruction and an L1 penalty term to further make the Autoencoder sparse, resulting in turning unnecessary weight parameters to zero. This, along with dropout helps the the network to generalize well on real-world data.

$$\underbrace{\left[\frac{1}{n}\sum_{i=1}^{n}[\hat{x}_i - x_i]^2\right]}_{\text{MSE Reconstruction Loss}} + \lambda . \underbrace{\left[\sum_{j}|w_j|\right]}_{\text{L1 penalty}} \tag{2}$$

x_i and $\hat{x}_i$ represent the spectrogram and the reconstructed spectrogram for audio input i respectively. $\mid w_j \mid$ is the absolute value of weights for layer j. λ is a regularization hyper parameter used to control extent of generalization. Each hidden layer, in both the encoder and decoder is followed by reLu, a non-linear activation function as described in Eq 3 to consistently introduce non-linearity in the entire network.

$$\hat{y_i^h} = max\left[0, ((w^j * x_i^j) + b^j)\right] \tag{3}$$

$\hat{y_i^j}$ is the reLu activation output for hidden layer j and input sample i. w^j and b^j are the layer weights and bias for layer j. x_i^j is the input i for layer j

Once the network is trained, we use the encoder component of the network alone to generate latent space representations of the input spectrogram. We use the la-

	Convolutional Anomaly Detector	Variational Anomaly Detector	Constrastive Anomaly Detector
Encoder CNN layers	5 layers (64, 128, 256, 128, 32 kernels respectively)		
Decoder CNN layers	5 layers (32, 128, 256, 128, 64 kernels respectively)		
Optimizer	Adam		
Learning rate	0.0001		
Learning rate decay	Exponential decay		
Epochs	300		
Dropout	0.2 dropout after first layer		
Loss function	Mean squared Error + L1 Penalty	Mean squared Error + KL divergence	Contrastive Loss function

Table 5.4: Summary of model hyperparameters

tent space directly to train a traditional anomaly detector. One Class Support vector machines(OC-SVM), Local Outlier Factor and Elliptic Envelope are used as anomaly detectors. We use implementations available Scikit-Learn library for all anomaly detectors. We experiment with rbf and $poly$ kernels for One Class SVMs, and retain default parameters in the other two detectors. All models are subject to 5-fold cross validation. In the results section we show the performance from each experiment. Table 5.4 list out all the final parameters used across all models in the study.

5.3 Anomaly detection with Variational Autoencoders

We trained a variational autoencoder to extract normally distributed bottleneck features. Here again, only COVID negative spectrograms are used during training, with the intuition that COVID positive samples will eventually form a distribution with mean and variance deviating from being normal. We use the same network architecture used in Convolutional Autoencoders for training the encoders and decoders. However, instead of generating one 1-dimensional array of bottleneck features, we generate two 1-dimensional vectors for mean and variance. We then sample one 1-dimensional vector from the two vectors, which ultimately forms the normally distributed bottleneck features. Along with the reconstruction loss, we use a divergence function that measures the deviation of bottleneck from a randomly sampled vector of same size with zero mean and unit variance. In particular, we calculate the Kullback–Leibler divergence to ensure normality in the Bottleneck features. The complete loss function of a variational Autoencoder is defined in Eq 4

$$\underbrace{\left[\frac{1}{n}\sum_{i=1}^{n}[\hat{x}_i - x_i]^2\right]}_{\text{MSE Reconstruction Loss}} + \underbrace{[KL\ (q_\theta(z \mid x_i)\ \parallel\ p(z))]}_{\text{KL Divergence}} \tag{4}$$

First term is the mean squared error between input sample x_i and the reconstructed sample $\hat{x}_i$. Second term is the KL divergence with $q_\theta(z \mid x_i)$ being the encoder's representation of the latent space which is compared to $p(z)$, a randomly sampled Gaussian data with zero mean and unit variance

5.4 Anomaly detection using Contrastive Learning

Objectives of Loss functions like Mean squared loss or Cross entropy loss is to learn to predict a value for a given input. In this study we explore a loss function called Contrastive Loss whose objective is to predict relative distances between two data points, in our case between input spectrogram and reconstructed spectrogram. Interestingly, it takes class label into consideration to model the loss function. Specifically, it takes input data, reconstructed data and the class label. The reason behind using the class label is to force the encoder to model similar Bottleneck features for same class and distant representations for different classes. In our case, contrastive loss essentially helps the encoder model two statistically different Bottleneck representations for the two classes in the dataset i.e. COVID and non-COVID. This helps Anomaly detectors perform better since it creates a nice gap in distribution of Bottleneck features of both classes. Constrative loss is defined in eq 5

$$L(x_i, \hat{x}_i, y) = y* \parallel x_i - \hat{x}_i \parallel + (1-y) * \max(0, m - \parallel x_i - \hat{x}_i \parallel) \tag{5}$$

where $\parallel x_i - \hat{x}_i \parallel$ is the Euclidean distance between input spectrogram and reconstructed spectrogram. m is the some margin value that enforces larger values in loss, in case of negative samples. This is the parameter that generate discrepancy in the loss value based on class. y is the binary class label. Bottomline, network trains on two different loss functions for two classes.

6 | Results and Evaluation

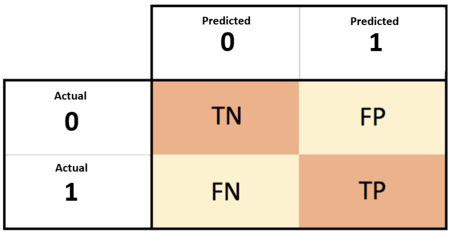
In this section we compare all the experiments described in chapter 5 and provide supporting graphs and metrics.

6.1 Evaluation Metrics

An operation or an algorithm can be used only if it optimally performs the designated task. So measuring the performance of operation becomes crucial to carry out tasks successfully. For example, performance of a sorting algorithm can be evaluated by measuring its run-time and space consumed. This metric can be used to compare multiple sorting algorithms. Likewise, in machine learning, performance of predictive models are measured using standard techniques discussed below. Each of the below listed technique checks if model is predicting as expected. Although the end goal of these techniques are same, they offer unique insights to evaluate model performance.

1. Confusion Matrix

   In case of binary classification, where the expected outcomes of machine learning



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Figure 6.1: Confusion Matrix

algorithm will be 0 or 1. However, in most classification algorithms, the output will be a logistic function, which is a continuous between zero to one. These continuous values are then categorized as zero or one based on a threshold value. Confusion matrix is a 2 by 2 matrix as described in fig 6.1. True Negatives(TN) are instances where the predicted label and true label matches, while the class label is zero. On the other hand, True positives(TP) are same as True Negatives while the class label is one. False Positive(FP) are instances where predicted label is one but the true label is zero, and False Negatives(FN) are instances where predicted label is zero and the true label is one. Confusion matrix is used to derive important evaluation metrics discussed below.

2. Precision

   Precision is the ratio of correct predictions of positive class to all positive predictions. It is defined by eq 1. For example, the metric measures how many of the people labelled diabetic are actually diabetic in real world?

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

3. Recall

   Recall is the ratio of correct predictions of positive class to all positive class labels. It is defined by eq 2. For example, the metric measures - Of all the people who are diabetic, how many how correctly classified as diabetic? This metric is highly used in mission critical applications such as medical systems where missing a positive observation can be fatal

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

4. F1 score

   The harmonic mean of Precision and Recall is F1 score. It is defined by eq 3. This metric helps evaluate models which are highly skewed due to class imbalance in training data.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

5. F_β score

   F_β is a variant of F1 score. It differs by using a β parameter to control the importance given to Precision and Recall while calculating the F value. β values above one give more weight to recall and less weight to precision. β values lesser than one more weight to precision and less weight to recall.

$$F_\beta = (1 + \beta^2) \frac{Precision * Recall}{(\beta^2 * Precision) + Recall} \tag{4}$$

6. AUC score

   It is the are covered under a ROC curve. ROC is abbreviated as Receiver Operating Characteristic curve. This curve measures the performance of a classification models at all possible thresholds. More specifically, it is a plot of True positive rate vs False positive rate.

6.2 Experiment-wise results

We replicate Brown et al [63] methodologies i.e. data preprocessing and model components and measure their performance on our data. The best performing model has Mel Filterbanks representation on cough and breath data with 5-fold cross validated Support Vector Machines with an rbf kernel. We observe an AUC of 62.5. We use this as a baseline to compare against our Deep Anomaly detectors. On these lines, Variational Anomaly detector with Elliptic Envelope using Mel Filterbank representation on cough data performs the best across all the experiments with an AUC of 65.7. Below we discuss the our explorations with each deep anomaly detector while listing all the combinations of data representations and variations(cough, breath etc) used.

6.2.1 Anomaly detection with Convolutional Autoencoders

A Convolutional Autoencoder is trained to reconstruct a spectrogram. Once the model is trained, we use it to generate Bottleneck features and feed it to three different Anomaly detectors, namely One Class SVM, Local Outlier Factor and Elliptic Envelope that takes in the Bottleneck features as input. This entire process is repeated for two different type of data representations - MFCC and Mel Filter Banks over four different data varaitions - Cough, Breath, Cough Breath, Alphabets. Table 6.1 shows results obtained. Fig 6.2 shows loss trajectory of autoencoder. Although the loss has reduced drastically, we can see that model performance is poor overall, irrespective of data representation and anomaly detector. Based on the results we notice that cough breath together in Mel Filterbank representation work well with an AUC of 57.5, as suggested in Brown et al [63]. We observe that bottleneck representations formed by the network are inconclusive to decide if an audio sample is to be considered an anomaly. Further, detailed analysis on this is carried out in chapter 7

6.2.2 Anomaly detection with Variational Autoencoder

As we saw in the experiments with Convolutional Autoencoders, the results were poor even though the reconstruction loss was very less. One possibility might be the fact that Bottleneck features for both the class did not have a clear differentiation, although the autoencorder was trained only on negative class. To exploit this assumption we use a Variational Autoencoder and train it on negative class observations. Intuition behind this approach is that the autoencoder will force the Bottleneck features of negative class to be normally distributed but will not have the same constraint when generating Bottleneck features for positive class, thereby creating a clear differentiation. Table 6.2 compares performance of VAEs across two data representations and two anomaly detectors. We can see a best AUC score of 65.7 for cough and breath data using Mel Filterbank representation and Elliptic Envelope, which is a significant rise from 57.5 in the previous best Convolutional Autoencoder experiment. Fig 6.3 and fig

6.4 shows the trajectory of Reconstruction Loss and KL Divergence. This experiment confirms that forcing the network to generate bottleneck features forming a specific distribution, in particular a normal distribution helps identify anomalies better than before. In the next experiment we will see the effects of forcing data from both class to form different, but specific distributions and see if it aids in better anomaly detection. In chapter 7 we support our claims by visualizing spectrogram reconstructions and bottleneck features.

6.2.3 Anomaly Detection using Contrastive Learning

As opposed to training a network to form Gaussian Bottleneck features for negative class spectrograms, in Constrastive Learning the goal learn two statistically different bottleneck representations by making use of data from both the classes and supply the class labels to network's loss function. This is better compared to Variational Autoencoder where we do not learn any specific representation for positive class data since the model only gets trained on negative classes. Table 6.3 shows the results of optimizing Convolutional Autoencoder with Contrastive Loss. Based on the outcomes of previous two experiments with data representations, we stick to only performing our experiments on Mel filterbanks as it consistantly outperforms MFCC representations. Best AUC of 63.4 comes from using both cough and breath data, and training it with Convolutional Autoencoder and One class SVM, however, Variational Autoencoder with ELliptic Envelope still outperforms this result. Comparing results from table 6.1 we can see significant rise in all the metrics only by replacing Mean squared loss with Contrastive loss. Fig 6.2 shows the training loss trajectory. This clearly shows using Constrative learning instead of MSE helps network learn to create distinct bottleneck representations on a class level. We do not experiment with Variational Autoencoders since replacing MSE loss with Contrastive loss would train the network for 2 different goals. Contrastive loss tries to build two different Bottleneck feature distributions where as a Variational Autoencoder tries to model a Gaussian Bottleneck for both classes. Further, detailed analysis on this is carried out in chapter 7

Convolutional Auto-Encoder	Anomaly Detectors					
	MFCC			Mel Filter Banks		
	OC SVM	Local Outlier Factor	Elliptic Envelope	OC SVM	Local Outlier Factor	Elliptic Envelope
Cough						
AUC	51	52.5	50.5	52.5	**55**	52
F1 score	17.5	32.5	21.9	37	**38**	35.2
F2 score	21.9	34.6	22.5	37.5	**39.3**	36.8
Breath						
AUC	50.5	52.2	51.2	52.3	**54.2**	51
F1 score	18.2	31.1	22	37	**37.5**	34.9
F2 score	21.2	34.3	22.9	37.5	**38.8**	35.9
Cough & Breath						
AUC	53.4	54.4	53.1	55.8	**57.5**	53.1
F1 score	19.2	34.1	24.2	39.8	**41.4**	36.7
F2 score	23.3	36.5	25.1	41.1	**42.9**	38
Alphabets						
AUC	50.5	51.1	50.8	51.1	**53.2**	51.9
F1 score	16.8	32	21.2	36.7	**37.1**	34.8
F2 score	21	34.2	22	37	**37.9**	35.9

Table 6.1: Convolutional Autoencoders with different Anomaly detectors

Variational Auto-Encoder	Anomaly Detectors					
	MFCC			Mel Filter Banks		
	OC SVM	Local Outlier Factor	Elliptic Envelope	OC SVM	Local Outlier Factor	Elliptic Envelope
Cough						
AUC	56	55	58	56.5	55	**63.09**
F1 score	42.3	39.9	47.6	46.8	40.6	**53.2**
F2 score	43.9	40.3	48.1	47.3	41.5	**55.9**
Breath						
AUC	56.6	55.9	58.3	56.9	55.8	**63.8**
F1 score	42.6	41	48.1	48	41	**53.9**
F2 score	44.1	41.6	50.9	49.2	42.2	**55.1**
Cough & Breath						
AUC	58.1	57.7	60.9	58.4	58	**65.7**
F1 score	44.9	43	49.2	48.9	41.4	**55.6**
F2 score	45.6	44.3	51	49.8	42.6	**57**
Alphabets						
AUC	55.2	54.1	56.3	55.8	55.2	**62.2**
F1 score	41.1	38.7	46.3	42.4	39.9	**52.1**
F2 score	42.3	39.8	47.9	43.1	41	**54.9**

Table 6.2: Variational Autoencoders with different Anomaly detectors

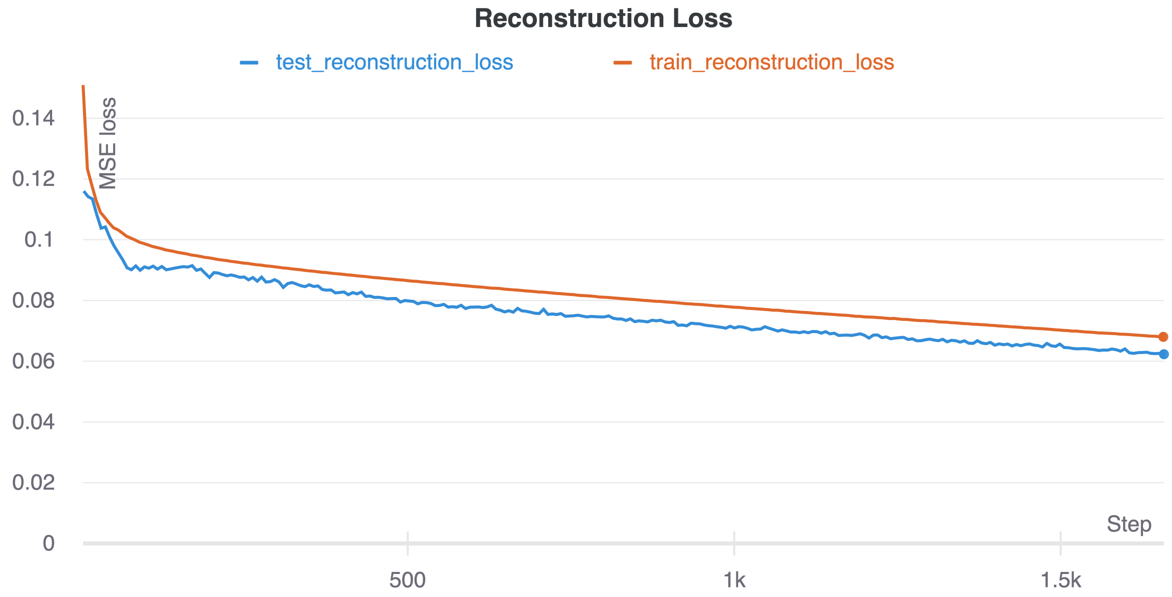


Figure 6.2: Convolutional Autoencoder reconstruction loss

6.3 Results Summary

Table 6.4 contains over all the experiments on best performing data variation which is cough and breath data. We observe that Variational Anomaly detection using EllipticEnvelope performs best across all experiments conducted with an AUC of 65.7. We also observe that replacing MSE loss with Contrastive learning methods improves the bottleneck encoding capabilities of Convolutional Autoencoders, the AUC rises to 63.4 from 57.5. On the other hand, Mel filters turn out to be the best data representation for the task of COVID detection. As initially described, along with the Autoencoder experiments we also replicated the experiment from Brown et al. [63] on MIT and Coswara datasets to compare performance of our proposed deep anomaly detectors against their method. We follow the data processing and modelling approach as described in the paper and train the models on MIT and Coswara datasets combined and recieve an AUC of 61.2 which is outperformed by our Variational Anomaly detection experiment. However, using their dataset, they claim an AUC of 82.

Convolutional Autoencoder + Contrastive Loss	Anomaly Detectors		
	OC SVM	Local Outlier Factor	Elliptic Envelope
Cough			
AUC	**61**	60.03	58
F1 score	**50.54**	49.8	46.2
F2 score	**52.56**	51.12	48.4
Breath			
AUC	**61**	60.1	57.5
F1 score	**51**	50.2	45.9
F2 score	**52.9**	52.1	47.4
Cough & Breath			
AUC	**63.4**	63.6	59.8
F1 score	**53.3**	52.9	47.8
F2 score	**55.1**	54.5	49.3
Alphabets			
AUC	**59**	58.8	56.3
F1 score	**50.2**	49.1	45.9
F2 score	**51.9**	51.8	48

Table 6.3: Convolutional Autoencoders with Contrastive Learning

Deep Feature Extractor	Metrics	Anomaly Detectors						Binary Classification
		MFCC			Mel Filter Banks			-
		OC SVM	Local Outlier Factor	Elliptic Envelope	OC SVM	Local Outlier Factor	Elliptic Envelope	-
Convolutional Autoencoders	AUC	53.4	54.4	53.1	55.8	**57.5**	53.1	-
	F1 score	19.2	34.1	24.2	39.8	**41.4**	36.7	-
	F2 score	23.3	36.5	25.1	41.1	**42.9**	38	-
Variational Autoencoders	AUC	58.1	57.7	60.9	58.4	58	**65.7**	-
	F1 score	44.9	43	49.2	48.9	41.4	**55.6**	-
	F2 score	45.6	44.3	51	49.8	42.6	**57**	-
Convolutional Autoencoders + Contrastive Loss	AUC	-	-	-	**63.4**	63.6	59.8	-
	F1 score	-	-	-	**53.3**	52.9	47.8	-
	F2 score	-	-	-	**55.1**	54.5	49.3	-
Brown et al.	AUC	-	-	-	-	-	-	61.2
	F1 score	-	-	-	-	-	-	51.5
	F2 score	-	-	-	-	-	-	50.4

Table 6.4: Summary of all experiments conducted

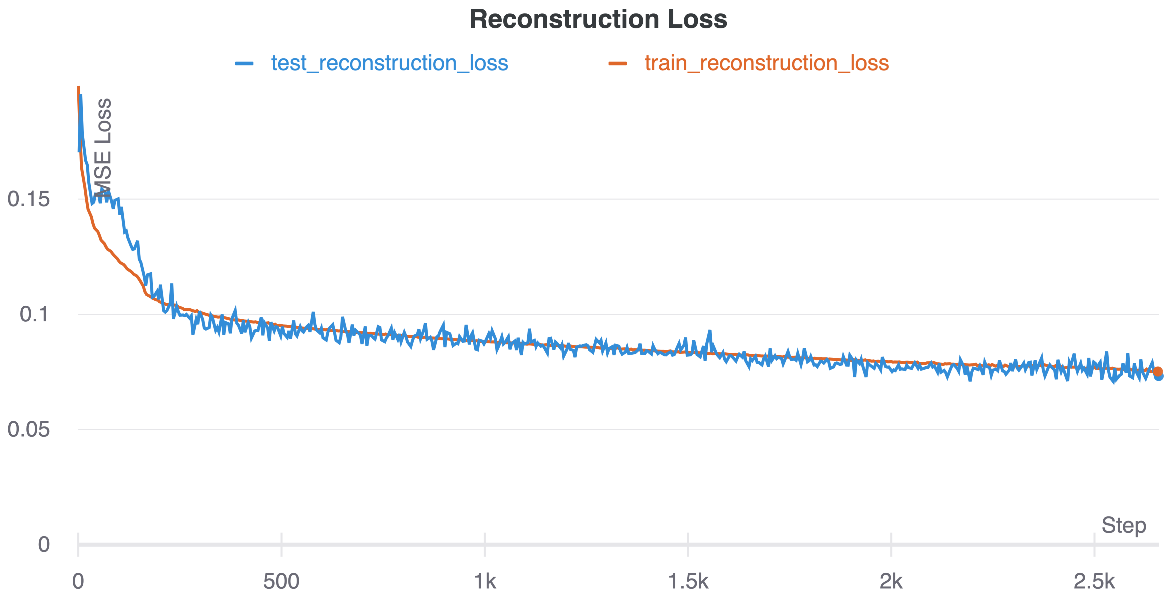


Figure 6.3: Variational Autoencoder reconstruction loss

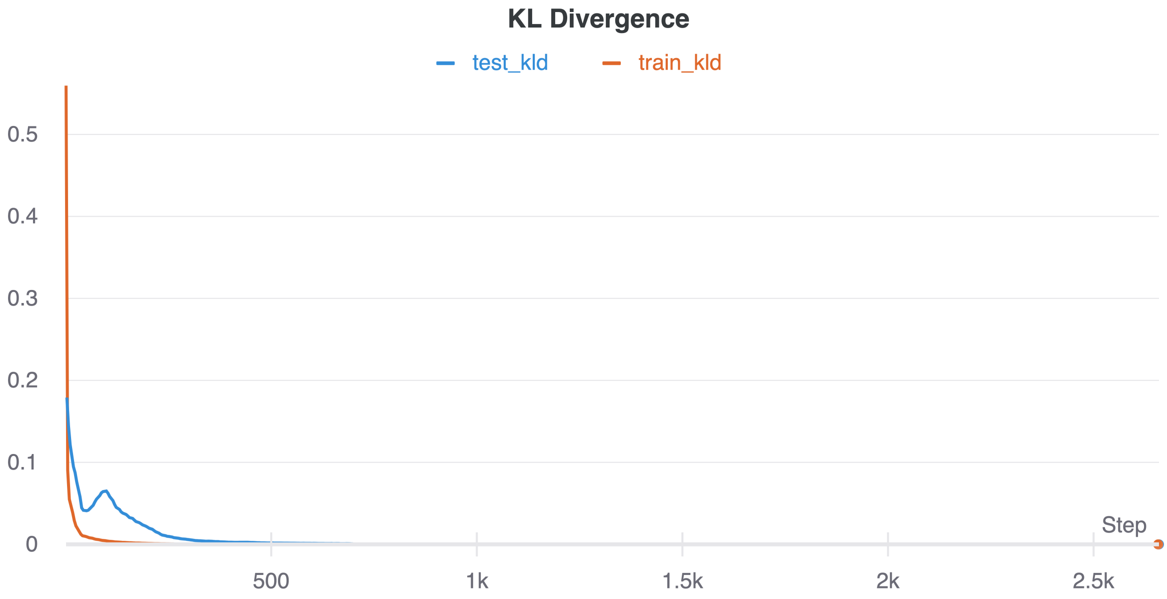


Figure 6.4: KL Divergence of Variational Autoencoder

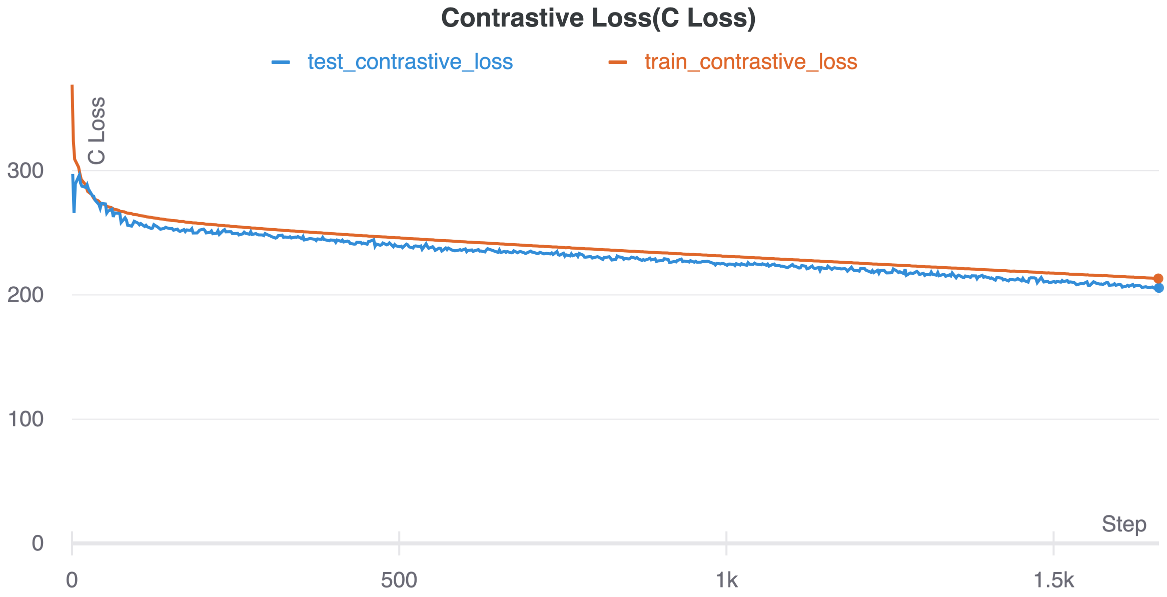


Figure 6.5: Contrastive Loss

7 | Discussion

Based on the performance of all the experiments we ran, and the results we displayed on the previous section, the Variational Autoencoders with Elliptic Envelope is our best performing model with an AUC of 65.7.

In this section we go over some observations we made from our experiments. Specifically, we list out experiments that worked as per our intuition, experiments that we feel is not the right fit for the task and conclude the chapter with limitations noted during the course of this study.

Bottleneck features from Variational Autoencoders are good representations of audio data to detect anomalies. Further fig 7.4 confirms our claim. It is a 2-d t-SNE plot of the bottleneck features from Variational Autoencoders. We can see that COVID positive samples being clustered in the lower left corner of the chart. Intuitively, this means although the decoder parameters have generalized reconstructing from both classes, there is a statistical difference in bottleneck features. The decoder also does a good job at reconstructing spectrograms from both classes 7.3. On the other hand, we observe that using Contrastive learning methods instead of MSE loss for training Convolutional Autoencoders helps creating distinct class wise representations of the bottleneck features. It does two things as expected. First, it does a good job in reconstructing images from both class, fig 7.5. Second, it creates two statistically different underlying bottleneck features which can be observed in fig 7.6.

Further, we notice that Convolutional Autoencoders fail to create bottleneck representations that aid anomaly detectors. We can observe that in its corresponding t-SNE plot 7.2. The observations of both classes are completely spread across the 2-d plane making it harder to draw a decision boundary. However, we can see from fig 7.1 that the network does a good job a reconstruction.

From the experiments run during the study, we made some collective observations that might potentially be the limitations faced by the techniques employed.

1. Every experiment reconstructs the input spectrogram very well. Alternatively, this means there is very little difference in the input images across both classes. This can be a point of ambiguity in model training.
2. Out of the 150 million cases reported, our results are based on patterns found in only 150 cases, out of which only 120 cases were used for training. Having access to more COVID positive audio samples will help improve model performance

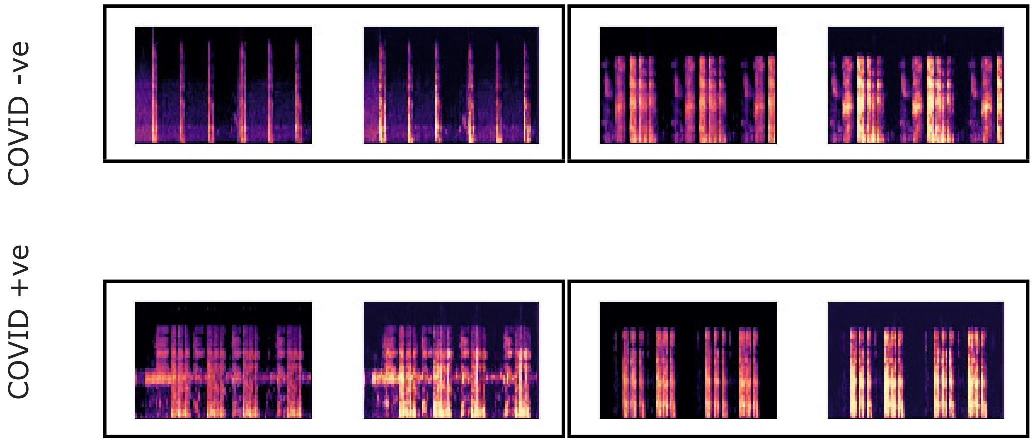


Figure 7.1: Convolutional Autoencoder Reconstructions(Left: Input | Right: Reconstructed)

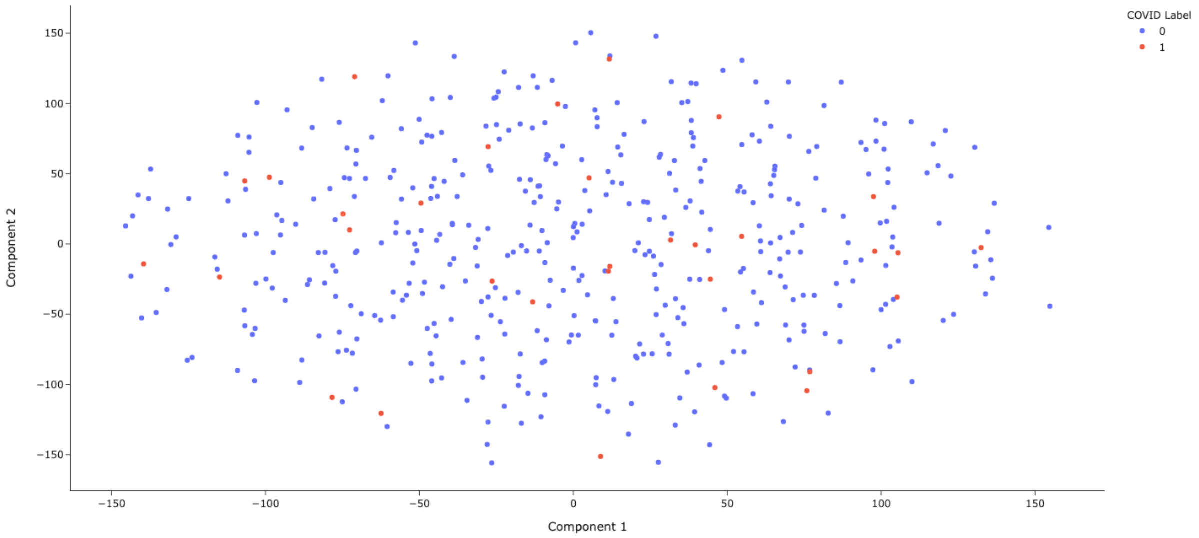


Figure 7.2: t-SNE viz of Bottleneck features from Convolutional Autoencoders

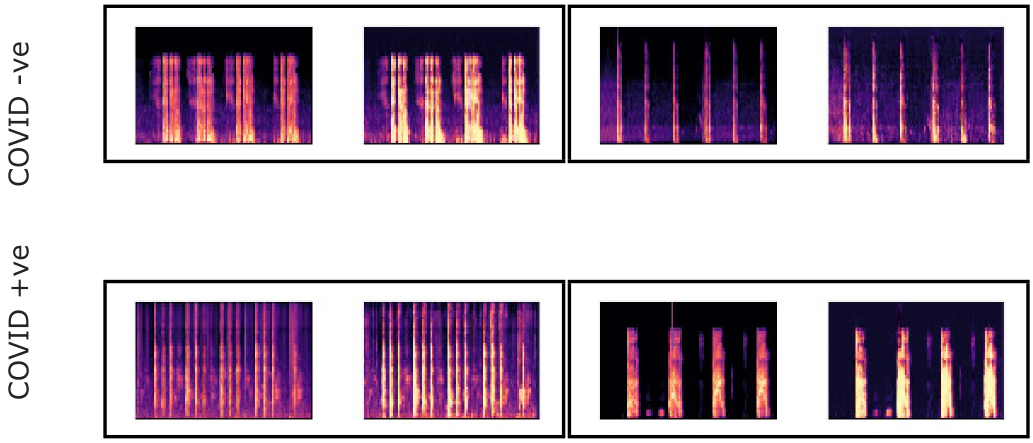


Figure 7.3: Variational Autoencoder Reconstructions(Left: Input | Right: Reconstructed)

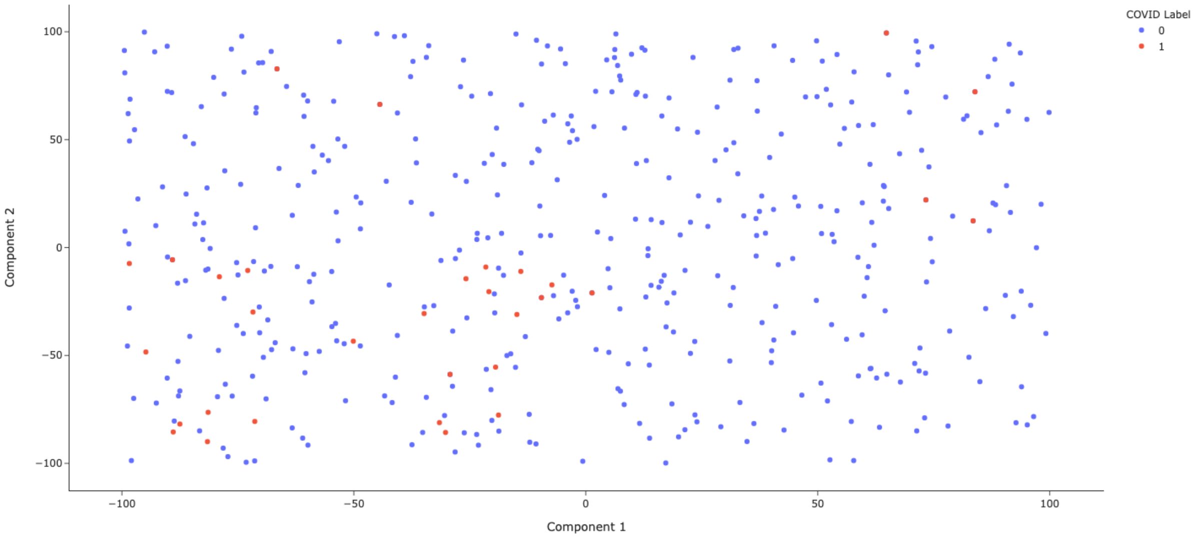


Figure 7.4: t-SNE viz of Bottleneck features from Variational Autoencoders

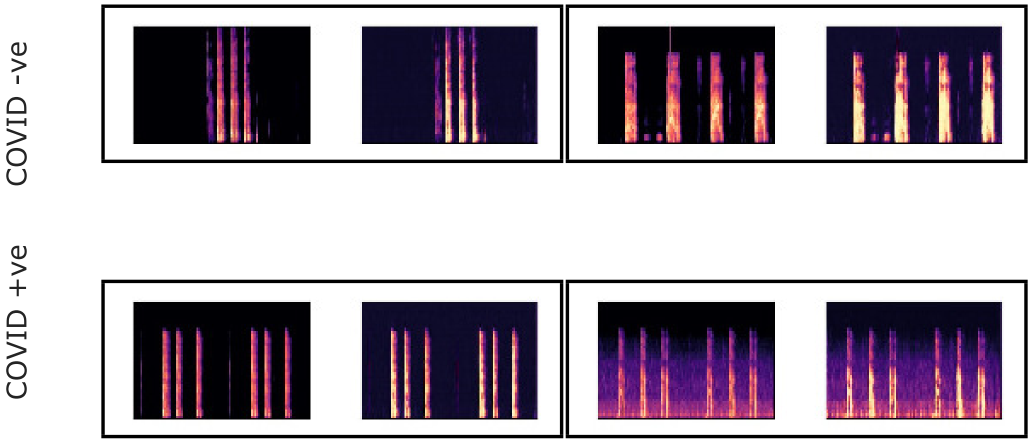


Figure 7.5: Contrastive Learning Reconstructions(Left: Input | Right: Reconstructed)

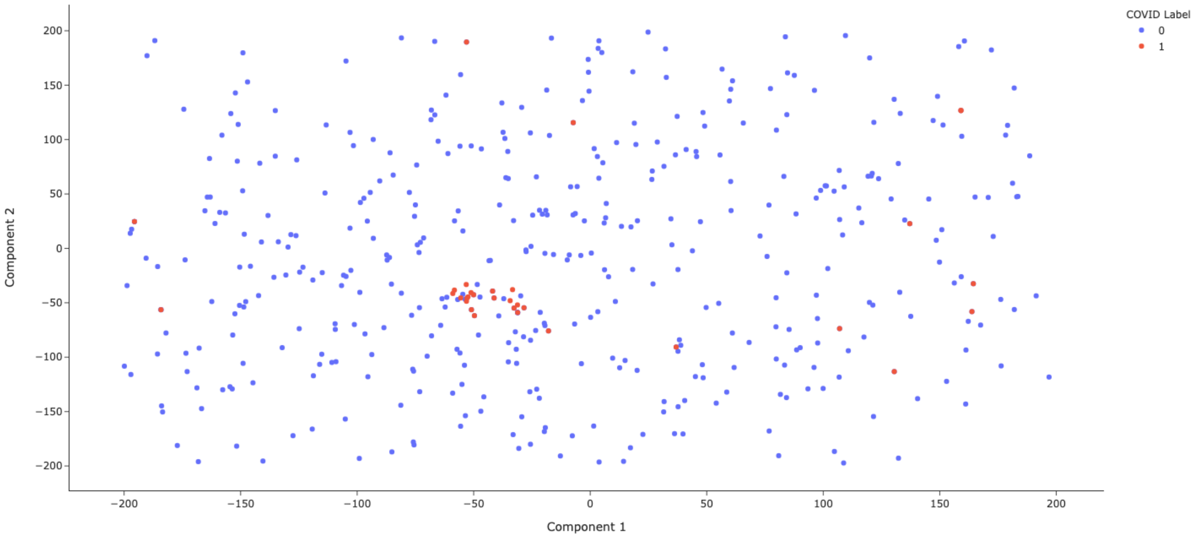


Figure 7.6: t-SNE viz of Bottleneck features from Constrative Learning

8 | Conclusions and Future Work

8.1 Conclusion

Passive assessment of subjects from smartphone-sensed audio can be useful in mitigating the spread of the disease. In this thesis, we explored passive identification of COVID-related audio symptoms such as coughing and breathing patterns from smartphone-captured audio using two publicly available datasets [35] [20]. Various representations of cough and breath samples were explored including MFCCs and Mel Filterbanks. We explored various anomaly detectors and autoencoders to detect COVID-positive subjects including convolutional Autoencoders, variational autoencoders, and contrastive learning methods whose bottleneck features were fed to traditional anomaly detectors. The variational autoencoder with the elliptic envelope as the anomaly detector analyzing Mel Filterbanks audio representations performed best with an AUC of 65.7. We found that training the autoencoder forcefully to learn different representations for COVID positive and negative samples worked very well as voice samples for both cases exhibited similar patterns (discussed in chapter 7).

8.2 Future Work

1. Gather more COVID audio data especially positive cases, which would yield a more balanced dataset, making it possible to explore state-of-the art audio classification models [65, 66].

2. Add more non-COVID cough samples [67] to make cough detection more robust against incorrectly classifying non COVID coughs as covid coughs
3. Compare our best performing approaches with more state-of-the-art approaches in audio COVID detection.

Bibliography

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019, 2019.

[2] Gauri Deshpande and Björn Schuller. An Overview on Audio, Signal, Speech, and Language Processing for COVID-19. *arXiv e-prints*, art. arXiv:2005.08579, May 2020.

[3] Stoll R. Haghi M, Thurow K. Wearable devices in medical internet of things: Scientific research and commercially available devices. pages 4–15, 2017. doi: 10.4258/hir.2017.23.1.4.

[4] Blaine Reeder and Alexandria David. Health at hand: A systematic review of smart watch uses for health and wellness. *Journal of Biomedical Informatics*, 63:269–276, 2016. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2016.09.001. URL `https://www.sciencedirect.com/science/article/pii/S1532046416301137`.

[5] Sumit Majumder and M. Jamal Deen. Smartphone sensors for health monitoring and diagnosis. *Sensors*, 19(9), 2019. ISSN 1424-8220. doi: 10.3390/s19092164. URL `https://www.mdpi.com/1424-8220/19/9/2164`.

[6] S. N. Murthy, F. Asani, S. Srikanthan, and E. Agu. Deepseas: Smartphone-based early ailment sensing using coupled lstm autoencoders. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4911–4918, 2020. doi: 10.1109/BigData50022.2020.9377885.

[7] W. Gerych, E. Agu, and E. Rundensteiner. Classifying depression in imbalanced datasets using an autoencoder- based anomaly detection approach. In *2019 IEEE*

13th International Conference on Semantic Computing (ICSC), pages 124–127, 2019. doi: 10.1109/ICOSC.2019.8665535.

[8] Yasin Özkanca, Miraç Öztürk, Merve Ekmekci, David Atkins, Cenk Demiroglu, and Reza Hosseini Ghomi. Depression screening from voice samples of patients affected by parkinson's disease. *Digital Biomarkers*, 3:72–82, 06 2019. doi: 10.1159/000500354.

[9] Kim Berninger, Jannis Hoppe, and Benjamin Milde. Classification of speaker intoxication using a bidirectional recurrent neural network. volume 9924, pages 435–442, 09 2016. ISBN 978-3-319-45509-9. doi: 10.1007/978-3-319-45510-5_50.

[10] Björn Schuller, Stefan Steidl, Anton Batliner, Florian Schiel, Jarek Krajewski, Felix Weninger, and Florian Eyben. Medium-term speaker states—a review on intoxication, sleepiness and the first challenge. *Computer Speech and Language*, 28(2):346–374, 2014. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2012.12.002. URL https://www.sciencedirect.com/science/article/pii/S0885230812001027.

[11] 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*, 1:203–206, 2020. doi: 10.1109/OJEMB.2020.2998051.

[12] Björn W. Schuller, Dagmar M. Schuller, Kun Qian, Juan Liu, Huaiyuan Zheng, and Xiao Li. COVID-19 and Computer Audition: An Overview on What Speech and Sound Analysis Could Contribute in the SARS-CoV-2 Corona Crisis. *arXiv e-prints*, art. arXiv:2003.11117, March 2020.

[13] Ali Imran, Iryna Posokhova, Haneya N. Qureshi, Usama Masood, Muhammad Sajid Riaz, Kamran Ali, Charles N. John, MD Iftikhar Hussain, and Muhammad Nabeel. Ai4covid-19: Ai enabled preliminary diagnosis for covid-19 from cough samples via an app. *Informatics in Medicine Unlocked*, 20:100378, 2020.

ISSN 2352-9148. doi: https://doi.org/10.1016/j.imu.2020.100378. URL https://www.sciencedirect.com/science/article/pii/S2352914820303026.

[14] Gauri Deshpande and Björn Schuller. An Overview on Audio, Signal, Speech, and Language Processing for COVID-19. *arXiv e-prints*, art. arXiv:2005.08579, May 2020.

[15] Piyush Bagad, Aman Dalmia, Jigar Doshi, Arsha Nagrani, Parag Bhamare, Amrita Mahale, Saurabh Rane, Neeraj Agarwal, and Rahul Panicker. Cough Against COVID: Evidence of COVID-19 Signature in Cough Sounds. *arXiv e-prints*, art. arXiv:2009.08790, September 2020.

[16] *MIT Covid Audio Dataset*. URL https://opensigma.mit.edu/.

[17] William Thorpe, Miranda Kurver, Gregory King, and Cheryl Salome. Acoustic analysis of cough. pages 391 – 394, 12 2001. ISBN 1-74052-061-0. doi: 10.1109/ANZIIS.2001.974110.

[18] Hanieh Chatrzarrin, Amaya Arcelus, Rafik Goubran, and Frank Knoefel. Feature extraction for the differentiation of dry and wet cough sounds. In *2011 IEEE International Symposium on Medical Measurements and Applications*, pages 162–166, 2011. doi: 10.1109/MeMeA.2011.5966670.

[19] C. Infante, D. Chamberlain, R. Fletcher, Y. Thorat, and R. Kodgule. Use of cough sounds for diagnosis and screening of pulmonary disease. In *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 1–10, 2017. doi: 10.1109/GHTC.2017.8239338.

[20] *Indian Institute of Science, Bangalore, .* URL https://github.com/iiscleap/Coswara-Data.

[21] Shikha Gupta, Jafreezal Jaafar, Wan Fatimah Wan Ahmad, and Arpit Bansal. Feature extraction using mfcc. *Signal*

& Image Processing : An International Journal, 4:101–108, 08 2013. doi: 10.5121/sipij.2013.4408.

[22] S. K. Kopparapu and M. Laxminarayana. Choice of mel filter bank in computing mfcc of a resampled speech. In *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, pages 121–124, 2010. doi: 10.1109/ISSPA.2010.5605491.

[23] Haydée F. Wertzner, Solange Schreiber, and Luciana Amaro. Analysis of fundamental frequency, jitter, shimmer and vocal intensity in children with phonological disorders. *Brazilian Journal of Otorhinolaryngology*, 71(5):582–588, 2005. ISSN 1808-8694. doi: https://doi.org/10.1016/S1808-8694(15)31261-1. URL https://www.sciencedirect.com/science/article/pii/S1808869415312611.

[24] Joffrey Leevy, Taghi Khoshgoftaar, Richard Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5, 11 2018. doi: 10.1186/s40537-018-0151-6.

[25] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002. ISSN 1076-9757.

[26] https://www.nasa.gov/multimedia/podcasting/twan$_1$1$_2$3$_1$2.*html*,2012.

[27] Dror Ben-Zeev, Rui Wang, Saeed Abdullah, Rachel Brian, Emily Scherer, Lisa Mistler, Marta Hauser, John Kane, Andrew Campbell, and Tanzeem Choudhury. Mobile behavioral sensing for outpatients and inpatients with schizophrenia. *Psychiatric Services*, 67:appi.ps.2015001, 12 2015. doi: 10.1176/appi.ps.201500130.

[28] Sohrab Saeb, Emily Lattie, Konrad Kording, and David Mohr. Mobile phone detection of semantic location and its relationship to depression and anxiety. *JMIR mHealth and uHealth*, 5:e112, 08 2017. doi: 10.2196/mhealth.7297.

[29] Walter Gerych, Emmanuel Agu, and Elke Rundensteiner. Classifying depression in imbalanced datasets using an autoencoder- based anomaly detection approach. pages 124–127, 01 2019. doi: 10.1109/ICOSC.2019.8665535.

[30] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, page 3–14, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329682. doi: 10.1145/2632048.2632054. URL https://doi.org/10.1145/2632048.2632054.

[31] http://www.who.int/en/.

[32] Seasonal influenza report, centers for disease control and prevention (cdc). URL http://www.cdc.gov/flu/.

[33] Google flu trends. URL http://www.google.org/flutrends/intl/en_us.

[34] Gianni Barlacchi, Christos Perentis, Abhinav Mehrotra, Mirco Musolesi, and Bruno Lepri. Are you getting sick? predicting influenza-like symptoms using human mobility behaviors. *EPJ Data Science*, 6:27, 12 2017. doi: 10.1140/epjds/s13688-017-0124-6.

[35] K. Farrahi, M. Cebrian, S. Moturu, A. Madan, and A. Pentland. Sensing the "health state" of a community. *IEEE Pervasive Computing*, 11(04):36–45, oct 2012. ISSN 1558-2590. doi: 10.1109/MPRV.2011.79.

[36] Tomasz Grzywalski, Marcin Szajek, Honorata Hafke-Dys, Anna Bręborowicz, Jędrzej Kociński, Anna Pastusiak, and Riccardo Belluzzo. Respiratory system auscultation using machine learning - a big step towards objectivisation? page PA2231, 09 2019. doi: 10.1183/13993003.congress-2019.PA2231.

[37] Yohan Chon, Nicholas D. Lane, Fan Li, Hojung Cha, and Feng Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In

Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12, page 481–490, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312240. doi: 10.1145/2370216.2370288. URL https://doi.org/10.1145/2370216.2370288.

[38] Kiran K. Rachuri, Mirco Musolesi, Cecilia Mascolo, Peter J. Rentfrow, Chris Longworth, and Andrius Aucinas. Emotionsense: A mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, page 281–290, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605588438. doi: 10.1145/1864349.1864393. URL https://doi.org/10.1145/1864349.1864393.

[39] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, page 45–57, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334945. doi: 10.1145/2742647.2742674. URL https://doi.org/10.1145/2742647.2742674.

[40] D. Oletic and V. Bilas. Energy-efficient respiratory sounds sensing for personal mobile asthma monitoring. *IEEE Sensors Journal*, 16(23):8295–8303, 2016. doi: 10.1109/JSEN.2016.2585039.

[41] Charles Bales, Muhammad Nabeel, Charles John, Usama Masood, Haneya Qureshi, Hasan Farooq, Iryna Posokhova, and Ali Imran. Can machine learning be used to recognize and diagnose coughs?, 04 2020.

[42] Emiel Miltenburg, Benjamin Timmermans, and Lora Aroyo. The vu sound corpus: Adding more fine-grained annotations to the freesound database. 05 2016.

[43] Igor Miranda, Andreas Diacon, and Thomas Niesler. A comparative study of features for acoustic cough detection using deep architectures *. volume 2019, pages 2601–2605, 07 2019. doi: 10.1109/EMBC.2019.8856412.

[44] Pavol Partila, Jaromir Tovarek, Jan Rozhon, and Jakub Jalowiczor. Human stress detection from the speech in danger situation. page 31, 05 2019. doi: 10.1117/12.2521405.

[45] Lu-Shih Low, Namunu Maddage, Margaret Lech, Lisa Sheeber, and Nicholas Allen. Influence of acoustic low-level descriptors in the detection of clinical depression in adolescents. pages 5154 – 5157, 04 2010. doi: 10.1109/ICASSP.2010.5495018.

[46] Francisco Contreras-Ruston Adrián Castillo-Allendes. Voice therapy in the context of the covid-19 pandemic: Guidelines for clinical practice. 08 2020. doi: https://doi.org/10.1016/j.jvoice.2020.08.001.

[47] Neil Zeghidour, Nicolas Usunier, Gabriel Synnaeve, Ronan Collobert, and Emmanuel Dupoux. End-to-end speech recognition from the raw waveform. pages 781–785, 09 2018. doi: 10.21437/Interspeech.2018-2414.

[48] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomput.*, 184(C):232–242, April 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2015.08.104. URL https://doi.org/10.1016/j.neucom.2015.08.104.

[49] H. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22: 400–407, 2007.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[51] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7, 07 2020. doi: 10.1186/s40537-020-00320-x.

[52] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001. ISSN 0899-7667. doi: 10.1162/089976601750264965. URL https://doi.org/10.1162/089976601750264965.

[53] https://scikit-learn.org/stable/.

[54] Hannah Peckham, Nina Gruijter, Charles Raine, Anna Radziszewska, Coziana Ciurtin, Lucy Wedderburn, Elizabeth Rosser, Kate Webb, and Claire Deakin. Male sex identified by global covid-19 meta-analysis as a risk factor for death and itu admission. *Nature Communications*, 11, 12 2020. doi: 10.1038/s41467-020-19741-6.

[55] https://iisc.ac.in/, .

[56] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, page 319, Berlin, Heidelberg, 1999. Springer-Verlag. ISBN 3540667229.

[57] Chengxi Ye, Matthew Evanusa, Hua He, Anton Mitrokhin, Tom Goldstein, James A. Yorke, Cornelia Fermüller, and Yiannis Aloimonos. Network deconvolution, 2020.

[58] https://pytorch.org/.

[59] https://pandas.pydata.org/.

[60] https://numpy.org/.

[61] https://librosa.org/doc/latest/index.html.

[62] https://plotly.com/.

[63] Chloe Brown, Chauhan Jagmohan, Andreas Grammenos, Jing Han, Apinan Hasthanasombat, Dimitris Spathis, Tong Xia, Pietro Cicuta, and Cecilia Mascolo. Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data. 06 2020.

[64] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

[65] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet, 2019.

[66] Prateek Verma and Jonathan Berger. Audio transformers:transformer architectures for large scale audio understanding. adieu convolutions, 2021.

[67] https://research.google.com/audioset/dataset/cough.html.