ISSN No:-2456-2165

COVID-19 Diagnosis using Cough Recordings

Gayathri B R, Karthika R, Sneha A, S Varsha
Department of Mathematics,
Amrita School of Physical Sciences, Kochi, Amrita Vishwa Vidyapeetham, India

Abstract:- The COVID-19 pandemic has caused significant disruptions to global health, so-ciety, and the economy. Rapid and accurate detection of COVID-19 is crucial in minimizing community outbreaks and controlling the spread of the virus. This study proposes an audio-based digital testing method for COVID-19, eliminat-ing the need for patients to travel to testing laboratories. By analyzing cough noises using machine learning and deep learning techniques, the presence of COVID-19 can be detected and classified. The study evaluates multiple ma-chine learning models on the Coughvid dataset and assesses their performance in terms of accuracy. The results reveal that gradient boost achieves the highest accuracy of 88.82%, followed closely by Xgboost with an accuracy of 88.53%. Decision tree-based models, such as the Voting Classifier and Adaboost, also exhibit strong performance with accuracies above 88%. Logistic Regression, Deep Belief Network, MLP, Random Forest, and CNN demonstrate accuracies ranging from 87% to 88%. However, Linear Discriminant Analysis, PCA, Au-toencoder, and Naïve Bayes achieve comparatively lower accuracies, suggesting potential limitations in capturing the complexity of the dataset. The proposed audio-based digital testing method offers a promising approach to COVID-19 detection, providing a non-invasive and cost-effective solution for widespread testing and monitoring. The findings highlight the importance of leveraging machine learning techniques in healthcare and pave the way for further ad-vancements in audio-based COVID-19 detection methods.

Keywords:- *COVID-19, Cough Diagnosis, Deep Learning, Machine Learning, CNNs, Ensemble Methods, Voting Classifiers, Coughvid Dataset.*

I. INTRODUCTION

The SARS-CoV-2 virus, which is the source of the coronavirus disease (COVID-19) pandemic, started in the Chinese city of Wuhan and has since quickly spread to many other nations [1]. Many cases have been documented globally, and the number of cases is continually rising. It has severely destabilised the world's health, society, and economy.

Coughing is one of the main signs of the illness that are seen in people

[2] It might be challenging to determine which respiratory ailment a cough is associated with because cough is a symptom of many different respiratory conditions. Moreover, the COVID-19 disease spread quickly from the droplets of the infected patient, thus aiding in the unintentional spread of the illness. In order to minimise community outbreaks, the situation necessitates the rapid and accurate detection of COVID-19 through regular and broad testing.

Most infected individuals will experience mild to moderate respiratory in-fection, while others may recover without the need for special care [3]. In some circumstances, the patient will get extremely ill and might need medical care. Serious illnesses are more likely to affect older persons, people with other chronic respiratory disorders, cancer, diabetes, or cardiovascular disease. Being well-informed about the illness and having a clear understanding of how the virus spreads are two of the best strategies to stop or at least slow down its spread. When an infected person coughs, sneezes, speaks, or breathes, little liquid par-ticles from their mouth or nose carry the virus [4]. So, the major strategy is to maintain isolation or engage in self-quarantine when there are similar symptoms.

Currently, the most used COVID-19 diagnosis method is the RT-PCR test, which is pricy and has a number of drawbacks. Its drawbacks include the necessity for qualified medical professionals, disposable testing kits, intrusive nature, and the length of time required to acquire findings. Also, due to the severity of the sickness, it may not always be possible for the patient to be present at the test centre. Also, there is a chance that the disease could spread within the testing facilities. The emergence of the omicron form has made home-based COVID-19 monitoring quite popular. Further screening or monitoring techniques that could ease the load on testing centres, be used from a distance, and be appropriate for extensive testing and monitoring are desperately needed.

In this study, we propose an audio-based digital COVID-19 testing method that would significantly slow the spread of the disease because it eliminates the need for the patient to travel to the lab for the test. Sound data represented digitally may disclose important information that is undetectable by humans. Furthermore, there is convincing evidence that COVID-19 may be identified in cough noises using machine learning and deep learning techniques. Cough noises may enable the sickness to be detected and will have a significant financial impact because they are easily transformed to information signals and saved as digital files [5] [6] [7].

II. RELATED WORKS

The most recent research on using cough sounds to diagnose COVID19 is re-viewed in this part, along with knowledge on how to classify sounds using ma-chine learning and deep learning. Pre-processing, extraction, and classification stages make up the sound classification process. The majority of sound de-tection surveys concentrate on sound synthesis and sound recognition using conventional machine learning methods. The majority of works on sound detec-tion concentrate on sound synthesis and sound recognition using conventional machine learning methods.

A supervised deep neural network-based cough sound analysis for COVID-19 identification, which offers cutting-edge performance on the benchmark datasets in measures like Accuracy and AUC, has been presented in [8]. The suggested multi-branch model, known as MSCCov19Net, is denoted by the inputs Mel Frequency Cepstral Coefficients (MFCC), Spectrogram, and Chromagram. The system is evaluated on two clinical and non-clinical datasets that have only been used for testing, and it is trained using publically accessible crowdsourced datasets. The suggested system outperforms the six most prominent deep learn-ing architectures on four datasets, according to experimental results, since it is more generalizable. They utilised the Coughvid, Coswara, Virufy, and No-CoCoDa databases. The MFCC-based model, the Spectrogram-based model, the Chromagram-based model, and the Ensemble MSCCov19Net model are the trained models.

They employed log-mel spectrogram features from the audio signals of people coughing in [9]. The crowd-sourced cough samples obtained from COVID-19-positive and -negative patients across a wide range of demographic backgrounds are described in the COUGHVID datasets used here. They looked into the usage of VGG-13-style convolutional networks for detection. By doing data augmen-tation utilising the COVID-19 positive cough sounds from the COUGHVID dataset and using ensembles of two VGG13 models, which were trained using the cross-entropy loss, they improved the overall generalisation performance.

In [10], Pahar et al. use a deep learning-based automatic cough classifier to separate COVID-19 coughs from healthy coughs and tuberculosis (TB) coughs. They employed the synthetic minority oversampling method to deal with unbal-anced data. The Coswara and Sarcos dataset is what is being used. The cough data were utilised to train and assess a CNN, LSTM, and Resnet50. The cough data included 1.68 hours of TB coughs, 18.54 minutes of COVID-19 coughs, and 1.69 hours of healthy coughs from 47 TB patients, 229 COVID-19 patients, and 1498 healthy patients. The F1 score of 92.5% and 86.3% from the pre-trained Resnet50 demonstrates the great accuracy of the model.

[11] is a model built on a convolutional neural network (CNN) and coupled with a sound camera to visualise cough sounds. The binary classifiers were streamlined using VGGNet, GoogLeNet, and ResNet. A total of 39 examples were covered by the training, and the performance was verified using an F1 score of 91.9% and a test accuracy of 97.2%.

[12] is a system for automatically and non-intrusively recognising cough in-cidents based on both accelerometer and audio inputs. The audio signals were recorded by an external microphone on the same smartphone. 14 adult male patients' cough and non-cough occurrences totaling over 6000 each were used. A foundation for deep architecture is provided by logistic regression (LR), support vector machine (SVM), and multilayer perceptron (MLP). The cross-validation method utilised residual-based architecture (Resnet50), long short-term memory (LSTM) network, and convolutional neural network (CNN), which all produced accuracy scores of 0.98 and 0.99.

In [13] Dentamaro et al. presents a novel deep neural network architecture for audio classification (based on breath and cough) which can be engineered for smartphones in a distributed scenario. Here Auditory Cortex ResNet, briefly AUCO ResNet, is proposed and tested and also been tested on the famous UrbanSound 8K dataset, achieving a state of accuracy.

[14] used Resnet50, LSTM, and CNN (used transfer learning). They came to the conclusion that the pre-trained Resnet50 classifier performs optimally or very optimally across all datasets and all three audio classes whether either fine-tuned or utilised as a bottleneck extractor. In addition to better performance, transfer learning using the larger dataset without COVID-19 labels also resulted in a significantly lower standard deviation of the classifier AUC, indicating better generalisation to omitted test data. This standard deviation was decreased, and performance was close to ideal, by using bottleneck features, which are retrieved by the pre-trained deep models and are thus also a technique of incorporating out-of-domain data.

[15] is a deep learning-based model for classifying coughs in children that can identify between pathological coughs in kids caused by conditions like asthma, upper respiratory tract infections (URTI), and lower respiratory tract infections from healthy coughs in kids (LRTI). The models employed are Mel-Frequency Cepstral Coefficients (MFCCs)-based Bidirectional Long-Short-Term Memory Networks (BiLSTM) and LSTMs. When used to classify for just two classes of coughs—healthy or pathology—accuracy increased to over 84%; when used to distinguish between three different pathologies, the accuracy exceeded 91% for all three; however, when used to classify for four different pathologies, accuracy decreased (one was consistently misclassified as the other); and a longitudi-nal study of MFCC feature space when comparing pathological and recovered coughs collected from the same subjects revealed that pathological colitis The data is divided into two cohorts: the pathological cohort (which included res-piratory conditions such as LRTI, URTI, and asthma; LRTI included a spec-trum of respiratory diseases such as bronchiolitis, bronchitis,

bronchopneumo-nia, pneumonia, and lower respiratory tract infection) and the healthy cohort. From Singapore's KK Children's Hospital, participants were sought out. The Children's Emergency Department, the Respiratory Ward, and the Respiratory Clinic were used to recruit pathological cohorts. The sounds of their coughing were captured on first appearance at the hospital. The Children Surgical Unit was used to find the healthy cohorts. The Healthy vs. Pathology (2-Class), Healthy vs. LRTI, Healthy vs. URTI, Healthy vs. Asthma, and Healthy vs. Pathology (4-Class) models are trained models.

In [16], they describe a method for converting patient cough recordings into frames and analysing waveforms based on various parameters to distinguish be-tween COVID-19 patients and healthy individuals. The dataset comprises of 86 cough audios captured at 44 kHz, of which 54 were recorded from patients who tested positive for COVID-19 and 32 from healthy individuals. Moreover, 18 healthy cough audio samples were taken from the Freesound database and 46 healthy cough audio samples from the Coswara database at the Indian Insti-tute of Science, bringing the total number of healthy audio samples to 96 and the overall number of cough audio samples to 150. The three most appropri-ate machine learning classification models are used for classification: Logistic Regression, Support Vector Machines (SVM), and Random Forest. The first fif-teen dominant features of these three classifiers are provided in order to obtain the result, and the best classifier for the COVID-19 cough detection algorithm's implementation in the real world is determined.

The open cough dataset is used in [17] to propose a CNN-based audio clas-sifier. The dataset is manually categorised into types of cough before being divided into Covid and Non-Covid classes. The second dataset, the ESC-50 dataset, is a collection of brief environmental audio encompassing 2000 clips of 44.1 kHz with 50 classes. The dataset is made up of 871 cough YouTube videos and 40 audio files. The two methods that are suggested in this research are based on mfcc characteristics, which are features that speech recognition systems use. This method outperformed the spectrogram-based approach, pro-viding test accuracy and sensitivity of 70.58% and 81%, respectively.

In [18] they came up with an idea to use the cough audio sample around the world to develop an AI model in order to predict the chances of having the disease. Open-source datasets from Coswara containing 1543 samples and Coughvid containing 20072 samples were used. ROC-AUC algorithm was used giving an accuracy of 77.1%.

In [19], Barata et al. put out a step towards cough detection that is afford-able, scalable, and device-independent. They conducted a lab experiment with 43 participants and used 5 different recording equipment to capture 6737 cough samples and 8854 control sounds. Then, in order to lessen the disparity between devices, they reimplemented two strategies from earlier work and examined their performance in two distinct scenarios across devices utilising an effective CNN architecture and an ensemble-based classifier. The method outperformed ear-lier learning algorithms and provided mean accuracies between [85.9%, 90.9%], demonstrating consistency among devices (SD = [1.5%, 2.7%]).

III. METHODOLOGY

A. Dataset Description

The CoughVid dataset[20] we used includes audio files and metadata related to coughing sounds. The information was gathered from people who have COVID-19 that was suspected or known to exist, as well as from healthy people for comparison.

Each audio file is saved in the WAV format and is given a special UUID. A collection of audio samples that were synthesised by adding background noise to the original cough recordings are also included in the dataset. To give algorithms for cough detection more realistic testing environments, these clips are included.

Table 1: Related Works

Author	**Year**	**Dataset**	**Model**
Ulukaya et al. [8]	2023	Coughvid, Coswara, Virufy, NoCoCoDa	CNNmodel, ResNet50 ,Efficient NetBo, MobileNetV2, Xception
Sunil et al. [9]	2022	DiCOVA 2021, Coughvid	VGG13
M. Pahar et al. [10]	2022	Sarcos, Brooklyn, TASK dataset, Wallacedene, Coswara, ComParE	CNN, LSTM
Lee, Gyeong-Tae, et al. [11]	2022	GoogleAudioSet, HU-MAN20200923, SMI-office	VGGNet, GoogLeNet, ResNet

Pahar, Madhurananda, et al. [12]	2022	Real coughset	LR, SVM, MLP, CNN, LSTM, Resnet50
Dentamaro, Vincenzo, et al. [13]	2022	Google Audio Set, Freesound, Librispeech	CNN, LSTM, Resnet50
Pahar M et al. [14]	2021	Coswara, Cambridge dataset	AUCO ResNet ,DenseNet, Inception ResNet, ResNet 50, Shallow SVM, Shallow Random Forest, Shallow KNN
Balamurali et al. [15]	2021	Unspecified cough sounds	BiLSTM
Vrindavanam, Jayavrinda, et al. [16]	2021	Coswara	Logistic Regression, Support Vector Machines (SVM), Random Forest
Bansal et al. [17]	2020	SARS-CoV-2	CNN with MFCC input
Chaudhari et al. [18]	2020	Coswara, Coughvid	1D and 2D CNNs, LSTM, CRNN
Barata, Filipe, et al. [19]	2019	Live datasets	Random Forest with PCA, K-Nearest Neighbor with LocalHu Moments

It includes a metadata file in CSV format that provides additional informa-tion about each audio recording in the dataset. It contains 27550 rows and 11 columns, with each row representing a unique audio sample and each column containing various information about the recordings.

The metadata file contains the following fields:

1. UUID: A unique identifier for each audio recording in the dataset.
2. Datetime: The date and time when the audio recording was obtained.
3. Cough detected: Ranges from 0 to 1, with higher values indicating a higher percentage of coughing sounds detected in the recording.
4. SNR: (Signal-to-Noise Ratio) is a measure of the strength of a signal rel-ative to the background noise present in the recording.
5. Latitude: The latitude of the location where the audio was recorded, if available.
6. Longitude: The longitude of the location where the audio was recorded, if available.
7. Age: The age of the person making the coughing sound, if available.
8. Gender: The gender of the person making the coughing sound, if available.
9. Respiratory Condition: Indicates whether the person making the coughing sound had a pre-existing respiratory condition, if available.
10. Fever muscle pain : Indicates whether the person making the coughing sound had fever and muscle pain, if available.
11. Status: Indicates whether the recording was obtained from a COVID-19 patient or a healthy individual or if it is symptomatic, if available.

B. Data Preprocessing

Prior to conducting any analysis, the dataset was subjected to a series of preprocessing steps to ensure that the data was in a suitable format for analysis. It was transformed to ensure that all numerical attributes were on a consistent scale.

The age attribute was normalized to give values 0, 1, 2 and 3 to male, female, other and null values respectively. The categorical attributes like respiratory condition and fever muscle pain were given values 0, 1 and 2 indicating False, True and null values respectively. Finally, the status attribute was given values 0, 1, 2 and 3 indicating healthy, COVID-19, symptomatic and null values respectively.

C. Proposed Models

➢ *Gradient Boost*

Gradient Boosting [21] is a machine learning technique that can be used for classification tasks. It is an ensemble

method that combines multiple weak classifiers, typically decision trees, to create a strong predictive model. In gradient boosting for classification, the algorithm iteratively builds an en-semble of decision trees [22]. Each tree is trained to correct the mistakes made by the previous trees in the ensemble. The algorithm starts with an initial model, usually a simple one like a single decision tree or a constant prediction. Then, it sequentially adds decision trees to the ensemble, each one focused on minimizing the errors or residuals of the previous model. During the training process, gradient boosting optimizes a loss function [23] that quantifies the difference between the predicted labels and the true labels of the training data. The algorithm uses gradient descent to minimize this loss function, updating the model by taking steps in the direction of steepest descent. The final prediction of the gradient boost-ing [22] model is made by aggregating the predictions of all the individual trees in the ensemble. Typically, this aggregation is done by taking the majority vote (in the case of binary classification) or by using soft voting (probabilities) for multi-class classification. Gradient boosting has proven to be a powerful technique for classification tasks [22], often outperform-ing other algorithms in terms of accuracy and robustness. Some popular implementations of gradient boosting for classification include XGBoost, LightGBM, and CatBoost.

- *Logistic Regression*

It is a statistical method used to analyse the relationship between depen-dent and independent variables. The main goal is to predict the probabil-ity of dependent variable by taking a specific value which is based on the independent variables [24]. It transforms the linear relationship between the independent variables and the dependent variable into a non-linear relationship that ranges between 0 and 1 by fitting a logistic function to the data [25]. It is important to consider other algorithms and hyper-parameters to ensure the best performance. The methodology involves a rigorous approach to selecting the best hyperparameters for the logistic regression algorithm using grid search with cross-validation. It helps to optimize the model's performance and increase its accuracy in predicting the target variable [26].

- *Decision Tree*

A Decision Tree [27] is a supervised machine learning algorithm used for classification tasks. It creates a tree-like model where each internal node represents a feature and each leaf node represents a class label. During the training phase, the algorithm recursively splits the data based on the best features, using criteria like Gini impurity or information gain. In the prediction phase, new data points are classified by traversing the decision tree based on their feature values, ultimately assigning them a class label at a leaf node. Decision Trees are interpretable, handle both numerical and categorical features, and provide insights into feature importance. However, they can be prone to overfitting and bias towards unbalanced datasets. Techniques like pruning and ensemble methods such as Random Forests can mitigate these limitations and enhance classification accuracy.

- *Linear discriminant analysis*

LDA (Linear Discriminant Analysis) [28] is a dimensionality reduction technique commonly used for classification tasks. It aims to find a lin-ear combination of features that maximizes the separation between classes while minimizing the variance within each class. During training, LDA es-timates the mean vectors and covariance matrices for each class, and com-putes the linear discriminants that capture the most discriminative infor-mation. The input features are then projected onto this lower-dimensional subspace, reducing the dimensionality of the data. LDA assumes linear decision boundaries and works best when classes are well-separated. How-ever, it may not perform well with overlapping classes or when assump-tions are violated. LDA is a supervised learning algorithm and requires labeled training data. Overall, LDA is a valuable tool for dimensionality reduction and classification tasks, particularly when linear separability is present and assumptions are met.

- *Deep belief network (DBN)*

A Deep Belief Network (DBN) [29] is a powerful neural network archi-tecture that can be used for both generative modeling and classification tasks. In classification, a DBN is capable of learning hierarchical represen-tations of data, enabling it to capture intricate patterns and discriminative features. The DBN consists of multiple layers of Restricted Boltzmann Machines (RBMs), which are trained in an unsupervised manner layer by layer. This pretraining phase allows the DBN to learn a compressed representation of the input data. Each RBM captures the statistical de-pendencies among its hidden and visible units, gradually building a deeper and more expressive model. Once the RBMs are pretrained, the DBN un-dergoes a fine-tuning phase where supervised learning techniques, such as backpropagation, are used to adjust the weights and biases. This fine-tuning process further refines the model's parameters by utilizing labeled data to optimize its classification performance. DBNs excel in capturing complex relationships within data by learning multiple levels of abstrac-tion. This hierarchical representation enables them to extract meaningful features and make accurate predictions in classification tasks. DBNs have been successfully applied to various domains, including image recognition, natural language processing, and speech recognition.

- *Principal component analysis (PCA)*

Principal Component Analysis (PCA) [30] is a commonly used dimen-sionality reduction technique that can also be used for classification tasks. PCA aims to transform high-dimensional data into a lower-dimensional subspace while preserving the maximum amount of variance. The new subspace is defined by a set of orthogonal vectors, known as principal components, which capture the most important patterns in the data. In classification tasks, PCA can be used to reduce the number of features and improve the computational efficiency of the learning algorithm. By pro-jecting the original data onto the principal components, the new features can be selected based on the amount of variance they explain, thereby

selecting the most informative features for classification. This process can help to reduce overfitting and improve generalization performance, partic-ularly when dealing with high-dimensional data.

- *Multilayer Perceptron (MLP)*

Multi-Layer Perceptron (MLP) [31] is a type of artificial neural network commonly used for classification tasks. It consists of multiple layers of interconnected nodes called neurons, with each neuron applying an ac-tivation function to its inputs. MLPs are capable of learning complex nonlinear relationships between input features and target outputs. In the context of classification, MLPs can be trained using a supervised learn-ing approach. The network is fed with input features, and the weights of the connections between neurons are adjusted through a process called backpropagation. During training, the network learns to map the input features to the corresponding class labels, gradually improving its abil-ity to make accurate predictions. MLPs can handle both numerical and categorical input features, making them versatile for various types of clas-sification problems. They have the ability to learn and capture intricate patterns in the data, allowing for high flexibility and expressive power. Additionally, MLPs can automatically extract relevant features from raw data, reducing the need for manual feature engineering.

- *Autoencoder*

An autoencoder [32] is a type of neural network that can be used for both unsupervised learning and feature extraction in classification tasks. The basic idea behind an autoencoder is to learn a compressed representation of the input data by training the network to reconstruct the input from a lower-dimensional representation. The autoencoder consists of two main components: an encoder that maps the input data to a lower-dimensional representation, and a decoder that maps the lower-dimensional represen-tation back to the input space. During training, the network is optimized to minimize the difference between the input and the reconstructed out-put. In classification tasks, an autoencoder can be used to extract features from the input data that are most relevant to the classification task. Once the features have been extracted, they can be fed into a separate classifi-cation algorithm, such as a support vector machine or logistic regression, to perform the actual classification. Autoencoders are particularly useful for tasks where the input data is high-dimensional and noisy, as they can learn to extract the most salient features while ignoring the noise. They have been successfully applied to a variety of classification tasks, including image recognition, speech recognition, and natural language processing.

- *Voting Classifier*

The Voting Classifier [33] is an ensemble learning method used for classifi-cation tasks. It combines the predictions of multiple individual classifiers and determines the final prediction by majority voting or by weighted voting. Each individual classifier in the ensemble can be a different clas-sification algorithm or even multiple instances of the same algorithm with different hyperparameters. In majority voting, the class label that receives the highest number of votes from the individual classifiers is selected as the final prediction. In weighted voting, each classifier's vote is weighted according to its performance or confidence level. The Voting Classifier can handle binary and multi-class classification problems. The Voting Classi-fier benefits from the wisdom of multiple classifiers and can often improve overall prediction accuracy compared to using a single classifier. It is es-pecially useful when the individual classifiers have different strengths and weaknesses or when they perform well on different subsets of the data.

- *Convolutional Neural Network (CNN)*

A Convolutional Neural Network (CNN) [34] is a powerful deep learning model widely used for image classification tasks. Its architecture consists of convolutional layers that apply filters to extract features from images, pooling layers that downsample the features, and fully connected layers that produce the final classification output. CNNs excel at capturing spatial hierarchies and local patterns in images, allowing them to learn intricate features from raw pixel data. Through a training process involv-ing forward and backward propagation, CNNs optimize their weights to minimize a loss function. CNNs have achieved remarkable success in im-age classification benchmarks and have been applied to diverse domains such as natural language processing and speech recognition. While CNNs require ample training data and computational resources, advances in techniques like pretraining and transfer learning have made them more accessible for a wide range of classification tasks.

- *Random forest*

Random Forest [35] is an ensemble learning algorithm used for classifica-tion tasks that combines multiple decision trees. It derives its name from the random selection of features and data samples used in each tree. Ran-dom Forest builds a collection of decision trees, where each tree is trained on a random subset of the training data and considers a random subset of features at each node. This randomness helps reduce overfitting and increases the diversity of the trees. During prediction, each tree indepen-dently classifies the input data, and the final prediction is determined by majority voting or averaging the individual tree predictions. This ensem-ble approach improves overall accuracy and robustness. Random Forest handles high-dimensional datasets with a large number of features, auto-matically handles feature selection, and accommodates both categorical and numerical features. It also provides a measure of feature importance, aiding in understanding the relevance of features in the classification pro-cess. Random Forest's advantages include resilience to overfitting, effec-tive handling of noisy data, outliers, and missing values. However, it can be computationally expensive and may require hyperparameter tuning. Random Forest finds applications in diverse domains such as healthcare, finance, and bioinformatics, owing to its versatility, robustness, and ability to handle complex classification problems.

➢ *Adaboost*

AdaBoost (Adaptive Boosting) [36] is an ensemble learning algorithm used for classification tasks that combines multiple weak classifiers to create a strong classifier. It works by iteratively training weak classifiers on differ-ent weighted versions of the training data, with a focus on the misclassi-fied examples. In each iteration, the weights of the misclassified examples are increased, while the weights of the correctly classified examples are decreased. This process allows subsequent weak classifiers to pay more attention to the challenging examples, improving the overall classifica-tion performance. AdaBoost is particularly effective in handling complex datasets and boosting the performance of weak classifiers. By combin-ing multiple weak classifiers, it can capture different aspects of the data and make more accurate predictions. AdaBoost is known for its ability to handle both categorical and numerical features, and it is robust against overfitting. It also has the advantage of handling class imbalance by as-signing higher weights to the minority class examples, ensuring that they are correctly classified.

➢ *XGboost*

XGBoost [22] [37] is a popular open-source machine learning library that is used for gradient boosting. It is designed to be highly efficient, scal-able, and flexible, and is often used for classification and regression tasks. XGBoost works by iteratively adding new trees to a model, with each tree correcting the errors made by the previous tree. The methods includes by using XGBoost to build a classification model on a dataset. XGBoost incorporates several key features that contribute to its effectiveness in

classification. It includes regularization techniques to prevent overfitting, such as L1 and L2 regularization, and the ability to handle missing values within the data. It also provides a flexible framework for defining custom evaluation metrics and supports parallel processing to speed up training. Furthermore, XGBoost has a built-in capability to handle class imbalance. By assigning weights to different classes or using sampling techniques, XG-Boost can address the challenge of imbalanced datasets and provide better predictive performance for minority classes.

➢ *Naive Bayes*

Naive Bayes [38] is a popular machine learning algorithm used for classi-fication tasks. It is based on Bayes' theorem and assumes that the fea-tures are conditionally independent given the class variable. Despite its simplistic assumption, Naive Bayes often performs well in practice and is widely used due to its simplicity and efficiency. The algorithm works by calculating the probability of each class given a set of features using Bayes' theorem. It estimates the class probabilities by multiplying the prior probabilities of the classes with the conditional probabilities of the features given each class. The class with the highest probability is then assigned as the predicted class for the input data. Naive Bayes is par-ticularly suited for text classification tasks, such as spam detection and sentiment analysis. It works well with high-dimensional datasets and can handle large feature spaces efficiently. Naive Bayes classifiers are known for their fast training and prediction times, making them suitable for real-time applications. There are different variants of Naive Bayes, including Gaus-sian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes, which are suitable for different types of data and assumptions about the distribution of features.

IV. RESULT ANALYSIS

Based on the analysis of the results conducted on the Coughvid dataset, multiple machine learning models were evaluated for classification tasks. The accuracy of each model was assessed to determine their performance. Among the models tested, Gradient Boost achieved the highest accuracy of 88.82%, making it the top-performing model. Xgboost closely followed with an accuracy of 88.53%, demonstrating its effectiveness in classifying the Coughvid data. The Voting Classifier, Decision Tree, and Adaboost models also performed well, all achieving accuracies above 88%. Logis-tic Regression, Deep Belief Network, MLP, Random Forest, and CNN at-tained accuracies ranging from 87% to 88%, showcasing their capability in accurately classifying the Coughvid data. However, Linear Discriminant Analysis, PCA, Autoencoder, and Na¨ıve Bayes achieved comparatively lower accuracies, suggesting potential limitations in capturing the com-plexity of the dataset. In conclusion, the results indicate that ensemble methods like Gradient Boost, Xgboost, and the Voting Classifier, as well as decision tree-based models, offer strong classification performance for the Coughvid dataset.

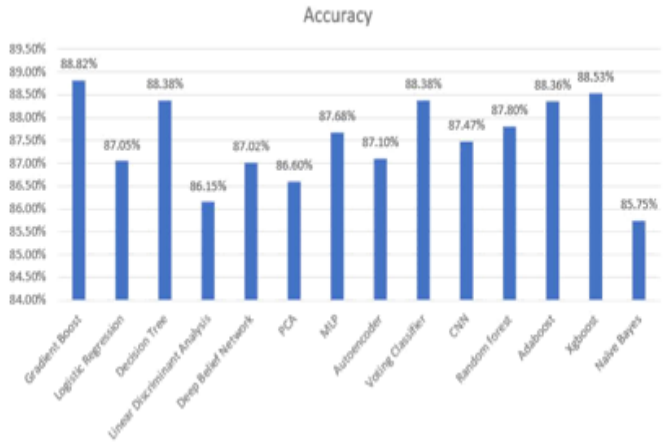

Fig 1: Accuracy Scores

V. CONCLUSION

In conclusion, this study proposes an audio-based digital COVID-19 test-ing method that eliminates the need for patients to travel to testing lab-oratories. By analyzing sound data digitally, valuable information about the presence of COVID-19 can be detected, which may not be discernible to humans.

Machine learning and deep learning techniques have shown promise in identifying COVID-19 from cough noises. The results of the analysis conducted on the Coughvid dataset demonstrated the effective-ness of various machine learning models in classifying COVID-19. Gra-dient Boost emerged as the top-performing model with an accuracy of 88.82%, closely followed by Xgboost with an accuracy of 88.53%. The Voting Classifier, Decision Tree, and Adaboost models also achieved accu-racies above 88%. Logistic Regression, Deep Belief Network, MLP, Ran-dom Forest, and CNN demonstrated strong classification capabilities with accuracies ranging from 87% to 88%. However, Linear Discriminant Anal-ysis, PCA, Autoencoder, and Na¨ıve Bayes achieved comparatively lower accuracies, suggesting potential limitations in capturing the complexity of the dataset.

The proposed audio-based digital COVID-19 testing method has the po-tential to significantly slow down the spread of the disease by enabling remote testing and reducing the burden on testing centers. By leveraging machine learning techniques for COVID-19 detection from cough noises, it offers a non-invasive and cost-effective approach for widespread testing and monitoring. This research opens up possibilities for further advance-ments in audio-based COVID-19 detection methods and emphasizes the importance of leveraging machine learning and deep learning techniques in the field of healthcare.

REFERENCES

[1]. World Health Organization. Coronavirus disease (covid-19) pan-demic, 2021.
[2]. Centers for Disease Control and Prevention. Symptoms of coron-avirus, 2021.
[3]. World Health Organization. Covid-19 clinical management: Living guidance, 2021.
[4]. Centers for Disease Control and Prevention. How covid-19 spreads. https://www.cdc.gov/coronavirus/2019-ncov/ prevent-getting-sick/how-covid-spreads.html, 2021. Accessed on April 18, 2023.
[5]. J. Zhang, B. Xie, K. Hashimoto, and X. Su. Cough sound analysis for covid-19 detection. IEEE Journal of Engineering in Medicine and Biology, pages 1–1, 2021.
[6]. Ricardo Matos, Ricardo Afonso, Jo˜ao Bispo, Ricardo Correia, Rui Paiva, and Rosaldo Rossetti. Review on covid-19 diagnosis through cough sounds analysis. IEEE Access, 9:73175–73184, 2021.
[7]. Abhishek Vaid, Sriram K. Jaladanki, Hongming Xu, Benjamin Ng, and Ming Gao. Audio-based cough monitoring for covid-19 detection using machine learning. IEEE Transactions on Instrumentation and Measurement, 70:1–11, 2021.
[8]. Serkan Ulukaya, Ahmet Adem Sarıca, O˘guz Erdem, and Ahmet Karaali. Msccov19net: multi-branch deep learning model for covid-19 detection from cough sounds. Medical & Biological Engineering & Computing, pages 1–11, 2023. Epub ahead of print.
[9]. Sunil Rao, Vivek Narayanaswamy, Michael Espositoa, Jayaraman J. Thiagarajanb, and Andreas Spaniasa. Covid-19 detection using cough sound analysis and deep learning algorithms. School of ECEE, Sen-SIP Center, Arizona State University, 2022.
[10]. Mrinmoy Pahar, Sanjoy Mandal, Chiranjib Das, and Sarat Konwer. Automatic tuberculosis and covid-19 cough classification using deep learning. In 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), pages 1–9, Prague, Czech Re-public, 2022. IEEE.
[11]. Gyeong-Tae Lee, Tae-Woo Kim, Joon-Ho Cho, and Sung-Woo Park. Deep learning based cough detection camera using enhanced features. Expert Systems with Applications, 206:117811, 2022.
[12]. Madhurananda Pahar, Saurabh Sahoo, Suvendu Swain, Saurav Bera, Soura Das, and Goutam Panda. Automatic non-invasive cough de-tection based on accelerometer and audio signals. Journal of Signal Processing Systems, 94(8):821–835, 2022.
[13]. Vincenzo Dentamaro, Eugenio Di Sciascio, Salvatore Sessa, Mario Malcangi, and Marcello Castellano. Auco resnet: An end-to-end network for covid-19 pre-screening from cough and breath. Pattern Recognition, 127:108656, 2022.
[14]. Madhurananda Pahar, Martha Klopper, Robin Warren, and Thomas Niesler. Covid-19 detection in cough, breath and speech using deep transfer learning and bottleneck features. Computers in Biology and Medicine, 141:105153, Feb 2022.
[15]. B T Balamurali, Han In Hee, Supriya Kapoor, Onn H Teoh, Suang S Teng, Kae-Ping Lee, Dorien Herremans, and Jing-Ming Chen. Deep neural network-based respiratory pathology classification using cough sounds. Sensors, 21(16):5555, Aug 2021.
[16]. Jayavrinda Vrindavanam, Kavya Krishnan, Swetha Jayaraman, Shashank Bhat, and Anupama R Shankar. Machine learning based covid-19 cough classification models-a comparative analysis. In 2021 5th International Conference on Computing Methodologies and Com-munication (ICCMC). IEEE, 2021.
[17]. Vipin Bansal, Gaurav Pahwa, and Nirmal Kannan. Cough classifi-cation for covid-19 based on audio mfcc features using convolutional neural networks. In 2020 IEEE international conference on comput-ing, power and communication technologies (GUCON), pages 581– 585. IEEE, 2020.
[18]. Gautam Chaudhari, Xinyi Jiang, Ahmed Fakhry, Allen Han, Jiani Xiao, Shuang Shen, and Adil Khanzada. Virufy: Global applicability of crowdsourced and clinical datasets for ai detection of covid-19 from cough. arXiv preprint arXiv:2011.13320, 2020.
[19]. Filipe Barata, Lu´ıs Lopes, Jos´e Luis Oliveira, Rui Pedro Paiva, An-tonio Bugalho, Jose Gomes, Joao Figueiredo, and Ana Fred. Towards device-agnostic mobile cough detection with convolutional neural net-works. In 2019

IEEE International Conference on Healthcare Infor-matics (ICHI). IEEE, 2019.
[20]. Luka Orlandic, Tom´as Teijeiro, and David Atienza. The coughvid crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms. Scientific Data, 8(1):156, 2021.
[21]. Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5):1189–1232, 2001.
[22]. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boost-ing system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785– 794. ACM, 2016.
[23]. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The ele-ments of statistical learning. Springer, 2009.
[24]. David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. Logistic Regression Analysis. John Wiley Sons, 2013.
[25]. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning. Springer, 2013.
[26]. James Bergstra and Yoshua Bengio. Hyperparameter tuning in ma-chine learning: A practical guide. Neural networks: Tricks of the trade, pages 537–554, 2012.
[27]. Johannes F¨urnkranz. Decision Tree, pages 263–267. Springer US, Boston, MA, 2010.
[28]. Sunil Kumar Dash. A brief introduction to linear discriminant analy-sis. 2021. Published on August 18, 2021 and last modified on August 5, 2022.
[29]. Yuanyuan Shi, Jingqing Jiang, Xiaojing Fan, Jie Lian, Zhili Pei, and Mingyang Jiang. An improved dbn method for text classification. In Proceedings of the [Conference Name], volume 827 of Lecture Notes in Electrical Engineering. Springer, May 23 2022.
[30]. B. Shravan Kumar and Vadlamani Ravi. Text document classifica-tion with pca and one-class svm. In Proceedings of the [Conference Name], volume 515 of Advances in Intelligent Systems and Comput-ing. Springer, March 17 2017.
[31]. Francke Peixoto. A simple overview of multilayer perceptron (mlp). Data Science Blogathon, December 13 2020. Published on December 13, 2020 and Last Modified On December 18th, 2020.
[32]. Jason Brownlee. Autoencoder feature extraction for classification. Jason Brownlee's Machine Learning Mastery, December 7 2020. Pub-lished on December 7, 2020 in Deep Learning.
[33]. Mubarak Ganiyu. How voting classifiers work! a scikit-learn feature for enhancing classification. Towards Data Science, November 6 2020. Published in Towards Data Science, 4 min read.
[34]. Mohit Sewak, Md. Rezaul Karim, and Pradeep Pujari. Practical Con-volutional Neural Networks. Packt Publishing, February 2018.
[35]. Houtao Deng. An introduction to random forest: Illustration, in-terpretation, biases, and usage for outlier detection and clustering. Towards Data Science, December 2018.
[36]. Anshul Saini. Master the adaboost algorithm: Guide to implementing & understanding adaboost. Algorithm Beginner Machine Learning Technique, September 2021.
[37]. Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and Yoshua Ben-gio. Mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2018.
[38]. Daniel Jurafsky and James H. Martin. Naive Bayes and Sentiment Classification, chapter 4. Publisher, draft of january 7, 2023 edition, 2023.