COVID-19 Classification Using Cough Sounds

Dandy Arif Rahman
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
U-CoE AI-VLB
Bandung, Indonesia
23520043@std.stei.itb.ac.id

Dessi Puji Lestari
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
U-CoE AI-VLB
Bandung, Indonesia
dessipuji@stei.itb.ac.id

2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA) | 978-1-6654-1743-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAICTA53211.2021.9640278

***Abstract*— Coronavirus disease 2019 (COVID-19) is the currently happening pandemic. Up until mid-2021, the total cases of COVID-19 have reached 171 million worldwide. The virus is mainly transmitted through droplets generated when an infected person coughs, sneezes, or exhales. The most common occurring symptoms are fever, cough, and fatigue. The current diagnosis method is done through Reverse-Transcription Polymer Chain Reaction (RT-PCR) testing. Even though this is the current gold standard, this method has several downsides. The RT-PCR is costly, time-consuming, and can lead to another infection if done improperly. In this paper we try to utilize AI to classify COVID-19 using cough sound. This method can work as a triaging tool to help prioritize a person to get future-diagnosis.**

In this research, our contribution is trying several feature extractions, imbalance handling and modelling techniques to classify COVID-19 using cough sound. We obtained the best result using the combination of NMF-Spectrogram feature, undersampling method, and SVM. It gives the sensitivity of 90.9%, specificity of 55.6% and overall AUC-ROC of 73.3%. We also discovered that the NMF-Spectrogram feature works better than MFCC-based features.

Keywords—COVID-19, MFCC, NMF, Spectrogram, SVM, KNN, XGBoost

I. INTRODUCTION

First identified in Wuhan, China in December 2019, COVID-19 is an infectious disease caused by Severe Acute Respiratory Syndrome Corona Virus 2 (SARS-CoV-2). This disease attacks the sufferer's respiratory system which makes breathing difficult and can cause death. On March 11, 2020, the World Health Organization declared COVID-19 a global pandemic. Until now, COVID-19 has become a frightening epidemic in all corners of the world. Hundreds of millions of people have been infected and millions of people lost their lives due to the pandemic.

One effective way to reduce the transmission of COVID-19 that can be done is to conduct mass tests and isolate the COVID-19 cluster. However, this will be difficult to do by relying only on current testing methods, namely rapid antibody tests, antigens, and RT-PCR. On the economic side, these tests are quite expensive, moreover the implementation of these tests requires someone to come to a certain test center and most likely will interact with other people, this can increase the risk of transmitting COVID-19. Therefore, we need a test method that is economical, accurate, safe, and can be applied en masse.

Sound signals generated by the human body (sounds of breath, coughing, heartbeat, etc.) have been used routinely by medical personnel to diagnose a disease. One of the symptoms of COVID-19 is cough, especially dry cough. In this study, we hypothesize that cough sound could be a valuable information to detect whether someone is infected by the virus. In recent months, similar studies have been carried out with sampling of cough sounds and the construction of a COVID-19 classification model against these datasets [1]–[4]. This test method (detection COVID-19 by cough sound) is very economical, does not require contact (thereby reducing the risk of COVID-19 transmission), can be carried out in bulk and the results are fast. The results of these studies also show quite promising results. As the topic of COVID-19 detection using cough sound is still relatively new, there are still many aspects that can be developed, and this also what motivates this research.

II. RELATED WORKS

Even though this field is still considered as a new topic, several research have been conducted. Some research focuses only on data gathering using crowdsourcing method, while the other even have tried to create the classification model.

The first related work was developed by MIT researchers [2]. This model was trained using a dataset that they collected themselves, Opensigma dataset [2]. Broadly speaking, this model utilizes the MFCC features of cough recordings and the architecture used by the Convolutional Neural Network (CNN). This CNN architecture consists of 3 ResNet50 architectures arranged in parallel, then this architecture is added with an average pooling layer to unify the three outputs of ResNet50. Then at the end of the architecture there is a fully connected layer and a dense binary layer with sigmoid activation that can produce positive or negative COVID-19 prediction outputs. This model produces a sensitivity value of 98.5% & a specificity value of 94.2% (AUC 0.97) and for asymptomatic cases it produces a sensitivity value of 100% & a specificity of 83.2%. This model utilizes 4 biomarkers, namely muscular degradation, changes in vocal cords, changes in sentiment, and changes in the lung and respiratory tract. The use of these four biomarkers was inspired by the success of the research team in detecting Alzheimer's disease using these four biomarkers.

The next classification model still utilizes the CNN architecture. This model [5] implements end-to-end CNN that accepts a cough sample as input and produces an output in the form of a prediction of the presence of COVID-19 in the voice sample. The input used in this model is a cough that has been cut into segments with a length of 2 seconds. Then the logmelspectrogram feature is extracted from the cough sound sample. This model utilizes the ResNet-18 model and adds a pooling layer, 2 linear layers, and a final predictive layer with 2 neurons and a softmax activation function to predict whether the sample is COVID-19 or not.

The dataset used in this model training is a dataset collected by the researchers themselves, namely the Cough Against Covid dataset [5]. As explained in the dataset subsection, the size of this dataset is not too large, so in this study augmentation was carried out to enrich the dataset. The augmentation technique used is adding environmental background noise with the ESC-50 dataset [6] and performing time and frequency masking [7] on the input spectrogram.

Noise is selected at random during the training time and the amplitude of the noise is randomly modulated between 0.4 and 0.75 before being augmented to the input sample.

In this model, pretraining is also carried out to tolerate not too many datasets. First, ResNet-18 is initialized using the weights obtained from the pretrained ImageNet, while the weights for the other additional layers are initialized randomly. Then, the model was pretrained on an open source dataset [8]–[10] to detect the presence of cough detection in the dataset. The pretraining dataset used was imbalanced (4,793 coughing voices & 31,909 non-coughing voices), so that oversampling was carried out in the minority class. In this study, the ground truth used was the result of the RT-PCR test. Although this method is the best method for diagnosing COVID-19, this method sometimes also produces incorrect predictions with sensitivity values of 70% and specificity of 95% [11]. This can affect model training. Therefore, the standard label smoothing technique is applied [12]

In the study conducted by (Brown et al., 2020) the classification modeling of COVID-19 was carried out using Logistic Regression (LR), Gradient Boosting Trees and Support Vector Machines (SVM). The dataset used in this model is a dataset that they collect through crowdsourcing. The features used in this study consisted of two groups, namely handcrafted features and features obtained from transfer learning. The handcrafted features used consist of duration, onset, tempo, period, RMS energy, spectral centroid, roll-off frequency, zero-crossing, MFCC, delta MFCC, delta delta MFCC. The total of these handcrafted features has dimensions of 477. Meanwhile, the features obtained from transfer learning are the result of feature extraction of the VGGish pretrained model [13]. The features obtained from this pretraining model have 256 dimensions, so that the total features are 733 dimensions. After that, the model training uses 80% of the dataset and is tested on the remaining 20%.

III. Dataset

In this research, the data is obtained from 2 open source datasets, the COUGHVID dataset [3] and the Coswara dataset [9].

The first dataset was collected by researchers from the Embedded System Laboratory (ESL), cole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. This dataset was collected between April 1, 2020 to September 10, 2020 via a web application developed by the research team. This dataset consists of more than 20,000 records and 1,010 of them claim to have been exposed to COVID-19. The data collected include recordings of coughing sounds, self-reported status (COVID-19, symptomatic, healthy, none), and user metadata (age, gender, conditions experienced by the user now, geolocation). The coughing recording is mandatory, but the user metadata is optional, so it can be filled in or not by the user. A total of 1000 records were annotated by an expert pulmonologist, the annotated attributes include quality, type of cough, audible dyspnea, audible wheezing, audible stridor, audible choking, audible nasal congestion, nothing specific, impression.

The second one was collected by a group of researchers from Indian Institute of Science. The dataset includes sound data and its corresponding metadata. For sound data, they focus on nine different categories, namely, breathing (two kinds; shallow and deep), cough (two kinds; shallow and heavy), sustained vowel phonation (three kinds; /ey/ as in made, /i/ as in beet, /u:/ as in cool), and one to twenty digit counting (two kinds; normal and fast paced). For the metadata information, they collect information such as, age, gender, location (country, state/province), current health status (healthy / exposed / cured / infected) and the presence of co-morbidity (pre-existing medical conditions). No personally identifiable information is collected and it is also anonymized during storage. The whole data contains of 1,503 recordings and 108 of them are identified as positive from the disease.

IV. Methods

A. Data Preprocessing

Before going further to the classification process, we do some data preprocessing on both datasets. In the COUGHVID Dataset, the preprocessing carried out include taking the status with the labels 'COVID-19' and 'healthy', then filtering the data with a cough_detected value above 0.9 to get a record containing only cough records. After that, the other columns are dropped.

In the Coswara dataset, the data preprocessing carried out is to only take the value of ‘covid_status‘ with the label ‘healthy’ and ‘recovered_full’ to be combined into a negative label, while positive cases of COVID-19 are taken from the labels ‘positive_mild’, ‘positive_asymp’, and ‘positive_moderate’. After that, the other columns are dropped. Then, for each record in the Coswara dataset, there are 2 cough recording files taken, cough-heavy and cough-shallow.

In the last stage of data preprocessing, the data in each dataset consists of recordings of coughing sounds and their corresponding labels. The positive label COVID-19 is annotated with the value '1', while the negative label is annotated with the value '0'. After going through the data preprocessing stage, the data distribution of each dataset after preprocessed is shown in TABLE I.

TABLE I. Preprocesed Dataset

Dataset	#positive	#negative	Total
COUGHVID-19	440	4680	5120
Coswara	216	2385	2601

B. Feature Extraction

For the feature, we try 4 differents extraction methods, namely MFCC-raw, MFCC-stats, NMF-spectrogram, and NMF-MFCC.

1) MFCC-raw

In this setup we use MFCC with 13 components, its delta, and its delta-delta, resulting to 39 dimensions. After we get the MFCC, we padded it to the maximum frames in the dataset.

2) MFCC-stats

MFCC-stats are produced by taking the statistical values of MFCC-raw. We calculate the mean, median, skewness, max, min, and std of the MFCC-raw and concatenate them, resulting to 234 dimensions feature vector.

3) NMF-spectrogram

To calculate the NMF-spectrogram feature, we first have to do a fourier transform on the audio files to get the spectrum features. After that, we perform a Non-negative Matrix

Factorization on the spectrum. Then, we took the resulting non-negative matrix without the temporal values.

4) NMF-MFCC

In general, the NMF-MFCC feature use all the same step of MFCC, except it doesn't use the mel filter bank. The mel filter bank is replaced with the NMF-spectrogram. First, we took some samples of cough recording. Then we concatenate all those recording and perform the NMF-spectrogram feature extraction. After we get the NMF-spectrogram feature, we replace the mel-filter bank in the MFCC step.

C. Data Splitting

After going through the feature extraction process, the training data and test data are divided into each dataset. The ratio used in both datasets is the same, 9:1 training data and test data respectively. This division is carried out in a stratified manner so that the percentage of the amount of data in each class remains the same.

D. Imbalanced Data Handling

Based on the distribution of the two datasets which can be seen in TABLE I, both data experienced imbalanced problems. The comparison between positive and negative classes is at a ratio of 1:10. This needs to be addressed so that the classification model made does not tend to predict the majority class (negative class). In this study, oversampling & undersampling techniques were used to make the distribution of each class balanced.

The first technique is random undersampling, this technique reduces the amount of data in the majority class randomly. The second technique is random oversampling, this technique duplicates the number of minority data randomly. The last technique is the Synthetic Minority Over-Sampling Technique (SMOTE), this technique performs oversampling by synthesizing data on minority class data, so that there is no duplication of data. It should be noted that oversampling & undersampling are only performed on training data.

E. Modelling

In this research, we try 3 different shallow modelling methods, namely Support Vector Machine (SVM), K-nearest neighbour (KNN), and Extreme Gradient Boosting (XGBoost).

1) Support Vectore Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm that is quite popular to solve classification and regression problems. SVM has been used to solve problems in various machine learning domains, including computer vision [14], natural language processing [15], neuroimaging [16], and bioinformatics [17]. The main idea of SVM is to find a hyperplane that can separate data points into each class. The term "support vector" itself is the data points that are closest to the hyperplane. Therefore, the support vector plays an important role in determining the hyperplane and more specifically in determining the accurate class for the test data.

The advantages of the SVM algorithm include good accuracy and relatively short training time for small and clean datasets. The disadvantage of this algorithm is that it is not suitable for large datasets, this is because the hyperplane search problem in SVM is a quadratic minimization problem, so the training time of the SVM model will increase quadratically as the dataset size increases. In addition, SVM is also not suitable for datasets that are noisy and have intersecting classes.

2) K-nearest neighbour (KNN)

The K-nearest Neighbors (KNN) algorithm is a supervised machine learning algorithm for classification. This algorithm is a machine learning algorithm that is very simple and straight forward. The basic concept of KNN is to determine the class of a data point based on the nearest K data points around it, then the class determination is taken from the majority of the surrounding data point classes. The 'nearest' data point is determined based on the calculation of the distance between the surrounding data points and the reference data point. Distance calculations can be done using Euclidian distance, Manhattan distance, or other distance calculations. This concept is based on a very straight forward assumption, namely that adjacent data points will have the same class. KNN has advantages such as not requiring time for training, simple and straightforward, and easy to implement. Meanwhile, KNN has weaknesses, namely it cannot work well when the dimensions of the features are very high, sensitive to outliers, and computation for high predictions is required because it requires calculating the distance of all data points to reference data points.

3) Extreme Gradient Boosting (XGBoost)

The Extreme Gradient Boosting (XGBoost) algorithm is a derivative of the decision tree algorithm. This algorithm is a supervised machine learning algorithm for both classification and regression. XGBoost utilizes ensemble boosting machine learning techniques on the decision tree algorithm. In general, boosting aims to improve performance, in the case of machine learning, boosting is a sequential ensemble learning technique to transform weak models into robust models.

In XGBoost, a set of decision trees is trained sequentially. First, a decision tree is trained on the existing dataset, then based on the learning results, the next decision tree is trained. Errors that occur in the initial decision tree are minimized by the next decision tree. This is done continuously sequentially so that in the end a robust model is obtained.

The term 'gradient' in XGBoost is a special case of the boosting technique, namely by minimizing errors by utilizing the gradient descent algorithm. Meanwhile, the term 'extreme' refers to the gradient boosting model developed by [18]. This model is built with advantages, namely maximizing learning outcomes and minimizing the use of computing resources. This is what makes XGBoost widely used today.

V. Experiment

In general, the experiment will be carried out on 2 different datasets, the COUGHVID dataset [3] and the Coswara dataset [9]. The models will be trained and tested on their respective datasets. In the training of each model, 5-fold cross validation was carried out. The experimental scenario carried out in this study is a grid search. The variables in this experiment include datasets, feature extraction, handling imbalanced datasets, models and hyperparameters of each of these models. In detail, the experimental scenario can be seen in TABLE II.

TABLE II. Experiment Scenario

Variables		Value	Combinations
Dataset		COUGHVID, Coswara	2
Features		MFCC-raw, MFCC-stats, NMF-Spectrogram, NMF-MFCC	4
Imbalance Handling		*Random Oversampling, Random Undersampling,* SMOTE	3
Model	SVM Hyperparameter	C: {1, 2, 3}	3 + 4 + 9 = 16
	KNN Hyperparameter	n_neighbors: {3, 5}, p: {1, 2}	
	XGBoost Hyperparameter	n_estimators: {10, 20, 50}, max_depth: {2, 5, 10}	
Total Combination			384

TABLE III. Experiment Result for COUGHVID Dataset

Feature	Model	Imbalance Handling	Sensitivity	Specificity	AUC-ROC
MFCC-raw	KNN	Undersampling	0.568	0.630	0.599
		SMOTE	0.614	0.511	0.562
		Oversampling	0.227	0.797	0.512
	SVM	Undersampling	-	-	-
		SMOTE	-	-	-
		Oversampling	-	-	-
	XGBoost	Undersampling	0.455	0.560	0.507
		SMOTE	0.273	0.774	0.523
		Oversampling	0.295	0.722	0.509
MFCC-stats	KNN	Undersampling	0.510	0.486	0.500
		SMOTE	0.510	0.409	0.461
		Oversampling	0.240	0.738	0.491
	SVM	Undersampling	0.000	0.985	0.493
		SMOTE	0.000	1.000	0.500
		Oversampling	0.000	1.000	0.500
	XGBoost	Undersampling	0.490	0.539	0.513
		SMOTE	0.300	0.713	0.505
		Oversampling	0.460	0.558	0.509
NMF-MFCC	KNN	Undersampling	0.477	0.509	0.493
		SMOTE	0.591	0.553	0.572
		Oversampling	0.273	0.730	0.501
	SVM	Undersampling	0.977	0.019	0.498
		SMOTE	0.000	1.000	0.500
		Oversampling	0.000	1.000	0.500
	XGBoost	Undersampling	0.523	0.549	0.536
		SMOTE	0.477	0.626	0.551
		Oversampling	0.523	0.585	0.554
NMF-spectrogram	KNN	Undersampling	0.614	0.417	0.515
		SMOTE	0.773	0.368	0.570
		Oversampling	0.409	0.753	0.581
	SVM	Undersampling	0.659	0.419	0.539
		SMOTE	0.273	0.677	0.475
		Oversampling	0.432	0.662	0.547
	XGBoost	Undersampling	0.545	0.534	0.540
		SMOTE	0.545	0.596	0.571
		Oversampling	0.500	0.664	0.582

The experiment was carried out based on the experimental scenario shown in TABLE II. The metrics used in this study are sensitivity, specificity, and AUC-ROC. Detailed experimental results can be seen in TABLE III and TABLE IV. In both tables, the highest AUC-ROC values are highlighted using green. Experiments on the SVM model and with the MFCC-raw feature only run on the Coswara dataset and handling imbalanced undersampling datasets, this is due to the very long time used to train the model.

TABLE IV. Experiment Results for Coswara Dataset

Feature	Model	Imbalance Handling	Sensitivity	Specificity	AUC-ROC
MFCC-raw	KNN	Undersampling	0.545	0.556	0.551
		SMOTE	0.545	0.481	0.513
		Oversampling	0.364	0.820	0.592
	SVM	Undersampling	0.000	1.000	0.500
		SMOTE	-	-	-
		Oversampling	-	-	-
	XGBoost	Undersampling	0.636	0.527	0.582
		SMOTE	0.500	0.753	0.627
		Oversampling	0.500	0.674	0.587
MFCC-stats	KNN	Undersampling	0.591	0.657	0.624
		SMOTE	0.773	0.586	0.679
		Oversampling	0.500	0.845	0.673
	SVM	Undersampling	1.000	0.008	0.504
		SMOTE	0.000	1.000	0.500
		Oversampling	0.000	1.000	0.500
	XGBoost	Undersampling	0.682	0.640	0.661
		SMOTE	0.500	0.695	0.597
		Oversampling	0.455	0.816	0.635
NMF-MFCC	KNN	Undersampling	0.500	0.603	0.551
		SMOTE	0.500	0.690	0.595
		Oversampling	0.273	0.845	0.559
	SVM	Undersampling	1.000	0.008	0.504
		SMOTE	0.000	1.000	0.500
		Oversampling	0.000	1.000	0.500
	XGBoost	Undersampling	0.682	0.640	0.661
		SMOTE	0.545	0.753	0.649
		Oversampling	0.409	0.795	0.602
NMF-spectrogram	KNN	Undersampling	0.500	0.628	0.564
		SMOTE	0.773	0.594	0.683
		Oversampling	0.364	0.837	0.600
	SVM	Undersampling	0.909	0.556	0.733
		SMOTE	0.636	0.640	0.638
		Oversampling	0.682	0.724	0.703
	XGBoost	Undersampling	0.727	0.615	0.671
		SMOTE	0.591	0.649	0.620
		Oversampling	0.545	0.736	0.641

VI. Analysis

This chapter will provide a more detailed analysis of the experimental results. In TABLE V it can be seen that the 10 best experimental results based on the AUC-ROC value and TABLE VI are the 10 lowest.

TABLE V. Best-10 Experiment Result

Dataset	Feature	Imbalance Handling	Model	*Sensitivity*	*Specificity*	AUC-ROC
Coswara	NMF-spectrogram	Undersampling	SVM	0.909	0.556	0.733
Coswara	NMF-spectrogram	Oversampling	SVM	0.682	0.724	0.703
Coswara	NMF-spectrogram	SMOTE	KNN	0.773	0.594	0.683
Coswara	MFCC-stats	SMOTE	KNN	0.773	0.586	0.679
Coswara	MFCC-stats	Oversampling	KNN	0.500	0.845	0.673
Coswara	NMF-spectrogram	Undersampling	XGB	0.727	0.615	0.671
Coswara	MFCC-stats	Undersampling	XGB	0.682	0.640	0.661
Coswara	NMF-MFCC	Undersampling	XGB	0.682	0.640	0.661
Coswara	NMF-MFCC-stats	SMOTE	XGB	0.545	0.753	0.649
Coswara	NMF-spectrogram	Oversampling	XGB	0.545	0.736	0.641

The ten best consistent results were obtained from experiments on the Coswara dataset, while on the other hand the seven lowest values came from the COUGHVID dataset. The boxplot in Fig. 1 shows the same trend, the median, minimum, and maximum values of the AUC-ROC dataset Coswara are higher than those produced by the COUGHVID dataset. Based on these two facts, it can be seen that the Coswara dataset produces a better model than the

COUGHVID dataset. This is supported by direct observation of the recorded coughing data for each of these datasets. After listening to several voice recording samples, the COUGHVID dataset found that there were still some voice samples other than cough mixed with coughing. Meanwhile, in the Coswara dataset, although there were also some recordings containing a mixture of coughing and non-coughing sounds, the number was much lower than the COUGHVID dataset. Based on this, it can be said that the recorded files in the Coswara dataset are cleaner than the COUGHVID dataset.

TABLE VI. Worst-10 Experiment Result

Dataset	Feature	Imbalance Handling	Model	*Sensitivity*	*Specificity*	AUC-ROC
Coswara	MFCC-raw	Undersampling	SVM	0.000	1.000	0.500
Coswara	NMF-MFCC	Oversampling	SVM	0.000	1.000	0.500
Coswara	NMF-MFCC	SMOTE	SVM	0.000	1.000	0.500
COUGHVID	MFCC-stats	Undersampling	KNN	0.510	0.486	0.500
COUGHVID	NMF-MFCC	Undersampling	SVM	0.977	0.019	0.498
COUGHVID	NMF-MFCC	Undersampling	KNN	0.477	0.509	0.493
COUGHVID	MFCC-stats	Undersampling	SVM	0.000	0.985	0.493
COUGHVID	MFCC-stats	Oversampling	KNN	0.240	0.738	0.491
COUGHVID	NMF-spectrogram	SMOTE	SVM	0.273	0.677	0.475
COUGHVID	MFCC-stats	SMOTE	KNN	0.510	0.409	0.461

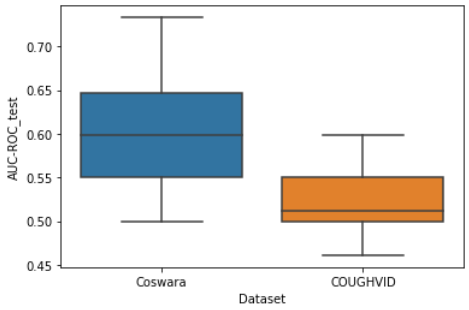

Fig. 1. AUC-ROC Score on Different Dataset

The next analysis is reviewed in terms of feature extraction. In TABLE V, 5 out of 10 best results were obtained using the NMF-spectrogram extraction feature. If seen in Fig. 2. AUC-ROC Score on Different Feature Extraction, it is also consistent that the box value in the boxplot for NMF-spectrogram feature extraction is above other feature extractions. This shows that this feature produces a better AUC-ROC value than other features.

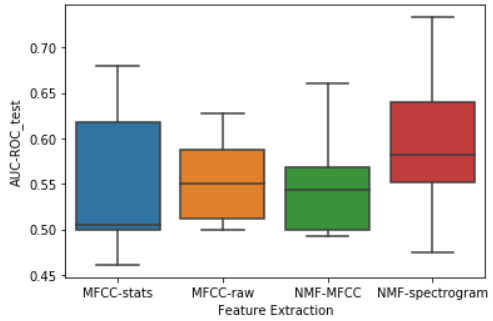

Fig. 2. AUC-ROC Score on Different Feature Extraction

In handling imbalanced datasets, the three techniques did not produce significant differences in values. If seen in Fig. 3, the values in the boxplot tend to be the same. However, considering the memory, complexity and computational time, the random undersampling technique is certainly superior. This is because the size of the data for the application of the random undersampling technique on imbalanced datasets is much smaller than the random oversampling or SMOTE techniques. For example, if the ratio of the number of classes is 1:10, the random undersampling technique will produce 2 data while the oversampling technique will produce 20 data. From the computational point of view, the imbalanced dataset technique itself, theoretically, random technique is cheaper than synthetic techniques such as SMOTE. So based on these facts it can be said that the random undersampling technique is preferable in this case.

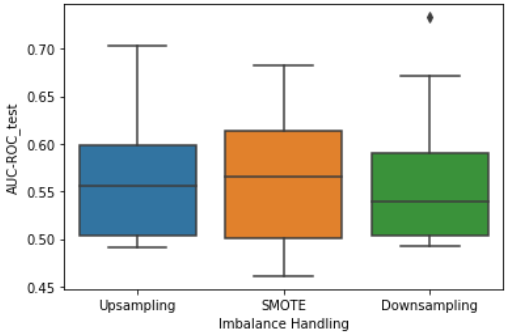

Fig. 3. AUC-ROC Score on Different Imbalanced Dataset Handling

The next analysis is in terms of modeling techniques. Based on Fig. 4 it can be seen that the XGBoost model is superior to the other two models. Meanwhile, SVM is far below with several outliers that produce AUC-ROC values that are superior to the KNN and XGBoost modeling techniques.

Table VII. Experiment Result Using SVM

No	Dataset	Feature	Imbalance Handling	Sensitivity	Specificity	AUC-ROC
1	Coswara	MFCC-raw	Undersampling	0.000	1.000	0.500
2		MFCC-stats	Undersampling	1.000	0.008	0.504
3			SMOTE	0.000	1.000	0.500
4			Oversampling	0.000	1.000	0.500
5		NMF-MFCC	Undersampling	1.000	0.008	0.504
6			SMOTE	0.000	1.000	0.500
7			Oversampling	0.000	1.000	0.500
8		NMF-spectrogram	Undersampling	0.909	0.556	0.733
9			SMOTE	0.636	0.640	0.638
10			Oversampling	0.682	0.724	0.703
11	COUGHVID	MFCC-stats	Undersampling	0.000	0.985	0.493
12			SMOTE	0.000	1.000	0.500
13			Oversampling	0.000	1.000	0.500
14		NMF-MFCC	Undersampling	0.977	0.019	0.498
15			SMOTE	0.000	1.000	0.500
16			Oversampling	0.000	1.000	0.500
17		NMF-spectrogram	Undersampling	0.659	0.419	0.539
18			SMOTE	0.273	0.677	0.475
19			Oversampling	0.432	0.662	0.547

shows the experimental results with the SVM modeling technique only. It can be seen in the table that SVM is often difficult to study the data. This can be seen from the extreme sensitivity and specificity values that are inversely proportional. This shows that the model tends to predict only one class and ignores other classes, of course this is not desirable. However, if we look deeper, this only happens when MFCC based features are used. This shows that MFCC-based features cannot work well with the SVM model.

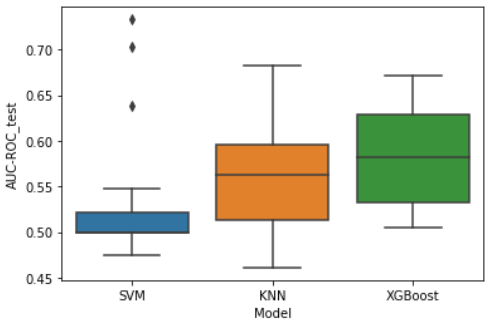

Fig. 4. AUC-ROC Score on Different Model

VII. Conclusions

In this paper, we performed multiple feature extraction, modelling technique, and imbalance data handling for COVID-19 classification system using cough recording. It can be concluded that the use of the Non-negative Matrix Factorization feature extraction technique on the spectrogram improves the classification performance. Extreme Gradient Boosting (XGBoost) modeling technique tends to produce better performance than Support Vector Machine (SVM) and K-nearest neighbor (KNN). The technique for handling imbalanced dataset random undersampling is preferable than the random oversampling technique or the Synthetic Minority Over-Sampling Technique (SMOTE) in terms of memory, complexity, and computation time. Cleanliness of voice recording data has a big effect on classification performance. It is shown that the dataset containing clean sound recordings has better performance. In this study, the best performance was obtained with a combination of NMF-spectrogram extraction features, random undersampling, and SVM on a dataset that selected a clean cough sound recording (Coswara dataset). However, it should be noted that the SVM modeling technique does not work well with features based on Mel Frequency Cepstrum Coefficients (MFCC).

References

[1] C. Brown *et al.*, "Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data," 2020, doi: 10.1145/3394486.3412865.

[2] J. Laguarta, F. Hueto, and B. Subirana, "COVID-19 artificial intelligence diagnosis using only cough recordings," *IEEE Open J. Eng. Med. Biol.*, 2020, doi: 10.1109/ojemb.2020.3026928.

[3] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset: a corpus for the study of large-scale cough analysis algorithms," *arXiv*. 2020.

[4] A. Pal and M. Sankarasubbu, "Pay Attention to the cough: early diagnosis of COVID-19 using interpretable symptoms embeddings with cough sound signal processing," *arXiv*. 2020.

[5] P. Bagad *et al.*, "Cough against COVID: evidence of COVID-19 signature in cough sounds," *arXiv Prepr. arXiv2009.08790*, 2020, [Online]. Available: http://arxiv.org/abs/2009.08790.

[6] K. J. Piczak, "ESC: dataset for environmental sound classification," 2015, doi: 10.1145/2733373.2806390.

[7] D. S. Park *et al.*, "Specaugment: a simple data augmentation method for automatic speech recognition," 2019, doi: 10.21437/Interspeech.2019-2680.

[8] F. Al Hossain, A. A. Lover, G. A. Corey, N. G. Reich, and T. Rahman, "FluSense: a contactless syndromic surveillance platform for influenza-like illness in hospital waiting areas," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, 2020, doi: 10.1145/3381014.

[9] N. Sharma *et al.*, "Coswara - a database of breathing, cough, and voice sounds for COVID-19 diagnosis," 2020, doi: 10.21437/Interspeech.2020-2768.

[10] E. Fonseca *et al.*, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," *arXiv*. 2018.

[11] J. Watson, P. F. Whiting, and J. E. Brush, "Interpreting a covid-19 test result," *The BMJ*. 2020, doi: 10.1136/bmj.m1808.

[12] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?," *arXiv*. 2019.

[13] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," 2017, doi: 10.1109/ICASSP.2017.7952132.

[14] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, vol. II, doi: 10.1109/ICCV.2005.239.

[15] J. Nivre *et al.*, "MaltParser: a language-independent system for data-driven dependency parsing," *Nat. Lang. Eng.*, vol. 13, no. 2, 2007, doi: 10.1017/S1351324906004505.

[16] M. Hanke, Y. O. Halchenko, P. B. Sederberg, S. J. Hanson, J. V. Haxby, and S. Pollmann, "PyMVPA: a python toolbox for multivariate pattern analysis of fMRI data," *Neuroinformatics*, vol. 7, no. 1, 2009, doi: 10.1007/s12021-008-9041-y.

[17] K. C. Dorff, N. Chambwe, M. Srdanovic, and F. Campagne, "BDVal: reproducible large-scale predictive model development and validation in high-throughput datasets," *Bioinformatics*, vol. 26, no. 19, 2010, doi: 10.1093/bioinformatics/btq463.

[18] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, vol. 13-17-August-2016, doi: 10.1145/2939672.2939785.