Indonesian Journal of Electrical Engineering and Computer Science
Vol. 38, No. 2, May 2025, pp. 940~949
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v38.i2.pp940-949

Detection of COVID-19 based on cough sound and accompanying symptom using LightGBM algorithm

Wiharto, Annas Abdurrahman, Umi Salamah
Department of Informatics, Faculty of Information Technology and Data Science, Universitas Sebelas Maret, Surakarta, Indonesia

Article Info

Article history:

Received Mar 19, 2024
Revised Oct 27, 2024
Accepted Nov 11, 2024

Keywords:

Coronavirus disease 19
Cough
LightGBM
Symptom

ABSTRACT

Coronavirus disease 19 (COVID-19) is an infectious disease whose diagnosis is carried out using antigen-antibody tests and reverse transcription polymerase chain reaction (RT-PCR). Apart from these two methods, several alternative early detection methods using machine learning have been developed. However, it still has limitations in accessibility, is invasive, and its implementation involves many parties, which could potentially even increase the risk of spreading COVID-19. Therefore, this research aims to develop an alternative early detection method that is non-invasive by utilizing the LightGBM algorithm to detect COVID-19 based on the results of feature extraction from cough sounds and accompanying symptoms that can be identified independently. This research uses cough sound samples and symptom data from the Coswara dataset, and cough sound's features were extracted using the log mel-spectrogram, mel frequency cepstrum coefficient (MFCC), chroma, zero crossing rate (ZCR), and root mean square (RMS) methods. Next, the cough sound features are combined with symptom data to train the LightGBM. The model trained using cough sound features and patient symptoms obtained the best performance with 95.61% accuracy, 93.33% area under curve (AUC), 88.74% sensitivity, 97.91% specificity, 93.17% positive prediction value (PPV), and 96.33% negative prediction value (NPV). It can be concluded that the trained model has excellent classification capabilities based on the AUC values obtained.

Corresponding Author:

Wiharto
Department of Informatics, Faculty of Information Technology and Data Science
Universiti Sebelas Maret
St. Ir. Sutami, No. 36A, Kentingan, Surakarta, Indonesia
Email: wiharto@staff.uns.ac.id

1. INTRODUCTION

Coronavirus disease 19 (COVID-19) is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and was first detected in the city of Wuhan, China, at the end of 2019. The symptoms vary, ranging from respiratory infection-like symptoms such as fever, cough, sore throat, stuffy nose, headache, muscle pain, and malaise to severe cases that can lead to pneumonia and death. Some symptoms can also occur but rarely, such as diarrhea and anosmia [1]. Clinically diagnosis of COVID-19 uses antigen-antibody and reverse transcription polymerase chain reaction (RT-PCR), with RT-PCR as the gold standard [1]. Two different methods have lately been put forth in the literature for the diagnosis of COVID-19 infection using computer temography (CT) or X-ray image analysis [2]. Unfortunately, this alternative method still has problems in terms of accessibility and has the potential to increase the risk of spreading COVID-19 because implementation involves many parties.

Several studies have also tried to use the sound of coughing to detect COVID-19. COVID-19 has symptoms of a dry cough with characteristics of higher frequency and shorter duration compared to coughs from other respiratory diseases [3]. The cough sound of COVID-19 also has different latent features and the risk of latent feature overlap with the feature of cough sound of other diseases is low. This difference occurs because COVID-19 infection affects the respiratory system uniquely than the other [2].

Various studies have explored different methods to extract features from cough sounds and patient symptoms for COVID-19 detection. In the work Chowdhury *et al.* [4], extra-trees trained on combined datasets achieved an area under curve (AUC) of 0.79, while HGBoost on the Coswara dataset reached an AUC of 0.66. Another study combined datasets from multiple sources, achieving an accuracy of 0.921 and an AUC of 0.973 with a VGGNet model. Unfortunately, in that study when using only the Coswara dataset, the model produced performance, for accuracy parameters of 0.712 and AUC of 0.781 [5]. This performance is still relatively low, if referring to the AUC value then the model is only included in the fair category.

Some studies have also explored a combination of cough sound features with symptom and respiratory condition data. The work Rahouma *et al.* [6] utilized patient voices and symptom data from the Coswara dataset. The neural network trained with cough sound alone achieved an average accuracy of 84% and an AUC of 82%. Using symptom data alone resulted in an average accuracy of 73% and an AUC of 78%. Combining both features yielded an average accuracy of 91% and an AUC of 88%. One study on the Coughvid dataset using a multi-branch deep learning network (MBDLN) achieved an AUC of 91% [7]. Lastly, a hierarchical multi-modal transformer (HMT) trained on symptom data and cough sound features from the Coughvid and Coswara datasets, achieved an average accuracy of 81.32% and an AUC of 82.06% [8].

Previous research that combined cough sound and symptom features was able to provide better performance, compared to using only cough sound features. Unfortunately, previous studies used symptoms generated from expert examination. This was shown in a study by [6], which used pneumonia and asthma symptom data, where these symptoms require expert diagnosis [9]. The same thing was also done in research conducted by [7], [8] using respiratory condition data from expert diagnosis [10]. Referring to a number of studies that have been conducted, it shows that the COVID-19 examination cannot be done independently. This is because it still requires the help of a doctor, namely when identifying symptoms, such as pneumonia, asthma, and respiratory conditions. If only relying on the cough sound feature, the model cannot produce optimal performance.

Referring to a number of previous studies, this study proposes a COVID-19 detection model that can be carried out independently, using the LightGBM classification algorithm. This model uses cough sound features and symptoms. The symptoms used are symptoms that can be recognized independently, so they do not require examination by a doctor. The model was developed using the Coswara dataset, with the symptoms used being difficulty breathing, runny nose, cough, fever, anosmia, muscle pain, sore throat, diarrhea, and fatigue. The performance of the proposed model is measured using the performance parameters of accuracy, sensitivity, specificity, AUC, positive prediction value (PPV), and negative prediction value (NPV).

2. METHOD

The work stages in this study start from the preparation of datasets and segmentation of cough sounds, data preprocessing, training three machine learning models trained with different feature subsets, and end with evaluating the performance of the three models that have been trained. The stages of the research process can be seen in Figure 1.

2.1. Preparing the dataset and cough segmentation

The dataset used comes from the Coswara dataset. It contains recordings of cough sounds, breathing, and pronunciation of some letters or phrases from volunteers, along with metadata containing information about clinical symptoms and health history [11]. Even though the volunteer submitted various types of sounds, this research only used cough sound because of its consistency across individuals, it's not affected by accent and the sharp sound is easily differentiated from other sounds [4].

To overcome the imbalance in the amount of data in the Coswara dataset, only a few relevant data labels were used and combined into two main classes, namely "negative" (0) class which includes data with the label "healthy" and the "positive" class (1) which includes data labeled as "positive_mild" and "positive_moderate". Each volunteer submitted two types of cough sound samples, "heavy_cough" and "shallow_cough", but only the heavy cough sounds were used in this study to ensure consistency [4]. The selected symptoms include difficulty breathing, cold, cough, fever, anosmia, muscle pain, sore throat, diarrhea, and fatigue. The selection of these 9 symptoms was made by excluding asthma, diabetes, ischemic heart disease, chronic obstructive pulmonary disease, and pneumonia, due to complex diagnostic methods involving various medical assessments [9], [12]–[15].

Cough sound segmentation was performed on all samples to create a new dataset consisting of a single cough sound. Segmentation will enhance consistency, ensure complete cough sounds, focus on relevant features, and increase training samples by using the hysteresis comparator method proposed by [10]. The result is a new dataset consisting of cough sound segment audio files, metadata with sample IDs, file locations, symptom data, and labels.

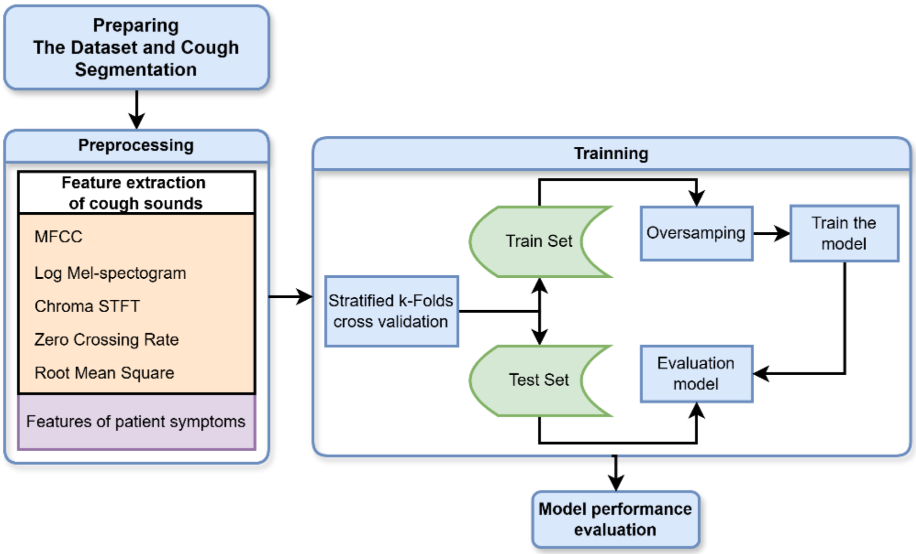

Figure 1. Stages of research

2.2. Preprocessing

At this stage, an array that will be used for the model input is formed. It will include cough sound features obtained through audio feature extraction using five methods: Log mel-spectrogram, mel frequency cepstrum coefficient (MFCC), chroma short time fourier transform (STFT), zero crossing rate (ZCR), and root mean square (RMS). Log mel-spectrogram represents the energy of the audio signal in the frequency and time domain. This is done by transforming the signal into the frequency domain using STFT, then converting the frequency into the Mel scale that is more suitable for human hearing by changing the frequency value f using (1).

$$Mel(f) = 2595 \times log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

Finally, apply logarithm scale of the energy to consider the logarithmic perception of hearing. The result is a matrix of energy spectrum in the mel scale for each time frame [16]. MFCC is intended to replicate human hearing characteristics. This involves transforming the signal into the frequency domain using STFT, converting the frequency into the mel scale using (1), taking the logarithm of the sound intensity, and applying the discrete cosine transform to generate cepstral coefficients, yielding a cepstral coefficient matrix [17].

Chroma feature divides the audio signal into chroma and pitch, mapping the frequency from the STFT into the chroma scale, and producing a vector of 12 chroma values representing 12 basic tones [18]. RMS is a simple feature that provides information about the strength or intensity of sound over a period of time [19]. RMS can be obtained using (2).

$$RMS = \sqrt{\frac{1}{n}\sum_n |x(n)|^2} \tag{2}$$

ZCR gives a rough estimate of the dominant frequency in the audio signal by counting how many times the sound amplitude crosses zero within a specific time [18]. The extraction results for each feature are processed by calculating the mean value for all frames for each feature coefficient or feature type. Next, the selected patient symptom data for each sample is converted into binary representation and added to the dataset.

2.3. Training

The next stage is to train the machine learning model using the stratified k-fold cross-validation (SKCV) method. SKCV will divide the dataset into k subsets called folds which size and data distribution are the same. Fold division is based on ID to avoid duplicate cough samples in training and testing subsets with the same ID. In every training process, one-fold will be selected as validation data and the rest will be used as data to train the model. Then the model performance is calculated from validation data [20].

Three models are trained with different feature subsets: cough sound (model 1), symptom data (model 2), and combination of both features (model 3). In each fold iteration, the training data will be oversampled first using support vector machine-synthetic minority oversampling technique (SVM-SMOTE) to handle class imbalance which combines SVM and SMOTE to create synthetic samples [21]. The oversampled data is then used to train LightGBM. LightGBM is a gradient boosting decision tree (GBDT) algorithm that implements gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). GOSS optimizes training efficiency by focusing on training models on data with large gradients and sampling data with small gradients. EFB will reduce feature complexity by combining mutually exclusive features [22]. Initially, LightGBM applies EFB to train data. Model is initialized with initial predictions minimizing loss. During GBDT implementation, LightGBM uses GOSS to resample the dataset. Information gain is calculated for features in the resampled dataset to build a new decision tree. A new decision tree is built on resampled data, updating the model every iteration. The ensemble of decision trees will form the final model [23].

2.4. Evaluation

Trained model performance is evaluated by averaging accuracy metrics like AUC, sensitivity, specificity, PPV, and NPV across all folds. The confusion matrix results from each fold are combined to assess overall performance. The confusion matrix summarizes the model performance by comparing the predicted labels with the actual labels from the dataset. It consists of four parts: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). This method will be used to calculate accuracy, sensitivity, specificity, PPV, and NPV in (3) to (7) [24]. The representation of the confusion matrix can be seen in Table 1. Accuracy measures how well the model can correctly predict results from all data.

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \tag{3}$$

AUC shows the model’s ability to differentiate between two different classes correctly. AUC measures the area under the receiver operating characteristic (ROC) curve, where a value of 0 indicates poor performance, a value of 1 indicates perfect performance and a value of 0.5 indicates random performance. Sensitivity, also known as recall, functions to measure how well the model can correctly identify positive cases.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

Specificity will measure how well the model can identify negative cases.

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

PPV also known as precision measures how well the model predicts the correct positive cases from all positive prediction results.

$$PPV = \frac{TP}{TP + FP} \tag{6}$$

NPVmeasures how well the model is in predicting the correct negative case from all negative predicted results.

$$NPV = \frac{TN}{TN + FN} \tag{7}$$

Table 1. Confusion matrix

True class	Predicted class	
	Negative	Positive
Negative	TN	FP
Positive	FN	TP

After performance evaluation, feature importance analysis is conducted using split and gain to assess the contribution of each feature to predictions. Gain indicates accuracy improvement, while split indicates feature usage in decision tree nodes [25]. This analysis is applied to model 3, so we can understand the relative contribution of the two types of features to the model's prediction performance.

3. RESULTS AND DISCUSSION

3.1. Preparing the dataset and cough segmentation

Table 2 contains the distribution of samples for each class after forming the label "healthy" as the negative class and the combination of samples labeled "positive_mild" and "positive_moderate" as data for the positive class. Then segmentation is performed on each sample using the hysteresis comparator method. Parameters, such as min_cough and cough_padding, were left with a default value of 0.2 as proposed in previous research [10]. Sometimes, not all samples were detected to have cough segments, either because there were none or the segmentation algorithm failed. A comparison of successful and failed cough segmentation results can be seen in Figure 2.

The possible cause is that the RMS value is too high in some samples which causes a weak cough sound not to be detected. Another influencing factor is the choice of parameter values, such as min_cough_len, where cough sounds that are too short may not be detected. After all recording samples were segmented, we obtained a dataset containing 4,411 segments originating from 1,579 samples with details as in Table 3.

Table 2. Sample distribution after label merging

Label	Gender	Number of samples	Total sample
0 (negative)	Male	1,068	1,432
	Female	364	
1 (positive)	Male	362	591
	Female	229	

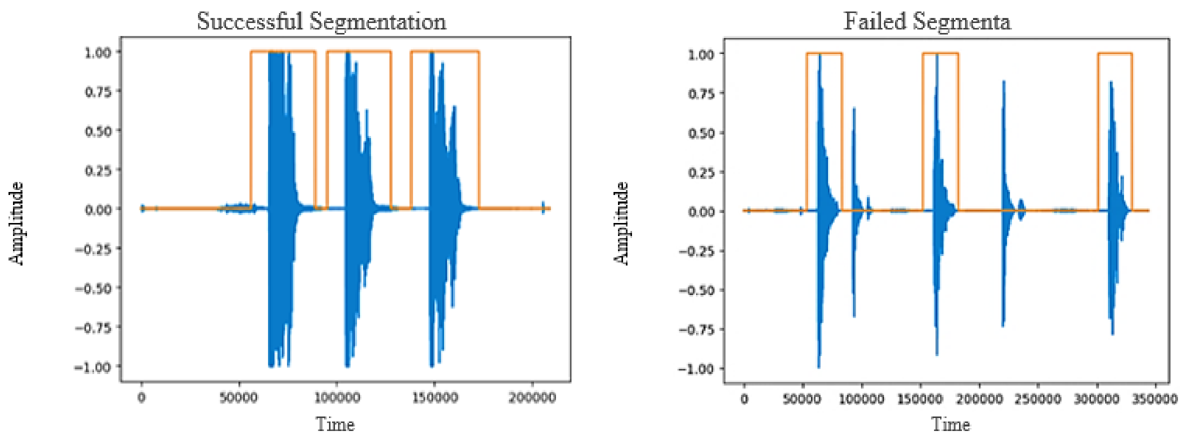

Figure 2. Example of cough sound segmentation results

Table 3. Distribution of cough sound segmentation results

Label	Gender	Number of segments	Total segments
0 (negative)	Male	2,658	3,336
	Female	678	
1 (positive)	Male	710	1,075
	Female	365	

3.2. Preprocessing

A library called librosa is used to extract features from cough segments. Parameters like n_fft, hop_length, and window are set to 2,048, 512, and 'hann' respectively for balanced frequency-time resolution [26]. For each feature extraction method, n_mfcc in MFCC is set to 39 based on [4], and parameters in log mel-spectrogram and chroma STFT are left with default values. While RMS and ZCR do not have specific parameters.

After the feature extraction process is complete, we calculate the average of each feature value across all frames. Thus, the results of the MFCC method have 39 features, log mel-spectrogram has 128 features, chroma STFT has 12 features, ZCR has 1 feature, RMS has 1 feature, and the total features of each cough sound are 181. Finally, feature extraction results from the cough sound are combined with the symptom data which has been converted into binary form, so that each segment has features with dimensions of 190. The final result is an array with dimensions of 192×4411 after adding the ID and label of the original sample for each segment.

3.3. Training

Three LightGBM models were trained on different feature subsets and evaluated for their performance using the stratified k-fold cross-validation method by dividing all sample IDs into five folds. Thus, each fold contains cough segments sourced from around 315 samples. In each fold iteration, the training data is oversampled using SVM-SMOTE from a library called imbalanced-learn [27]. We set k_neighbors parameter value in SVM-SMOTE to 1 because experiments with k-neighbor's values 1–5 show that 1 yield better performance, so synthetic samples are generated based on one nearest neighbor. Then, LightGBM models were trained using oversampled training data with hyperparameter settings from research results by [28] as in Table 4. The training process uses five folds so it will provide performance metrics for each model five times. The average model performance value is calculated, and the confusion matrix is summated.

Table 4. Distribution of cough sound segmentation results

Hyperparameter	Definition	Value
learning_rate	How much the model weights change each time the model is updated	0.03
colsample_bytree	Number of feature subsets at each iteration	0.28
subsample	Number of data subsets in each iteration	0.68
reg_alpha	L1 regularization	1
reg_lambda	L2 regularization	2
num_leaves	Maximum number of leaves in one tree	500
num_boost_round	Maximum number of boosting iterations	10,000
early_stopping_round	The maximum number of iterations to stop training if there is no increase in the performance metric value	100

3.4. Evaluation

After the training process is finished, all the performance metric values as well as the confusion matrix combined from five training iterations are obtained from three models trained with different feature subsets: cough sound model (model 1), symptom model (model 2), and combination model (model 3). The summed confusion matrix from all folds for all models can be seen in Figure 3 and all the performance metric values for all models can be seen in Table 5. Based on the confusion matrix of all models, we can see that model 1 which is trained based on cough sound feature extraction has a high true negative rate but also has high FN prediction. Model 2 which is trained with symptom data showed lower FN and higher TP rates. Model 3, which is combined cough sound features and symptom data showed the best performance, with the lowest FN rate and highest TP rate.

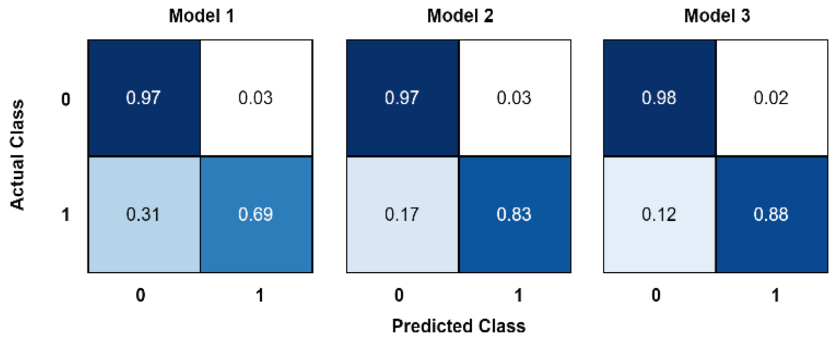

Figure 3. Summary of the confusion matrix for all models

From average performance of the three models in Table 6, can be drawn to the same conclusion as the analysis based on the confusion matrix that models trained with patient symptom data tend to perform better than models trained using cough sound features. Patient symptoms are categorical and directly indicate COVID-19 presence, while cough sound features have a wider range, making it harder for the model to find consistent patterns. However, combining both features improves the model's predictive ability, with improvements in several performance metrics. The model's accuracy, AUC, sensitivity, and specificity increased to 95.61%, 93.33%, 88.74%, and 97.91%, respectively. This increased sensitivity and specificity have positive implications for detecting COVID-19 cases and reducing prediction errors. The model also showed a higher prediction probability (PPV) value of 93.17% and NPV of 96.33%, indicating better prediction accuracy.

Table 5. Performance metric values for all models

	Fold	Accuracy	AUC	Sensitivity	Specificity	PPV	NPV
Model 1	1	0.898	0.833	0.711	0.954	0.822	0.917
	2	0.913	0.836	0.683	0.988	0.949	0.905
	3	0.888	0.831	0.724	0.937	0.779	0.918
	4	0.881	0.813	0.668	0.958	0.853	0.888
	5	0.910	0.824	0.657	0.991	0.958	0.901
	Mean	0.898	0.827	0.689	0.966	0.872	0.906
Model 2	1	0.945	0.917	0.860	0.973	0.911	0.955
	2	0.945	0.904	0.814	0.994	0.980	0.936
	3	0.936	0.907	0.858	0.957	0.840	0.962
	4	0.942	0.883	0.776	0.990	0.956	0.939
	5	0.911	0.887	0.832	0.942	0.843	0.937
	Mean	0.936	0.899	0.828	0.971	0.906	0.946
Model 3	1	0.916	0.875	0.794	0.956	0.852	0.935
	2	0.976	0.951	0.907	0.996	0.983	0.975
	3	0.965	0.955	0.936	0.974	0.913	0.981
	4	0.971	0.959	0.936	0.982	0.940	0.981
	5	0.952	0.927	0.865	0.989	0.970	0.945
	Mean	0.956	0.933	0.887	0.979	0.937	0.963

Table 6. Comparison with related research

Research	Dataset	Voice data	Symptom data	Model	AUC (%)
[4]	Coswara	Cough sound	-	HGBoost	66
[5]	Coswara	Cough sound	-	Mini VGGNet	78,1
[6]	Coswara	Patient's voice	-	Neural network	82
	Coswara	-	Pneumonia, asthma, difficulty breathing, diarrhea, fatigue, muscle aches, fever, cold, and sore throat		78
	Coswara	Patient's voice	Pneumonia, asthma, difficulty breathing, diarrhea, fatigue, muscle aches, fever, cold, and sore throat		88
[7]	Coughvid	Cough sound	Respiratory conditions and fever symptoms	MBDLN	91
[8]	Coswara and Coughvid	Cough sound	Respiratory conditions, symptoms of fever, and muscle pain	HMT	82
Proposed	Coswara	Cough sound	-	LightGBM	82,7
		-	Difficulty breathing, cold, cough, fever, anosmia, muscle aches, sore throat, diarrhea, and fatigue		89,9
		Cough sound	Difficulty breathing, cold, cough, fever, anosmia, muscle aches, sore throat, diarrhea, and fatigue		93

The feature importance graph analysis of the 5th iteration of model 3 training, as shown in Figure 4 can provide complementary insights. The features 'cough', 'fatigue', 'cold', and 'fever' have the highest gain, indicating their effectiveness in increasing model accuracy. Audio features such as 'zcr' and 'mfcc_7' also have higher gain values, indicating their importance in training a model to detect COVID-19 in terms of cough sound characteristics. On the other hand, feature importance ranking based on split shows that features like 'zcr', 'mfcc_12', 'mfcc_18', and 'cold' have the greatest influence in making decisions in the COVID-19 detection model. The split value indicates how often the feature is used in data splitting during the training process.

A comparison between feature importance based on gain and split shows similarities in features like 'cold', which are important for dividing data and increasing model accuracy. However, there are differences in features, such as 'zcr', which has the largest role based on split but has a lower gain than other features.

As for the related research that has been listed and explained previously, two studies only use cough sounds, the work of [4] and [5]. Three studies combine cough sounds with symptom data, research by [6]–[8].

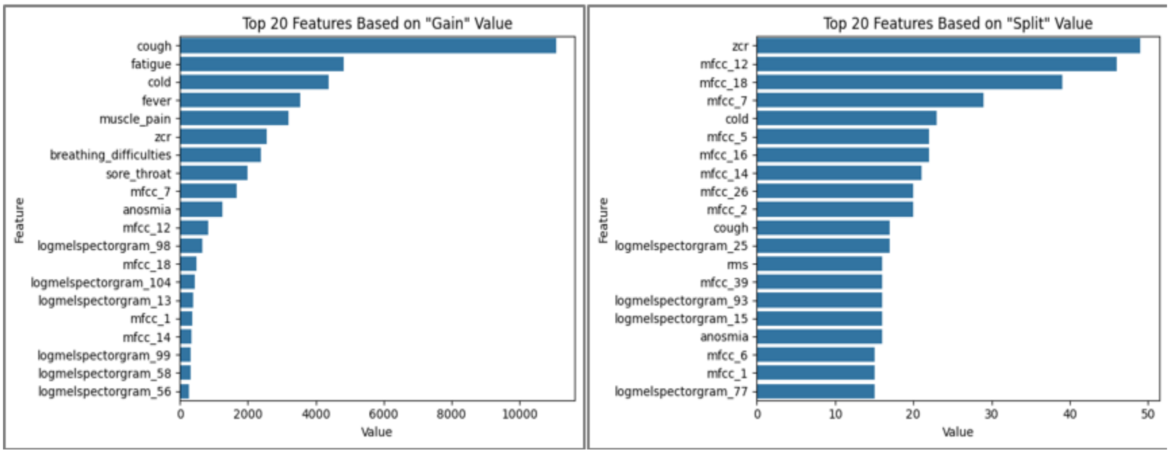

Figure 4. Feature importance analysis based on gain and split

Research [4] and [5] used cough sounds in the Coswara dataset to extract features using various methods. Chowdury *et al.* [4] used 5 different feature extraction methods to train HGBoost model and achieved an AUC value of 66%. Kho *et al.* [5] used MFCC feature to train Mini VGGNet and an AUC value of 78.1%. Both models were defeated by model 1, which had an AUC value of 82.7%. Fakhry *et al*. [7] used the augmented Coughvid dataset to train a MBDLN using MFCC features and spectrograms from cough sounds with additional symptom data, including respiratory conditions and fever symptoms. The model consisted of two deep neural network branches for MFCC features, another for symptom features, and a convolutional neural network (CNN) with residual network (ResNet)-50 architecture for spectrograms. The model can achieve an AUC value of 91%. Tang *et al.* [8] developed a HMT consisting of three branches: two multilayer perceptrons (MLPs) for processing symptom data and the average value of each MFCC feature coefficient of cough sounds, and a nested hierarchical transformer branch for extracting spectrogram features. The HMT model achieved an AUC value of 82%. Those two models can be outperformed by model 3 and it offers independent symptom identification, unlike previous research [7] and [8] that relied on expert doctors diagnosing respiratory condition abnormalities from cough recordings.

Meanwhile, Rahouma *et al.* [6] conducted a study using Coswara dataset to extract features from patient's voice, such as cough, breath, counting sounds, and English vowel pronunciation. They used 11 methods and 9 symptoms data, including asthma and pneumonia which require expert medical diagnosis. The proposed method only uses cough sounds, extracting features using only 5 methods and adding 9 symptoms data without including asthma and pneumonia or other symptoms that require expert medical diagnosis. However, the combined model in [6] had an AUC value of 88% that can be outperformed by the performance of model 3. The model trained using various types of sounds had an AUC value almost the same as model 1, with an AUC value of 82%. The model trained with symptom data obtained an AUC value of 78% and can be outperformed by model 2 with an AUC value of 89.9%. The proposed model has good classification capabilities, with a high AUC value indicating better performance in differentiating between positive and negative classes. Additionally, the proposed method has added value by using symptom data that can be identified independently.

4. CONCLUSION

From the research that has been conducted, it can be concluded that machine learning models trained only with patient symptom data perform better than models trained only with cough sound features. However, combining cough sound features and patient symptoms can improve model performance compared to just using one type of feature. The model trained using cough sound features and patient symptoms achieved an accuracy of 95.61%, AUC of 93.33%, sensitivity of 88.74%, specificity of 97.91%, PPV of 93.17%, and NPV of 96.33%. Feature importance analysis confirmed the importance of cough, fatigue, cold, and fever in improving the model’s accuracy. In addition, ZCR and MFCC are audio features that are often used to separate data during training. Overall, the proposed combined model was able to exceed the performance of related studies based on the AUC metric value of 93%, and it demonstrated excellent classification capabilities.

ACKNOWLEDGEMENTS

This work is partially supported by Universitas Sebelas Maret for providing research funding through the Hibah Riset Group scheme as stipulated in contract No. 194.2/UN27.22/PT.01.03/2024. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] I. Salahshoori *et al.*, "Overview of COVID-19 disease: virology, epidemiology, prevention diagnosis, treatment, and vaccines," *Biologics*, vol. 1, no. 1, pp. 2–40, 2021, doi: 10.3390/biologics1010002.

[2] A. Imran *et al.*, "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app," *Informatics in Medicine Unlocked*, vol. 20, 2020, doi: 10.1016/j.imu.2020.100378.

[3] Y. Bhanusree, B. S. Deeksha, V. Sohan, D. Ajay, and M. E. S. Ram, "A deep learning approach to predict to predict COVID-19 through cough analysis using CNN-BiDirectional LSTM," 2022, doi: 10.1109/ICONAT53423.2022.9726067.

[4] N. K. Chowdhury, M. A. Kabir, M. M. Rahman, and S. M. S. Islam, "Machine learning for detecting COVID-19 from cough sounds: an ensemble-based MCDM method," *Computers in Biology and Medicine*, vol. 145, 2022, doi: 10.1016/j.compbiomed.2022.105405.

[5] S. J. Kho *et al.*, "Malaysian cough sound analysis and COVID-19 classification with deep learning," *Intelligence-Based Medicine*, vol. 9, 2024, doi: 10.1016/j.ibmed.2023.100129.

[6] K. Rahouma, S. Ramzy, and M. Aly, "Artificial intelligence against virus changes: a long term detector of COVID-19 using the clinical symptoms and respiratory sounds," *Journal of Advanced Engineering Trends*, vol. 42, no. 2, pp. 289–299, 2023, doi: 10.21608/jaet.2022.132552.1149.

[7] A. Fakhry, X. Jiang, J. Xiao, G. Chaudhari, A. Han, and A. Khanzada, "Virufy: a multi-branch deep learning network for automated detection of COVID-19," 2021.

[8] S. Tang, X. Hu, L. Atlas, A. Khanzada, and M. Pilanci, "Hierarchical multi-modal transformer for automatic detection of COVID-19," in *ACM International Conference Proceeding Series*, 2022, pp. 197–202, doi: 10.1145/3556384.3556414.

[9] Y. Marushko, A. Abaturov, H. Beketova, V. Berezenko, T. Pochynok, and A. Churylina, "New views on the diagnosis and treatment of asthma in children," *Child's Health*, vol. 17, no. 2, pp. 107–113, 2022, doi: 10.22141/2224-0551.17.2.2022.1503.

[10] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms," *Scientific Data*, vol. 8, no. 1, 2021, doi: 10.1038/s41597-021-00937-4.

[11] N. Sharma *et al.*, "Coswara-A database of breathing, cough, and voice sounds for COVID-19 diagnosis," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020, pp. 4811–4815, doi: 10.21437/Interspeech.2020-2768.

[12] T. N. M. Aris, A. A. B. U. Bakar, N. Mahiddin, and M. Zolkepli, "A fuzzy inference model for diagnosis of diabetes and level of care," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 15, pp. 5962–5975, 2023.

[13] B. M. Kogan, L. E. Kuz'mishin, and A. A. Lavrov, "Importance of ultrasonic diagnosis to medical expert evaluation of the work capacity of ischemic heart disease patients," vol. 24, no. 8, pp. 38–41, 1984.

[14] J. L. Mulshine, "One screening for ischemic heart disease, lung cancer, and chronic obstructive pulmonary disease: a systems biology bridge for tobacco and radiation exposure," *American Journal of Public Health*, vol. 108, no. 10, pp. 1294–1295, 2018, doi: 10.2105/AJPH.2018.304655.

[15] A. Fulzele, P. Patil, A. Thakre, A. Mahale, S. Shelke, and C. Engineering, "A mobile application for early diagnosis of pneumonia," 2022.

[16] A. A. Abdelhamid *et al.*, "Robust speech emotion recognition using CNN+LSTM based on stochastic fractal search optimization algorithm," *IEEE Access*, vol. 10, pp. 49265–49284, 2022, doi: 10.1109/ACCESS.2022.3172954.

[17] N. Sengupta, M. Sahidullah, and G. Saha, "Lung sound classification using cepstral-based statistical features," *Computers in Biology and Medicine*, vol. 75, pp. 118–129, 2016, doi: 10.1016/j.compbiomed.2016.05.013.

[18] Z. Mushtaq and S. F. Su, "Efficient classification of environmental sounds through multiple features aggregation and data enhancement techniques for spectrogram images," *Symmetry*, vol. 12, no. 11, pp. 1–34, 2020, doi: 10.3390/sym12111822.

[19] T. A. Adesuyi, B. M. Kim, and J. Kim, "Snoring sound classification using 1D-CNN model based on multi-feature extraction," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 22, no. 1, pp. 1–10, 2022, doi: 10.5391/IJFIS.2022.22.1.1.

[20] S. Prusty, S. Patnaik, and S. K. Dash, "SKCV: stratified k-fold cross-validation on ML classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, 2022, doi: 10.3389/fnano.2022.972421.

[21] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011, doi: 10.1504/ijkesdp.2011.039875.

[22] G. Ke *et al.*, "LightGBM: a highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, pp. 3147–3155, 2017.

[23] N. Dunbray, R. Rane, S. Nimje, J. Katade, and S. Mavale, "A Novel novel prediction model for diabetes detection using gridsearch and a voting classifier between LightGBM and KNN," 2021, doi: 10.1109/GCAT52182.2021.9587551.

[24] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.

[25] Z. Zhang, T. Zhang, and J. Li, "Unbiased gradient boosting decision tree with unbiased feature importance," in *IJCAI International Joint Conference on Artificial Intelligence*, 2023, pp. 4629–4637, doi: 10.24963/ijcai.2023/515.

[26] B. McFee *et al.*, "Librosa: Audio and music signal analysis in in Python," in *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–24, doi: 10.25080/majora-7b98e3ed-003.

[27] G. Lema, Fernando Nogueira, and Christos K. Aridas, "Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[28] T. Hoang, L. Pham, D. Ngo, and H. D. Nguyen, "A cough-based deep learning framework for detecting COVID-19," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2022, pp. 3422–3425, doi: 10.1109/EMBC48229.2022.9871179.

BIOGRAPHIES OF AUTHORS

Wiharto received obtained a bachelor's degree in electrical engineering (B.E.) from Universitas Telkom, Indonesia, in 1999. He obtained a master's degree in computer science from Universitas Gadjah Mada, Indonesia, in 2004 and a Doctoral degree from the same University, in 2017. Currently he works as a lecturer in the Informatics Department, Faculty of Information Technology and Data Science, Universitas Sebelas Maret, Surakarta, Indonesia. His experience and areas of interest focus on artificial intelligence, computational intelligence, biomedical computation, and clinical decision support system. He can be contacted at email: wiharto@staff.uns.ac.id.

Annas Abdurrahman received the bachelor's degree in informatics from the Faculty of Information Technology and Data Science, Universitas Sebelas Maret, Surakarta, Indonesia, 2023. His research interests include soft computing, machine learning, and intelligent systems. He can be contacted at email: annasabdurrahman354@student.uns.ac.id.

Umi Salamah received her bachelor's degree from the Department of Mathematics, Universitas Sebelas Maret, Indonesia, in 1994. She received her master's and Doctoral degrees from the Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia, in 2002 and 2018, respectively. Her research interests include fuzzy logic and systems, image processing, applied mathematics, and computational sciences. She can be contacted at email: umisalamah@staff.uns.ac.id.