Analysis of Respiratory Sound Detection in Lungs Using Machine Learning Techniques

OPEN ACCESS

Volume: 13

Special Issue: 3

Month: February

Year: 2026

P-ISSN: 2321-788X

E-ISSN: 2582-0397

Citation:
Ravikumar, D., et al. "Analysis of Respiratory Sound Detection in Lungs Using Machine Learning Techniques." *Shanlax International Journal of Arts, Science and Humanities,* vol. 13, no. 3, 2026, pp. 340–48.

DOI:
https://doiorg/10.34293/sijash.v13iS3-i2-Feb.10302

Dr. D. Ravikumar
Department of Medical Electronics
SRM Valliammai Engineering College, Chennai

V. Devi
PG & Research Department of Computer Science

DR. Aswinkumar
Student, Department of CSE
Manipal Institute of Technology, Bangalore

S. Sathya
Department of Computer Applications
Thiruthangal Nadar College, Chennai

Abstract

Healthy life enhances the quality of life by allowing individuals in engage able activities and build meaning full relationship for every human. Exploring to air pollution and viral infection can exacerbate or weaken the immune systems defences, making individuals more susceptible to viral respiratory infections such as influenza, asthma, Chronic Obstructive Pulmonary Diseases (COPD), Respiratory Syncytial Virus (RSV), bronchitis and pneumonia. These viral pathogens can worsen existing lung conditions and increases vulnerability leads to respiratory diseases. The virus is mutable continuously, so scientists need more time to make a medical solution for the virus. Based on this reason, high deaths are discovered every day in our life. Thus, a primary shield for protection is required to control outbreaks and pandemics. Hence it is essential that everyone should have a Small portable Lung Disease detection Device that analyses the breath sound. The proposed methodology, based on the Audio sound is used to diagnose the disease in the lungs with assistant of machine learning techniques; our aim is to increases the correct prediction rates and diminishes the false detection rate. It seems that the random forest classifier performs well on this dataset, with a training accuracy of 1.0 and a testing accuracy of 0.95.

Keywords: Viral Infection, Lung Disease, Classifier, Machine Learning Techniques, Performance Metric

Introduction

Indeed with millions of data generating every day, leveraging machine learning can greatly assist in analysing and pre-processing the data to strengthen its quality and usefulness. Machine learning algorithm can help with tasks such as data cleaning, normalization, feature extractions and outlier detection improving the reliability and efficiency of downstream analysis and decision making process. Machine Learning uses many applications like Medical, Electronics, Business Analysis, Cyber Security, and more. There are three types of machine learning. Supervised Learning is trained on labelled data set which implies on both input and output. Unsupervised Learning where unlabelled automatically learns from the input data and furnishing the target. In Reinforcement learning the targets learns to interact with an environment in order to maximize some notion of cumulative rewards. It means the robot encircling in the environment provides an update to its individual based on the positive and negative rewards Example: Alexa Robot.

Reason for Machine Learning is predicted well as a breath sound

Breath sounds contains valuable information's about respiratory health using machine learning to predict health conditions or diagnose diseases based on breath sounds, is a promising area of research and development. The machine learning algorithms can be trained to analyse these sounds and detect indicative patterns of various respiratory diseases. The Data Pre-processing steps had utilized in this process which helps to discard the unwanted features to construct the data robust. Based on breath audio sound using the Mel-Frequency Cepstral Coefficient to convert the data and implement the best features to make the prediction as strongly. By leveraging machine learning, the proposed technique will improve early detection, diagnosis and monitoring of respiratory disorders ultimately leading to better potentate outcomes.

Proposed Systems and Methodology

In the proposed system the detailed process diagram for respiratory audio sound disease detection in lungs using machine learning techniques involves several steps. Here's a high-level overview:

Data Collection implies by gathering respiratory audio recordings from individuals with and without lung diseases. These recordings should cover various conditions such as asthma, pneumonia, bronchitis, etc. In Pre-processing the following processes are encapsulated with Noise removal which eliminates the background noise from the recordings, follows by Segmentation by dividing the recordings into smaller segments for analysis and feature extraction which extract relevant features from the audio signals, such as frequency, amplitude, duration of sounds, etc.

The data are labelled on indicating the presence or absence of lung diseases. Feature Selection are Choose the most relevant features for training the machine learning model. This could involve techniques like feature importance analysis.

In training model an appropriate machine learning algorithms has been selected such as SVM (Support Vector Machine), Random Forest, or Convolutional Neural Networks (CNNs), by splitting the dataset into training and testing sets for model evaluation. The trained model's performances are evaluated using metrics like accuracy, precision, recall, and F1-score. Adjust hyper parameters and/or try different algorithms to improve performance. Once the model meets the desired performance criteria, deploy it in a real-world setting for disease detection.

Continuously monitor the model's performance in production and update it as necessary to maintain its accuracy and effectiveness. The figure 1 illustrates each of these steps with appropriate connections and annotations to provide a clear understanding of the workflow.

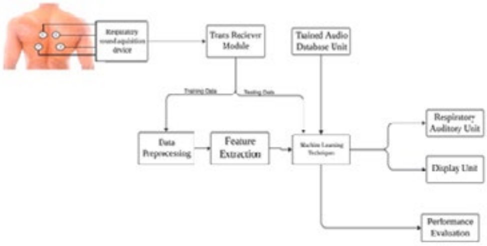

Fig. 1 Process Diagram of Respiratory Audio Sound Detection

Modules

Required Programming Language and Import Necessary Packages

Python stands out as the premier open-source programming language of the modern era due to its user-friendly nature, readability, and minimalistic coding requirements. Serving as a conduit for human-machine communication through coding, Python was conceived by Guido van Rossum in 1991. In contrast to projects developed in C or C++, which demand thousands of lines of code, Python projects typically, require only a fraction of that, often just a hundred lines. This efficiency has led to Python's widespread adoption in diverse domains such as YouTube, web development, and GUI applications. Moreover, Python's versatility is further augmented by essential machine learning packages like pandas, NumPy, Matplotlib, seaborn, and Scikit-learn.

Breath Audio Sound Collection and Description

The respiratory sound database, sourced from Kaggle, encompasses various audio categories, including Chronic Obstructive Pulmonary Diseases (COPD), Bronchiectasis, URTI, Asthma, Pneumonia, Healthy, Bronchiolitis and LRTI. Its significance lies in its multiclass capability, enabling the prediction of diverse respiratory diseases based on audio analysis.

One Hot Coding

One hot coding assists to transform Categorical Data into numerical data. In our Audio sound data, output has Categorical such as Chronic Obstructive Pulmonary Diseases (COPD), Bronchiectasis, URTI, Asthma, Pneumonia, Healthy, Bronchiolitis and LRTI. It is not numerical data, so employing the One Hot Coding; the categorical data are converted into Numerical Data. And also the machine learns when the data is numerical only. By using one hot encoding, categorical data related to respiratory diseases can be efficiently processed and utilized as input features for machine learning models. These models can then be trained to classify diseases, predict outcomes, or assist in diagnostic processes.

Removal less Label

Due to the limited number of audio files for Upper Respiratory Tract infections (URTI) and Asthma (only one or two), it has been decided to exclude them from the dataset due to their insufficient representation.

Data of Feature Extraction in Audio sounds Using Mel Frequency Cepstral Coefficients

The extracted audio sound passes to the Mel Frequency Coefficients Algorithm to transform our audio sound into numerical data features. This conversion is essential as machine learning algorithms only accept numerical data. The resulting features, denoted as X, are extracted alongside the corresponding output classes (COPD, Healthy, etc.), and represented as Y. In future applications, when an audio sample is inputted into the machine learning model, it extracts features and diagnoses the disease based on past training data.

Split Training and Testing

During the training phase, 70% of the Mel Frequency Coefficient converted numerical data is used to teach the machine learning model. This involves providing both input (features) and output (disease classes) to the model. Subsequently, the remaining 30% of the data is reserved for testing. During testing, only the input data is provided to the model, and the machine learning algorithm automatically generates predictions for the corresponding output (disease diagnosis) based on the learned patterns from the training data.

Conversion of Data one Dimension into Other Dimension

After extraction of the numerical features from the audio (.wav) files, those Data which is in 3 dimension array format, which has to be converted into 2 dimension array because the machine learning algorithm accepts 2 dimension array only. Similarly the target output found a two-dimension array, has to be converted into one dimension array.

Working Algorithms

Extra Tree Classifier

The Extra Tree Classifier, a classification-based tree approach extends beyond traditional decision trees by dividing the best nodes based on features rather than simply into two nodes per feature. Optimal scenarios occur when each target output is correctly split into separate leaf nodes, indicating the most advantageous feature columns. Conversely, when at least one target is split in one leaf node, it represents an average feature, while the worst feature occurs when both targets reach both leaf nodes; it is considered the worst outcome for that feature. This analysis helps determine the effectiveness of features in classification tasks.

Decision Tree Algorithm (DT)

The Decision Tree algorithm is indeed a supervised learning algorithm commonly used for classification tasks. It involves training the model on labeled data, where the algorithm learns to make decisions based on input features to classify data into different categories. Decision Trees typically have one root node and then split into two leaf nodes based on certain conditions or features, helping to organize and classify the data effectively.

Gini Index of Decision Tree

The process of selecting feature columns based on the Gini index using a Decision Tree algorithm for classification with two target outputs has been depicted. Consider a scenario with a two-target output. When a Decision Tree perfectly splits one target into one leaf node and the second target into the other leaf node for a specific characteristic, it signifies a helpful feature column with the Minimum Gini index. Additionally, if at least one target appears in one leaf node of a column feature, it is considered a good feature, also known as the minimal Gini Index. Conversely, when both targets appear as leaf nodes, it represents the worst attribute. Ultimately, the feature column with the least Gini index is chosen for training the Decision Tree. This approach helps prioritize and select the most effective features for training the model.

Random Forest Classifier Algorithm

The Random Forest Classifier (RFC) works over a single Decision Tree. Indeed, RFC is an ensemble method that combines multiple Decision Trees to improve predictive accuracy and reduce over fitting, which can be caused by high variance in individual trees.

RFC randomly selects subsets of the training data for each tree, typically around two-thirds of the data, ensuring diversity among the trees. This diversity helps in reducing variance while maintaining predictive accuracy. By aggregating the predictions of multiple trees through majority voting, RFC produces a more stable and reliable prediction compared to a single Decision Tree.

This approach effectively reduces the impact of variance and increases the robustness of the model, by using this simple method and after removing the minority of variance and getting the majority that is a correct output. Besides leading to improved generalization performance on unseen data. Overall, RFC is a powerful algorithm for classification tasks, especially when dealing with high-dimensional or noisy datasets.

KNN (K Nearest Neighbours) Algorithm

K-Nearest Neighbours (KNN) algorithm provides comprehensive details on its application in classification tasks. KNN is indeed a supervised learning algorithm that can be used for binary classification, multi-class classification, and regression problems, although it's commonly known for its effectiveness in classification tasks. In KNN, the algorithm calculates the Euclidean distance between the input data point and all other data points in the training set to determine the nearest neighbours. By considering the majority class among the nearest neighbours (typically the top KNN), KNN predicts the class of the input data point.

During training, each data point is used as a reference point to calculate distances to all other data points, and predictions are made based on the majority class among the nearest neighbours. By repeating this process for each data point, the algorithm gradually builds a model with increasing accuracy. Once trained, the model can be tested on a separate test dataset to evaluate its performance. If the accuracy on the test dataset is satisfactory, the model can be deployed for making predictions on new, unseen data.

Overall, KNN is a simple yet powerful algorithm for classification tasks, and its effectiveness depends on the quality of the training data and the appropriate choice of hyper parameters such as the number of neighbours (k).

Logistic Regression

The concept of logistic regression in both binary and multi-class classification scenarios has been depicted.

In binary logistic regression, the model predicts the probability that an input belongs to a particular class (e.g., 0 or 1). The output of the logistic regression model is typically between 0 and 1, representing the probability of belonging to the positive class. If the probability is greater than or equal to 0.5, the input is classified as belonging to the positive class; otherwise, it's classified as belonging to the negative class.

In multi-class logistic regression, the model can predict the probability of belonging to multiple classes. Each class has its own set of parameters, and the softmax function is used to convert raw scores into probabilities.

Regarding your concern about data splitting on two sides and the potential problem of a large number of one target transferring to the other side of the target, this could indicate a problem with the model's performance. It suggests that the model is not accurately capturing the relationship between the input and the targets. This issue could arise due to various factors such as inadequate features, over fitting, or under fitting.

To address this problem, it's important to carefully analyse the data, select appropriate features, and tune the model's hyper parameters. Additionally, techniques such as regularization can help prevent over fitting and improve the model's generalization ability. Regular monitoring of the model's performance on validation data is crucial for detecting and addressing such issues. Assume two y-axis targets and one input x-axis, with data split on two sides. The model is problematic if a large number of one target transfers to the other side of the target. The first target is over 0.5, whereas the second target is below 0.5.

Performance Metrics

Based on the comprehensive analysis and comparison of various classifiers for the Audio Sound Classification task, it's clear that the Random Forest Classifier stands out as the most suitable choice. Here's a summary of the reasons supporting this decision:

High Accuracy: The Random Forest Classifier achieves impressive training and testing accuracy scores of 1.0 and 0.95, respectively, indicating its ability to effectively classify audio sounds.

Precision and Recall: The classifier also demonstrates strong performance in precision, recall, and F1-score metrics, indicating its ability to correctly classify positive and negative instances with high confidence.

F1-Score

The accuracy statistic is the F1-Score. This is made up of a combination of precision and recall. True Positive divided by the sum of True Positive, False Positive, and False Negative is the mathematical expression for F1- Score.

$$\text{F1 Score} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive} + \text{False Negative}}$$

Confusion Matrix: The confusion matrix analysis further confirms the Random Forest Classifier's effectiveness, showing its ability to correctly classify instances across different classes.

Training Time: Despite its high performance, the Random Forest Classifier has a relatively short training time of 0.25 seconds, making it efficient for real-time or large-scale applications. Considering all factors, for an overall performance including accuracy, precision, recall, confusion matrix analysis, and training time, the Random Forest Classifier emerges as the best-performing classifier for the Audio Sound Classification dataset.

Hence, based on these results and insights, it's concluded that the Random Forest Classifier is indeed the most suitable and effective choice for the Audio Sound Classification task.

Confusion Matrix

The confusion matrix provides a comprehensive view of the model's performance across all output classes, allowing for a deeper understanding of its strengths and weaknesses. It's a valuable tool for evaluating the effectiveness of classification algorithms. The Confusion Matrix is a table structure that permits visualisation of an algorithm's performance by displaying the value count for how much data is properly predicted and not predicted for all output classes. TP and FN denote data that was successfully anticipated, while FP and TN denote data that was incorrectly predicted.

- True Positive (TP): These are cases where the actual output and predicted output are consistent; meaning the model correctly predicted the positive class.
- False Positive (FP): These are cases where the actual output is negative, but the model incorrectly predicted it as positive.
- False Negative (FN): These are cases where the actual output is positive, but the model incorrectly predicted it as negative.

- True Negative (TN): These are cases where the actual output and predicted output are consistent; meaning the model correctly predicted the negative class.

Here's a summary that the description of the confusion matrix and its components is accurate.

Results and Discussion

Based on the comparison of various performance metrics, it appears that the Random Forest Classifier outperforms the other classifiers in terms of both training and testing accuracy, as well as precision, recall, and F1-score. Here's a summary of the comparison:

Table 1. Comparison of Various Performance Metrics

Classifiers\Performance metrics	Trained Score Accuracy	Tested Score Accuracy	Precision	Recall	F1-Score	Training Time(Sec)
Random Forest Classifier	1.0	0.95	0.82	0.57	0.65	0.360
Extra Tree Classifier	1.0	0.82	0.23	0.29	0.25	0.010
Decision Tree Classifier	0.85	0.89	0.15	0.17	0.16	0.068
KNN Classifier	0.90	0.87	0.38	0.23	0.26	2.55
Logistic Regression	0.90	0.91	0.50	0.50	0.45	0.25

1. Random Forest Classifier: Training Score: 100%, Testing Score: 95%, Precision: 0.82, Recall: 0.57, F1-Score: 0.65, Training Time: <1 sec (0.25 sec)
2. Extra Tree Classifier: Training Score: 100%, Testing Score: 82%, Precision: 0.23, Recall: 0.29, F1-Score: 0.25, Training Time: 0.010 sec
3. Decision Tree Classifier: Training Score: 85%, Testing Score: 89%, Precision: 0.15, Recall: 0.17, F1-Score: 0.16, Training Time: 0.068 sec
4. KNN Classifier: Training Score: 90%, Testing Score: 87%, Precision: 0.38, Recall: 0.23, F1-Score: 0.26, Training Time: 2.55 sec
5. Logistic Regression: Training Score: 90%, Testing Score: 91%, Precision: 0.38, Recall: 0.23, F1-Score: 0.26, Training Time: 0.25 sec. This is the over-all about the explanation of the comparison of various performance metrics.

From the results, it's evident that the Random Forest Classifier achieves the highest testing accuracy and performs well across precision, recall, and F1-score metrics. Additionally, it has a relatively short training time compared to some other classifiers like Neighbour's. Therefore, the Random Forest Classifier appears to be the most beneficial model in this comparison.

Confusion Matrix

Confusion Matrix helps to find, how many actual output and predicted output is the same.The following below figures 2(a), 2(b), 2(c), 2(d) and 2(e) illustrate the outputs through the confusion matrix:

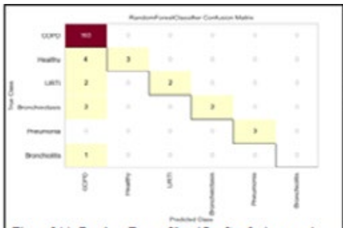

Figure 2(a): Random Forest Classifier Confusion matrix

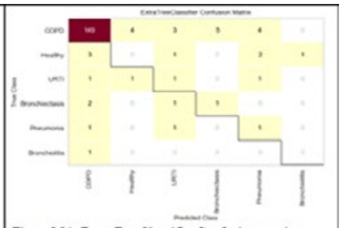

Figure 2(b): Extra Tree Classifier Confusion matrix

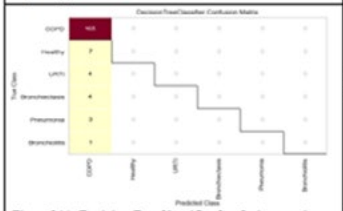

Figure 2(c): Decision Tree Classifier Confusion matrix

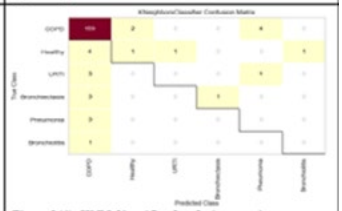

Figure 2(d): KNN Classifier Confusion matrix

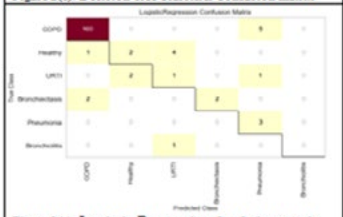

Figure 2(e): Logistic Regression Confusion matrix

In Confusion Matrix if the actual output data and predicted output data same, the accuracy is increased. Here, Confusion Matrix diagonal line says the correctly predicted data. Hence, the Random Forest Classifier Algorithm that the confusion matrix of diagonal line data found high values compared to all the classifiers. So we surely say Random Forest Classifier performs best in this Audio Sound Classification.

Conclusion

After thorough the following comprehensive analysis, it is evident that the Random Forest classifier is the optimal choice for our Audio Sound Classification task. It consistently outperforms other classifiers with a training accuracy of 1.0 and testing accuracy of 0.95, demonstrating excellent precision, recall, and confusion matrix performance. Moreover, its training time of 0.25 seconds is commendable. In particular, the Confusion Matrix of the Random Forest classifier stands out among all classifiers evaluated. Based on these findings, this classifier is the most effective model for this Audio Classification dataset.

References

1. H. G. Kim, N. Moreau, and T. We score, “Sound separation is based on behalf of the MPEG-7 base, ”IEEE Trans. Cycles System Video Technol., Vol. 14, no. 5, pages 716-725, 2004.
2. C.C.Lin, S. H. Chen, T.K.Truong, no-Y. Chang, “Audio classification and segmentation based on events and vector support mechanism,” IEEE Trans. Speech Process.,Vol.13, no. 5, pp 644-651, 2005.
3. Niu, J. L. et al. “Sputum detection by interpreting the time-frequency distribution of the respiratory sound signal using image processing techniques”. Bioinformatics. 34, 820–827 (2018).
4. D. Ravikumar, V. Devi, Arun Raaza. “Development of Brain Computer Interface using Neural Network”. Research J. Pharm. and Tech 2018; 11(10): 4397-4400. doi: 10.5958/0974-360X.2018.00804.1

5. Yadullahi, A., Giannouli, E. and Mousavi, Z. Monitoring and diagnosis of sleep apnea is based on pulse oximeter and tracheal sound signals. Med Biol Engg Comp. 48, 1087-1097 (2010).
6. Brower, R. G. et al. Ventilation with lower tidal volume than conventional tidal volume for acute lung injury and acute respiratory distress syndrome. N. Engg. J. Made. 342, 1301-1308 (2000).
7. Sutherasan, Y., Vargas, M. and Pelosi, P. "Protective mechanical ventilation in the non-injured lung: review and meta-analysis". Crit Care. 18, 211-223 (1997).
8. Luccini, A. et al. "Tracheal secretion management in a mechanically ventilated patient: comparison of standard evaluation and an acoustic secretion detector". Respiratory care. 56, 596-603 (2011).
9. Guglielminotti, J., Desmontes, J.M. and Dureuil, B. "Effects of Tracheal Suctioning on Respiratory Resistance in Mechanically Ventilated Patients. Cena. 113, 1335-1338 (1998).
10. Gurrala Chandrashekar, Arun Raaza, V. Rajendran, D. Ravikumar, "Side scan sonar image augmentation for sediment classification using deep learning-based transfer learning approach", Materials Today: Proc. Vol 80 (3),2023, Pages 3263-3273,ISSN 2214-7853,doi.org/10.1016/j.matpr.2021.07.222
11. Sarkar, M., Madabhavi, I., Niranjan, N. and Dogra, M. Auscultation of the respiratory system. Ann Thorac Med. 10, 158-168 (2015).
12. G. Manohar, V. K. Sundari, A. E. Pious, A. Beno, L. D. V. Anand and D. Ravikumar, "IoT based Automation of Hydroponics using Node MCU Interface," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 32-36, doi: 10.1109/ICIRCA51532.2021.9544637.
13. Anitha Vijayalakshmi, B., Senthil Kumar, A., Kavitha, V. et al. Transmitting patient's health care information using LEDs in hospitals through VLC technology. J Opt 53, 4623–4630 (2024). https://doi.org/10.1007/s12596-023-01650-8
14. Bahura, M. "Applied pattern recognition methods for the classification of respiratory sounds into normal and wheezing classes". Compute Biol Med. 39, 824-843 (2009).
15. Nogata, F. At al. Audio-visual recognition of auscultatory breathing sounds using Fourier and wavelet analysis. Asian Journal of Computers and Information Systems. 3, 96-105 (2015).
16. M Monisha, P Vijayalakshmi, V Rajendran, D Ravikumar. Artificial Intelligent Based E- Health Monitoring Using Low Power Device in Wide Area Network. New Ideas Concerning Science and Technology Vol. 11, Book Publisher International (a part of SCIENCEDOMAIN International), pp.128-136, 2021, ⟨10.9734/bpi/nicst/v11/8149D⟩. ⟨hal-05120251⟩