Advances in Artificial Intelligence and Machine Learning; Research 5 (4) 4484-4500 Received 03-08-2025; Accepted 01-11-2025; Published online 08-11-2025

Systematic Evaluation of Handcrafted Features and Classical Machine Learning for Respiratory Sound Analysis

Constantin Constantinescu ctin.constantinescu@ulbsibiu.ro
Department of Computer Science and Electrical Engineering
Lucian Blaga University of Sibiu
Sibiu, Romania

Remus Brad remus.brad@ulbsibiu.ro
Department of Computer Science and Electrical Engineering
Lucian Blaga University of Sibiu
Sibiu, Romania

Corresponding Author: Constantin Constantinescu

Abstract

The classification of respiratory diseases is an important problem that most researchers have tried to solve directly by using deep learning. Traditional machine learning with handcrafted features has been left behind and is less explored in this context, although it may offer efficiency and interpretability. In this paper, we performed a comparison between multiple machine learning algorithms. We applied the algorithms on respiratory sound data from the ICBHI Dataset. We extracted various features from the data, features that are more suitable for signals. We used the features both individually and together. The task was a classification one, both binary and multiclass. We ran each algorithm separately with the feature sets. For each algorithm, we performed more than 1200 runs using different parameters to optimize their learning and overall performance. Random Forest performed best, showing very promising results with accuracy and a F1 score of almost 80%, closely followed by k Nearest Neighbors. Other algorithms stood out only with certain features, while others returned suboptimal results. This experiment showed that a good choice of features, preprocessing and hyperpa rameter optimization play a very important role in classic machine learning that remains competitive in certain scenarios.

Keywords: Classic machine learning, Handcrafted features, Sound anomaly detection, Respiratory disease classification.

1. INTRODUCTION

Respiratory diseases are a major health burden where early diagnosis is essential. Hospitals are usually equipped with fancy imaging equipment for this. However, in primary care, this equipment is usually not available, but every physician owns and uses a stethoscope every day. With the development of electronic stethoscopes, we can now record and process respiratory sounds from

Citation: Constantin Constantinescu and Remus Brad. Systematic Evaluation of Handcrafted Features and Classical Machine Learning for Respiratory Sound Analysis. Advances in Artificial Intelligence and Machine Learning. 2025;5(4):249. https://dx.doi.org/10.54364/AAIML.2025.54249

the patients chest wall. This allows us to further develop systems for computer aided diagnosis. Using the combination of an electronic stethoscope and a machine learning driven solution, even personnel with minimal medical training can detect early possible signs of a respiratory disease. This is particularly useful in underdeveloped countries or generally in underserved areas, where not only expensive medical equipment is missing, but also doctors are not available.

The ICBHI data set [1], is a widely used reference in lung sound classification, so we will use it as well. A lot of work has been done on this dataset.

Most studies demonstrate the effectiveness of deep learning for this task. Convolutional neural networks work best with spectrograms or other time-frequency representations of signals. [2–10] use various self-developed CNN architectures for the task. [11] uses ResNet-18, [12–14], uses ResNet-24 and [15], uses ResNet-50.

Since sound is fundamentally a signal, multiple studies have tackled the task with recurrent neural networks. LSTM and Bi-LSTM are the most common [16–19], but other variants of RNNs are also present in the literature [20].

Some studies went a step further and show that a hybrid approach is even better. In [3, 21–23], the authors developed models that combine CNNs and RNNs by adding a LSTM layer to the original convolutional layers.

Compared to deep learning approaches, there are very few studies with machine learning based respiratory sound classification. [24] uses a tree model for binary classification. For a higher number of classes, [25] uses the SVM, [26] uses Hidden Markov Model and Gaussian Mixture Module and [27], uses Random Forest.

It can be clearly observed that most studies focus on deep learning and very few works have systematically analyzed traditional machine learning algorithms with handcrafted features on this dataset. Although deep learning is a very powerful tool, sometimes it is not necessary to use a forklift to pick up a toothpick. There are applications where classic machine learning reached results comparable to deep learning, showing that it is still relevant across domains. The authors of this paper [28], mention that the performance of support vector machines is equal or even superior to that of a neural network or a mixture model in photovoltaic applications [29–31]. This paper [32], even presents an example in the medical field, developing a machine learning based diagnosis tool for dysgraphia.

So, the question that we would like to answer in this study is: Can classical machine learning algorithms, combined with handcrafted sound features, provide reliable performance for respiratory sound classification (binary and multiclass) compared to deep learning approaches?

To try to answer this question, we did a systematic comparison of five machine learning algorithms: k-NearestNeighbours, LVQ, Support Vector Machine, Decision Tree and Random Forest. The task consists of classification of respiratory sounds, binary and multiclass. The sound features were extracted from the data and we performed hyperparameter optimization on the algorithms. Unlike previous studies on the ICBHI dataset that tested isolated algorithms or small sets of handcrafted features, this study systematically compares five classical ML algorithms across binary and multiclass tasks, exploring over 1,200 hyperparameter configurations. To our knowledge, this is the most extensive evaluation of traditional ML on ICBHI.

In this way, we get insights into when and why traditional machine learning is effective, insights that serve as basis for future work and discussions.

2. METHODOLOGY

2.1 Dataset

The ICBHI 2017 Respiratory Sound Database [1], was originally created for a scientific challenge that took place at the International Conference on Biomedical Health and Informatics in 2017. The data set contains lung sounds, recorded from the chest wall using electronic stethoscopes. The recordings come from 126 real patients and are a total of 5.5 hours. The recordings were segmented into respiratory cycles by respiratory experts, who also established for each cycle if it contains a crackle, a wheeze, or both. This resulted in 6898 respiratory cycles, of which 886 contain wheezes, 1864 crackles, and 506 both. In addition to this annotation at the cycle level, a diagnosis for each patient is also available. Among them, we can find chronic obstructive lung disease, lower respiratory tract infection, upper respiratory tract infection, pneumonia, asthma and others. At the recording level, we also have information about the stethoscop position on the patient's chest, the recording equipment and the acquisition mode. Demographic information is also available for each patient, but we will not take it into account for this study. The database is anonymized and freely available for research.

2.2 Preprocessing

Every recording has been resampled to 4000 samples per second using FFT-based band limited interpolation to avoid any aliasing effect.

2.2.1 Segmentation

The easiest way to perform segmentation on a sound data is to choose a fixed length, split the recording into segments with this uniform length, and leave the rest out. However, the respiratory sound is somewhat cyclic, so with fix length segmentation you risk splitting the sound in the middle of inhaling or exhaling. Although cyclic, every patient has a different respiratory rhythm, therefore the cycles have different lengths.

We chose this data set that has manually labeled cycles. Every recording has a separate file that contains the cycles on separate lines, with start point of the cycle, end point of the cycles, and whether it contains crackels, wheesez, or both. We extracted the cycles based on these data, but we still ran into an issue. Obviously, the cycles have different lengths, but for our machine learning approach, we need fixed-length arrays. To achieve this, we computed the median value of the cycles' lengths, resulting in **2.81** seconds per cycle. We then resampled each recording, actually performing time stretching on each segment in order to reach this particular length. If the segments have identical lengths, after the feature extraction, the arrays will also have uniform lengths.

2.2.2 Normalization

Before feeding the arrays to the machine learning algorithms, we normalize them. We ran every configuration with two normalization types, namely the nominal normalization and the sum one normalization and with no normalization at all. The nominal normalization divides each element of the array by the element with the highest value:

$$\tilde{x_i} = \frac{x_i}{\max_{j-0}^{n} x_j}$$

The sum one normalization performs a summarization of all elements of the array and then divides each element by the sum:

$$\tilde{x_i} = \frac{x_i}{\sum j - 0^{n-1} x_j}$$

2.3 Feature Extraction

Feature extraction is essential when working with classic machine learning algorithms. We chose 3 features in the time domain, 3 in the frequency domain, and a couple with both time and frequency information, namely a classic spectrogram and the very popular Mel Frequency Cepstral Coefficients.

In time domain, we calculated the amplitude envelope, the root mean square energy and the zero crossing rate.

The amplitude envelope provides a general idea of the loudness since the amplitude is directly related to the intensity of the sound [33]. Respiratory anomalies, such as crackles or wheezes, often appear as short bursts or continuous tonal sounds with distinct intensity patterns. AE can help distinguish such abnormal events from normal breathing. It is calculated for each frame and represents the maximum absolute value of the amplitude in this particular frame:

$$AE_t = max_{k=t\cdot K}^{(t+1)\cdot K-1} s(k)$$

The root mean square energy is, as the name already states, the root mean square of all the samples in a frame:

$$RMS_t = \sqrt{\frac{1}{K} \cdot \sum_{k=t\cdot K}^{(t+1)\cdot K-1} s(k)^2}$$

Also an indicator of loudness, this feature is less sensitive to extreme values, as opposed to the amplitude envelope[33]. It is useful for detecting continuous high-energy segments, a sign of wheezes.

The zero crossing rate [33], is a very popular feature that gives us the number of times a signal crosses the (zero-) horizontal axis. It helps us distinguish between percussive and pitched sounds and has proved to be very effective in some classification tasks. Higher ZCR indicates more noisy or percussive signals (crackles), while lower ZCR corresponds to more harmonic signals(wheezes). To calculate the zero crossing rate, we analyze two consecutive amplitudes and check their sign. If

the sign changed, then we have a zero crossing:

$$ZCR_t = \frac{1}{2} \sum_{k=t\cdot K}^{(t+1)\cdot K-1} |sgn(s(k)) - sgn(s(k+1))|$$

To get information about the frequency of the sound we calculated the band energy ratio, the spectral centroid, and the bandwidth.

The band energy ratio gives us information about the relation between the energy in the lower and higher frequency bands [33]. Basically, it tells us whether the lower frequency bands are dominant or the higher ones:

$$BER_t = \frac{\sum_{n=1}^{F-1} m_t(n)^2}{\sum_{n=F}^{N} m_t(n)^2}$$

Crackles usually exhibit higher frequency content compared to normal breathing.

The second feature of our choice from the frequency domain is the spectral centroid. It shows us the frequency bands where the most energy is concentrated:

$$SC_t = \frac{\sum_{n=1}^{N} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)}$$

This feature has been used in similar applications as ours, in applications for automatic diagnosis of COVID-19 from sound data, and also in other medical applications. Proven to work very well with the classic machine learning algorithm, it is considered one of the key features in the frequency domain[33].

Somewhat related to the spectral centroid, the bandwidth represents the weighted mean of the distance between each frequency band and the spectral centroid:

$$BW_t = \frac{\sum_{n=1}^{N} |n - SC_t| \cdot m_t(n)}{\sum_{n=1}^{N} m_t(n)}$$

It shows us how the energy is distributed between frequency bins. It is used a lot in music processing [33]. A broad bandwidth can indicate noisy signals (for example crackles), while narrower bandwidth suggests more tonal components (wheezes).

Moving on to more complex features, the spectrogram is a representation of the sound where both the time and the frequency components of the sound are represented. We can obtain a spectrogram starting from the wave-form representation of the signal and applying a short-time Fourier transform. In a spectrogram we have information about both time and frequency. Spectrograms are used in signal anomaly detection as input to convolutional neural networks. They are also used in the MIMII baseline algorithm [34], for the detection of sound anomalies in industrial machines. The spectrogram is the most direct way to visualize and analyze respiratory events.

Mel Frequency Cepstral Coefficients (MFCCs) are a type of feature representation commonly used in speech and music analysis. They are derived from the Mel Spectrogram, which is a modified version of the traditional spectrogram that is designed to better match the way the human ear perceives sound. MFCCs are calculated by applying the following steps to a Mel Spectrogram:

1. Take the logarithm of the power in each frequency bin
2. Apply the Discrete Cosine Transform (DCT) to the log-power spectrum
3. Keep only the first few DCT coefficients, typically between 12 and 40, depending on the application

The resulting coefficients represent a compact and robust summary of the spectral shape of the signal in the Mel-scale. MFCCs are commonly used in speech recognition, speaker identification, music analysis, and other audio-related tasks. They are particularly useful for deep learning models because they are relatively robust to variations in the signal caused by changes in the speaker's voice, background noise, and other factors. They are also relatively insensitive to changes in the signal caused by changes in the recording environment or equipment. Being so robust, they are very suitable for clinical data.

2.4 Algorithms

The algorithms of our choice were KNN, LVQ, SVM, Decision Tree, and Random Forest. We focused only on supervised learning, since we chose a labeled dataset.

2.4.1 KNN

The k-Nearest Neighbors (k-NN) algorithm assigns a class to each audio segment by determining the majority class among its k nearest neighbors in the feature space [35].

To calculate the similarity between the feature vectors we tried three methods.

Euclidian distance:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=0}^{n-1} (a_i - b_i)^2}$$

Manhatten distance:

$$d(\mathbf{a}, \mathbf{b}) = \sum_{i=0}^{n-1} |x_i - y_i|$$

Cosine similarity:

$$s(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

For constant k, we ran with odd values between **3** and **13**.

2.4.2 LVQ

Learning Vector Quantization is an algorithm where we take some representative feature vector for each class and compare all other vectors to these prototypes. The training vector is moved closer to the prototype with matching class and away from other prototypes[36]. Moving a vector closer and farther away from certain prototypes is done using a variable called a learning rate. To compare feature vectors with prototypes, we used the same distances as the kNN. For the learning rate, we chose values between **0.01** and **0.2**.

2.4.3 SVM

Support Vector Machines is a machine learning algorithm that tries to find the best boundary (hyperplane) that separates different classes, with the maximum margin. By margin, we understand the difference between the boundary and the feature arrays closest to the boundary, which are called support vectors [37]. If the data is not easy to separate linearly, the algorithms use a so-called ”kernel trick”, meaning that it maps the data into a higher-dimensional space, where it becomes separable. The most popular kernels are the linear kernel, which is the simple one that only works if the data is linearly separable. The polynomial kernel, on the other hand, allows for curved boundaries, while the RBF (Radial Basis Function) kernel is the most flexible and can create complex, nonlinear boundaries. The sigmoid kernel is also used for non-linear boundaries. It usually creates S-shaped boundaries and it rarely is used, since RBF is more stable.

Even with this trick, in reality the data is not always perfectly separable. We can choose to allow some misclassification, controlled by a parameter: C. A higher C means less tolerance, while a lower C may provide a better generalization.

In our experiment, we used all four kernels and C values between **0.1** and **10**.

2.4.4 Decision tree

Decision Tree works by splitting the data into subsets based on feature values and forming a tree structure, where the leaves are the final prediction [38]. It can be seen as a questionaire where you ask a series of yes/no questions to make a decision. One might wonder what question to ask? The goal is to divide the data into groups that are as pure as possible. So common criteria for splitting are the Gini Impurity, Entropy (Informational Gain), or the LogLoss.

Another hyperparameter in the Decision Tree Classifier is the splitter. Here, there are two options, and we tried both. An option is the best splitter, where the algorithm evaluates all features and chooses the best split. Another option is that the algorithm randomly chooses a subset of features and thresholds and makes a decision by evaluating only them.

2.4.5 Random forest

An extension of the Decision Tree with random splitter is the Random Forest. Instead of one big tree, Random Forest randomly builds many smaller trees and then combines their prediction using the majority vote [39]. The criteria are the same as we used for the decision tree. However, an important parameter is the number of estimators, or the number of trees in the forest. If the number is too small, the forest is unstable. Higher numbers means better prediction, but with the cost of more resources. There is no harm in having a very large number of trees, because the algorithm does not overfit. We used values between **50** and **250** when we observed that the accuracy stopped improving.

2.5 Experimental Setup

Starting for the ICBHI Dateset, we applied some preprocessing, consisting from resampling and segmentation of the recordings. The experiment was split into two tasks, namely binary classification and multiclass classification. Binary classification aims to determine whether a respiratory cycle contains anomalous events or not. Multiclass, with four outputs, aims to distinguish between respiratory cycles that contain crackles, wheezes, both, or no events. For both tasks, we performed the same algorithm runs.

In terms of features, we have three scenarios. In the first scenario, we combined the 6 basic sound features(AE, RMSE, ZCR, BER, SC, BW). In this paper, when we mention basic features, we refer to these six. Each of them captures different characteristics of the sound. Used individually as input for machine learning algorithms, the results are suboptimal. To have the entire picture, in our experiment we used all six of them combined in a single feature vector.

The spectrogram and the MFCCs already contain information about both time and frequency, so we used these two individually, with the flattened MFCCs representing the second scenario and the flattened spectrogram the third.

After preparing the feature vectors according to the three scenarios, we performed class balancing, normalization and a train-test split. In case of binary classification, the data set is fairly balanced. However, for the multiclass approach, we downsampled and upsampled the data to achieve balance. The downsampling was performed by randomly leaving out samples. Upsampling was performed by randomly duplicating samples. The samples were then normalized using one of the two normalization methods presented and divided into training and testing sets with an 80/20 ratio, maintaining the class distribution in both subsets.

For each task and scenario, we trained all algorithms multiple times with all combinations of hyperparameter values.

We evaluated the algorithms using 4 different metrics, namely accuracy, sensitivity (TPR), specificity (TNR) and F1-Score. A visual representation of these steps is shown in FIGURE 1.

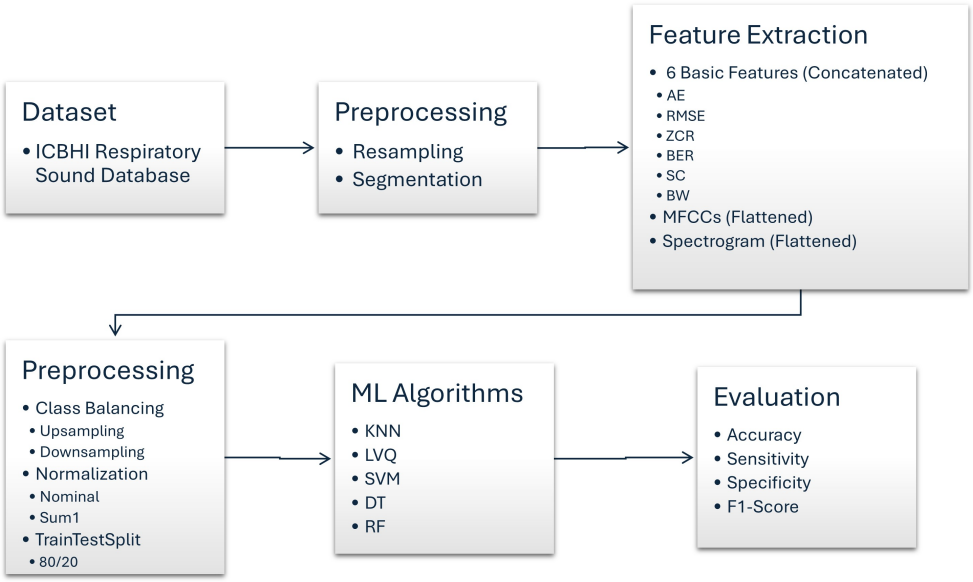

Figure 1: Visual representation of the experimental setup

3. RESULTS

In this section, we present the results of our systematic evaluation, comparing the performance of the classical machine learning algorithms across binary and multiclass respiratory sound classification tasks using different feature sets. FIGURE 2 shows an overview of the best F1-Scores in both binary and multiclass.

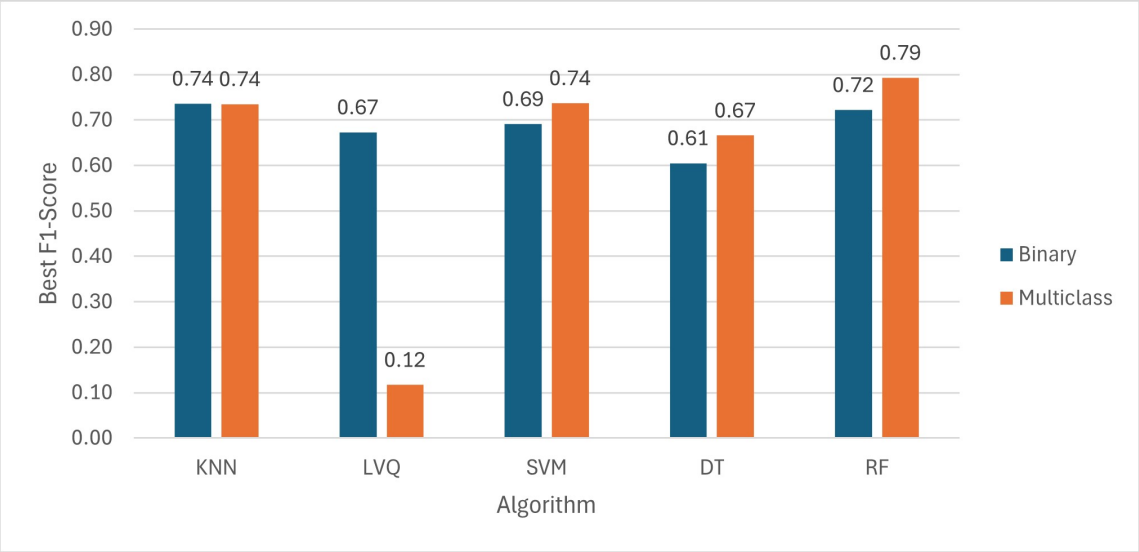

Figure 2: Visual representation of the best F1-Scores

3.1 Binary Classification Results

TABLE 1 shows the best results for each algorithm in all three feature scenarios for the binary classification task. The best overall configuration was obtained with **MFCC** features combined with the **KNN** classifier. The parameters of the algorithms were **k=3**, with **cosine** similarity and **nominal** normalization. This configuration achieved an accuracy and F1-Score of **0.736**. Closely following this, the runner-up configuration was obtained with **MFCC** features combined with a **Random Forest** classifier. In this setup, no normalization was applied, the **Gini** criterion was used for splitting, and the forest consisted of **250** trees. This configuration reached an accuracy of **0.719** and an F1-score of **0.713**. Compared to deep learning methods from the existing literature, deep learning models already reach accuracies of 90+% [3, 12, 40], while our approach topped out at 73%.

Table 1: Best binary classification results by feature set and classifier

Feature set	**Classifier**	**Metrics**				**Winning Config**
		Acc	TPR	TNR	F1	
Basic features	kNN	0.670	0.694	0.648	0.673	nominal, manhattan, k=7
	LVQ	0.573	0.900	0.260	0.673	sum1, euclidean, lr=0.01
	SVM	0.652	0.733	0.575	0.673	nominal, rbf, C=10
	DT	0.605	0.618	0.594	0.605	None, crit=entropy, split=random
	RF	**0.717**	**0.726**	**0.709**	**0.715**	None, crit=gini, n=250
MFCCs	**kNN**	**0.736**	**0.750**	**0.722**	**0.736**	nominal, cosine, k=3
	LVQ	0.576	0.614	0.540	0.586	nominal, manhattan, lr=0.05
	SVM	0.673	0.747	0.602	0.691	nominal, rbf, C=10
	DT	0.606	0.601	0.611	0.599	None, crit=entropy, split=random
	RF	0.719	0.747	0.691	0.722	None, crit=gini, n=250
Spectrogram	**kNN**	**0.670**	**0.637**	**0.702**	**0.654**	sum1, manhattan, k=5
	LVQ	0.577	0.750	0.411	0.634	sum1, euclidean, lr=0.05
	SVM	0.636	0.676	0.598	0.645	nominal, rbf, C=1
	DT	0.572	0.593	0.552	0.575	nominal, crit=entropy, split=random
	RF	0.657	0.679	0.636	0.659	nominal, crit=entropy, n=200

3.2 Multiclass Classification Results

TABLE 2 shows the best results for each algorithm in all three feature scenarios for the multi-class task. The best overall configuration for this task was obtained with the six **basic features** combined with a **Random Forest** classifier. In this setup, **nominal** normalization was applied, the **Gini** criterion was used for splitting, and the forest consisted of **200** trees. This configuration reached an accuracy of **0.777** and an F1-score of **0.793**, representing the strongest multiclass setup we obtained. The runner-up configuration was achieved with **MFCC** features combined with a **Random Forest** classifier. Here, the same criterion and number of trees were employed, but with **sum1** normalization. This configuration reached an accuracy of **0.769** and an F1-score of **0.792**.

Comparing these results with the literature, we notice that the results of deep learning models in multiclass [20, 21], are around 70%, lower than ours that peaked at almost 80%.

Table 2: Best multiclass classification results by feature set and classifier

Feature set	Classifier	Metrics				Winning Configuration
		Acc	TPR	TNR	F1	
Basic features	kNN	0.622	0.649	0.874	0.623	`sum1, manhattan, k=3.`
	LVQ	0.279	0.270	0.756	0.117	`nominal, euclidean, lr=0.01, iters=10.`
	SVM	0.478	0.449	0.804	0.437	`nominal, rbf, C=10.`
	DT	0.672	0.694	0.890	0.675	`sum1, crit=gini, split=best.`
	RF	**0.777**	**0.787**	**0.924**	**0.793**	`nominal, crit=gini, n=200.`
MFCCs	kNN	0.733	0.744	0.911	0.735	`nominal, manhattan, k=3.`
	LVQ	0.290	0.250	0.750	0.115	`nominal, euclidean, lr=0.01, iters=10.`
	SVM	0.521	0.518	0.838	0.519	`nominal, rbf, C=10.`
	DT	0.663	0.678	0.887	0.666	`sum1, crit=log_loss, split=best.`
	RF	**0.769**	**0.784**	**0.919**	**0.792**	`sum1, crit=gini, n=200.`
Spectrogram	kNN	0.683	0.701	0.893	0.691	`sum1, manhattan, k=3.`
	LVQ	0.298	0.250	0.750	0.115	`nominal, euclidean, lr=0.01, iters=10.`
	SVM	**0.721**	**0.737**	**0.905**	**0.737**	`nominal, rbf, C=10.`
	DT	0.641	0.666	0.878	0.650	`nominal, crit=entropy, split=best.`
	RF	0.720	0.736	0.904	0.743	`sum1, crit=gini, n=200.`

4. DISCUSSION

4.1 Algorithm Analysis

Random Forest. RF is robust and consistently the best performer in both tasks. It is also consistently strong across features. Respiratory sounds are very similar to each other and some features can be correlated. RF handles redundancy by training multiple decision trees on random subsets of features and then aggregating the predictions. This explains why RF delivered reliable, high F1 scores across feature sets and normalization strategies. Although it benefits from sum1 normalization, it works well with no normalization at all. While in the binary task, KNN slightly beats it, we can observe a very nice balance of Sensitivity/Specificity. For this application, best configuration are with 200-250 trees and Gini and log-loss criteria both work well. If you need a strong default classical baseline, RF is the safest option.

K-Nearest Neighbors. KNN is simple but effective, especially with proper distance metrics, in this case cosine similarity and Manhatten distance. Being distance-based and non-parametric, its performance depends a lot on feature representation and normalization. It performed best in

binary classification with MFCCs because MFCCs create a feature space where abnormal and normal cycles separate well, which is ideal for KNN's decision making. Generally, the nominal normalization suited KNN best. It yielded the best results with 3 neighbors, while slightly larger k (5-7) came second. KNN shines in binary and is otherwise a strong but not the best multiclass option.

Support Vector Machine. SVM shined in a particular combination: Multiclass with spectrograms as features. RBF kernel beats all other kernel options, while sigmoid consistently underperformed. Generally, results improve by increasing the value of the C parameter, peaking at 10. So, the combination SVM-RBF is valuable for multiclass spectrograms, probably because spectrograms are high dimensional and sparse and the kernel maps them intro an infinte-dimensional space where separation is easier. Otherwise, RF/KNN generally outperform it here.

Decision Tree. Decision tree had moderate performance being very sensitive to hyperparameters. Interestingly, in binary classification the random split performed better. General results show that decision tree metrics are around 0.60 in binary and 0.67 in multiclass demonstrating that single trees are consistently weaker than RF and KNN. This happens because of the limited data and correlated features that cause the model to overfit.

Learning Vector Quantization. LVQ had the lowest performance and may need more tuning. We found it to be unreliable because many configurations failed to make a prediction and ended up predicting a single class. The LVQ is either inappropriate for this application or needs substantial tuning changes. The representation of classes through a small number of prototypes is too fragile for noisy respiratory data.

4.2 Features Analysis

In terms of features, the MFCCs were the clear winner of our experiment. Basic features are competitive but below MFCC. In multiclass, the best result was achieved with these features, but MFCCs wins in most cases. Spectrograms are harder for classic machine learning and underperfomed when fed directly. The best model that actually performed with spectrograms was SVM-RBF (C=10) with Accuracy 0.721.

4.3 Normalization Impact

The impact of the normalization strategy is specific for every algorithm. Normalization is indeed very important for up to 7.4% improvement in our case. Nominal normalization is best suited for distance-based methods, while sum1 is optimal for ensemble methods. Three-based methods performed better without normalization in specific cases.

4.4 Statistical Summary

Scope (total search space). We evaluated 1,278 configurations in total (639 Binary and 639 Multiclass), evenly split across three feature sets (213 configs each per task: basic features, MFCCs, Spectrogram). TABLE 3 shows performance ranges for accuracy and F1-Scores from all configurations. The extremely low minimum from the F1-Score comes from the LVQ results, where it predicted only one class. TABLE 4 shows the mean, standard deviation and maximum value of the accuracy, at feature level, by task.

Table 3: Performance ranges across all configurations.

Metric	Task	Min	Q1	Mean	Median	Q3	Max
Accuracy	Binary	0.414	0.559	0.608	0.620	0.653	0.736
	Multiclass	0.230	0.309	0.525	0.565	0.676	0.736
F1-Score	Binary	0.022	0.567	0.617	0.639	0.687	0.736
	Multiclass	0.109	0.309	0.505	0.587	0.656	0.804

Table 4: Feature-level accuracy (mean ± SD; max) by task

Task	Feature set	Mean ± SD	Max
Binary	Basic features	0.613 ± 0.069	0.717
	MFCCs	0.618 ± 0.083	0.736
	Spectrogram	0.593 ± 0.049	0.670
Multiclass	Basic features	0.512 ± 0.181	0.790
	MFCCs	0.516 ± 0.195	0.779
	Spectrogram	0.546 ± 0.155	0.721

4.5 Practical Significance and Clinical Relevance

Demonstrating that classical machine learning can achieve competitive performance in respiratory sound classification has several implications for clinical practice and the broader field of medical AI.

First, classical methods are computationally lightweight compared to deep neural networks. They can run on low cost embedded or mobile devices. This is in particular important for low-resource environments, such as rural clinics or underserved regions, where these techniques help frontline healthcare workers perform reliable pre-screening of patients using an electronic stethoscope.

Second, traditional models are more interpretable than deep learning models. This is an advantage in clinical applications because algorithms like Random Forest or KNN allow the inspection of feature importance, showing which properties of the sound weighted in the decision making process. This makes clinicians trust intelligent diagnosis systems, who are often reluctant to use black-box systems.

Third, this study advances the current state of the art by showing that classical models do not always underperform compared to deep learning. Our results indicate that traditional methods can match or even exceed deep networks, provided that features and preprocessing are well chosen. This finding challenges the assumption that deep learning is always superior and encourages further exploration of hybrid solutions that combine the interpretability and efficiency of classical models with the representational power of neural networks.

Finally, these results show that traditional methods and deep learning can work together. For example, lightweight models could perform initial patient triage, flagging suspicious cases for further analysis using more computationally expensive deep learning models. This layered approach could enable the scalable and cost-effective deployment of respiratory sound analysis technologies.

5. CONCLUSIONS

The scope of this research was to determine whether classic machine learning techniques are reliable in respiratory sound classification, compared to deep learning methods. After analyzing the results of five classic machine learning algorithms trained in a total of more than 1,200 configurations, we can safely say that yes, classical machine learning is viable for respiratory sound analysis. With the right features and preprocessing, we achieved strong results. Compared to deep learning methods, we notice two aspects. On the one hand, in binary classification, classic machine learning models achieve reasonable results, but deep learning models are better. On the other hand, with 4 classes, the performance of deep learning models is weaker than that of classic machine learning. Random Forest is the most robust baseline followed by the kNN that also excels with appropriate parameters. SVM with RBF kernels is the best choice when dealing with raw spectrograms. Bottom line, classical methods can match or complement more complex models for these tasks without needing that much data and offering more interpretability. Their low computational cost and interpretability make them especially attractive for low-resource environments.

References

[1] Rocha B, Filos D, Mendes L, Vogiatzis I, Perantoni E, et al. A Respiratory Sound Database for the Development of Automated Classification. In: International Conference on Biomedical and Health Informatics. Springer. 2017:33-37.

[2] Perna D. Convolutional Neural Networks Learning From Respiratory Data. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE. 2018:2109-2113.

[3] García-Ordás MT, Benítez-Andrades JA, García-Rodríguez I, Benavides C, Alaiz-Moretón H. Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational Autoencoders for Unbalancing Data. Sensors. 2020;20:1214.

[4] Pham L, Phan H, Schindler A, King R, Mertins A, et al. Inception-Based Network and Multi-Spectrogram Ensemble Applied to Predict Respiratory Anomalies and Lung Diseases. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society. 2021:253-256.

[5] Shuvo SB, Ali SN, Swapnil SI, Hasan T, Bhuiyan MI. A Lightweight CNN Model for Detecting Respiratory Diseases From Lung Auscultation Sounds Using EMD-CWT-Based Hybrid Scalogram. IEEE J Biomed Health Inform. 2021;25:2595-2603.

[6] Borwankar S, Verma JP, Jain R, Nayyar A. Improvise Approach for Respiratory Pathologies Classification With Multilayer Convolutional Neural Networks. Multimedia Tool Appl. 2022;81:39185-39205.

[7] Nguyen T, Pernkopf F. Lung Sound Classification Using Snapshot Ensemble of Convolutional Neural Networks. In: 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE. 2020:760-763.

[8] Ren Z, Nguyen TT, Nejdl W. Prototype learning for interpretable respiratory sound analysis. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. 2022:9087-9091.

[9] Tariq Z, Shah SK, Lee Y. Lung Disease Classification Using Deep Convolutional Neural Network. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE. 2019:732-735.

[10] Shehab SA, Mohammed KK, Darwish A, Hassanien AE. Deep Learning and Feature Fusion-Based Lung Sound Recognition Model to Diagnoses the Respiratory Diseases. Soft Comput. 2024;28:11667-11683.

[11] https://www.isca-archive.org/interspeech_2020/yang20e_interspeech.pdf

[12] Rahman MM, Shokouhmand S, Faezipour M, Bhatt S. Attentional Convolutional Neural Network for Automating Pathological Lung Auscultations Using Respiratory Sounds. In: International Conference on Computational Science and Computational Intelligence (CSCI). IEEE. 2022:1429-1435.

[13] Gairola S, Tom F, Kwatra N, Jain M. Respirenet: A Deep Neural Network for Accurately Detecting Abnormal Lung Sounds in Limited Data Setting. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). 2021:527-530.

[14] Wang F, Yuan X, Meng B. Classification of Abnormal Lung Sounds Using Deep Learning. In 2023 8th International Conference on Signal and Image Processing (ICSIP). IEEE. 2023:506-510.

[15] Nguyen T, Pernkopf F. Lung Sound Classification Using Co-Tuning and Stochastic Normalization. IEEE Trans Bio Med Eng. 2022;69:2872-2882.

[16] Perna D, Tagarelli A. Deep Auscultation: Predicting Respiratory Anomalies and Diseases via Recurrent Neural Networks. In 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS). IEEE. 2019:50-55.

[17] Wall C, Zhang L, Yu Y, Mistry K. Deep Recurrent Neural Networks With Attention Mechanisms for Respiratory Anomaly Classification. In: International Joint Conference on Neural Networks (IJCNN). IEEE.2021:1-8.

[18] Yuming Z, Wenlong X. Research on Classification of Respiratory Diseases Based on Multifeatures Fusion Cascade Neural Network. In: 11th International Conference on Information Technology in Medicine and Education (ITME). IEEE. 2021:298-301.

[19] Khan R, Khan SU, Saeed U, Koo IS. Auscultation-Based Pulmonary Disease Detection Through Parallel Transformation and Deep Learning. Bioengineering (Basel). 2024;11:586.

[20] Kochetov K, Putin E, Balashov M, Filchenkov A, Shalyto A. Noise Masking Recurrent Neural Network for Respiratory Sound Classification. In: International Conference on Artificial Neural Networks. Springer. 2018:208-217.

[21] Petmezas G, Cheimariotis GA, Stefanopoulos L, Rocha B, Paiva RP, et al. Automated Lung Sound Classification Using a Hybrid Cnn-Lstm Network and Focal Loss Function. Sensors. 2022;22:1232.

[22] Fraiwan M, Fraiwan L, Alkhodari M, Hassanin O. Recognition of Pulmonary Diseases From Lung Sounds Using Convolutional Neural Networks and Long Short-Term Memory. J Ambient Intell Hum Comput. 2022;13:4759-4771.

[23] Alqudah AM, Qazan S, Obeidat YM. Deep Learning Models for Detecting Respiratory Pathologies From Raw Lung Auscultation Sounds. Soft Comput. 2022;26:13405–13429.

[24] Kok XH, Anas Imtiaz SA, Rodriguez-Villegas E. A Novel Method for Automatic Identification of Respiratory Disease From Acoustic Recordings. In: 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE. 2019:2589-2592.

[25] Chambres G, Hanna P, Desainte-Catherine M. Automatic Detection of Patient With Respiratory Diseases Using Lung Sound Analysis. In 2018 International Conference on Content Based Multimedia Indexing (CBMI). IEEE. 2018:1-6.

[26] Jakovljević N, Lončar-Turukalo T. Hidden Markov Model Based Respiratory Sound Classification. In: 2017 International Conference on Biomedical and Health Informatics. Springer. 2017:39-43.

[27] Wu L, Li L. Investigating Into Segmentation Methods for Diagnosis of Respiratory Diseases Using Adventitious Respiratory Sounds. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). 2020:768-771.

[28] Hafdaoui H, Boudjelthia EA, Bouchakour S, Belhaouas N. Using Machine Learning for Analysis a Database Outdoor Monitoring of Photovoltaic System. Int J Integr Eng. 2022;14:275-280.

[29] Hafdaoui H, Belhaouas N, Assem H, Hadjrioua F, Madjoudj N. Compare Between the Performance of Different Technologies of PV Modules Using Artificial Intelligence Techniques. J Ren Energies. 2024:99-106.

[30] Hafdaoui H, Boudjelthia EA, Bouchakour S, Belhaouas N, et al. Use an Artificial Intelligence Method (Machine Learning) for Analysis of the Performance of Photovoltaic Systems. J Renew Energ. 2022;25:199-210.

[31] Hafdaoui H, Boudjelthia EA, Bouchakour S, Belhaouas N. Employing Machine Learning by Classification for Analysis of a Monitoring Database From a Photovoltaic Module. Desalin Water Treat 2022;279:147-151.

[32] Deschamps L, Devillaine L, Gaffet C, Lambert R, Aloui S, et al. Development of a Pre-Diagnosis Tool Based on Machine Learning Algorithms on the BHK Test to Improve the Diagnosis of Dysgraphia. Adv Artif Intell Mach Learn. 2021;1:111-130.

[33] Constantinescu C, Brad R. An Overview on Sound Features in Time and Frequency Domain. Int J Adv Stat IT&C Econ Life Sci. 2023;13:45-58.

[34] Purohit H, Tanabe R, Ichige K, Endo T, Nikaido Y, et al. MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection. 2019. Arxiv preprint: https://arxiv.org/pdf/1909.09347

[35] Fix E, Hodges JL. Discriminatory Analysis, Nonparametric Discrimination; 1951;57:233-238.

[36] Kohonen T. The self-organizing map. Proceedings of the IEEE. 2002;78:1464-1480.

[37] Vapnik V. The Nature of Statistical Learning Theory. Science+Business Media. Springer. 2013.

[38] Quinlan JR. Induction of Decision Trees. Mach Learn. 1986;1:81-106.

[39] Breiman L. Random Forests. Mach Learn. 2001;45:5-32.

[40] Babu N, Pruthviraja D, Mathew J. Enhancing Lung Acoustic Signals Classification With Eigenvectors-Based and Traditional Augmentation Methods. IEEE Access. 2024;12:87691-87700.