Automatic non-invasive Cough Detection based on Accelerometer and Audio Signals

Madhurananda Pahar, Igor Miranda, Andreas Diacon, Thomas Niesler

Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa

{mpahar, trn}@sun.ac.za

Abstract

We present an automatic non-invasive way of detecting cough events based on both accelerometer and audio signals. The acceleration signals are captured by a smartphone firmly attached to the patient's bed, using its integrated accelerometer. The audio signals are captured simultaneously by the same smartphone using an external microphone. We have compiled a manually-annotated dataset containing such simultaneously-captured acceleration and audio signals for approximately 6000 cough and 68000 non-cough events from 14 adult male patients in a tuberculosis clinic. LR, SVM and MLP are evaluated as baseline classifiers and compared with deep architectures such as CNN, LSTM, and Resnet50 using a leave-one-out cross-validation scheme. We find that the studied classifiers can use either acceleration or audio signals to distinguish between coughing and other activities including sneezing, throat-clearing, and movement on the bed with high accuracy. However, in all cases, the deep neural networks outperform the shallow classifiers by a clear margin and the Resnet50 offers the best performance by achieving an AUC exceeding 0.98 and 0.99 for acceleration and audio signals respectively. While audio-based classification consistently offers a better performance than acceleration-based classification, we observe that the difference is very small for the best systems. Since the acceleration signal requires less processing power, and since the need to record audio is sidestepped and thus privacy is inherently secured, and since the recording device is attached to the bed and not worn, an accelerometer-based highly accurate non-invasive cough detector may represent a more convenient and readily accepted method in long-term cough monitoring.

Index Terms: accelerometer, audio, cough detection, LR, SVM, MLP, Resnet, CNN, LSTM

1. Introduction

Coughing is a common symptom of respiratory disease and the forceful expulsion of air to clear up the airway [1]. It is distinctive in nature than the other audio events and is an important indicator used by physicians for clinical diagnosis and health monitoring in more than 100 respiratory diseases [2], including tuberculosis (TB) [3], asthma [4], pertussis [5] and COVID-19 [6]. Machine learning algorithms can be applied on the acoustic features extracted from the cough audio for automatic cough detection and classification [7–9]. However, using an audio-based monitoring system creates privacy issues [10], specially when the audio is captured by a smartphone [11, 12] and unique filtering processes might be required to preserve the privacy for continuous monitoring [13].

Thus, using accelerometers can be an alternative to the audio. Moreover, due to its much lower sampling rates, it requires less computing and processing power than the audio [14]. Automatic cough detection based on accelerometer measure-

ments is also possible when the device is placed on the patient's body and the acceleration signals are used for feature extraction [15]. An accelerometer is insensitive to environmental and background noise, thus it can be used in conjunction with other sensors such as microphones, ECG and thermistors [16]. Body-attached accelerometers have for example proved to be useful in detecting coughs when placed in contact with a patient's throat [17, 18] or at the laryngeal prominence (Adam's apple) [15]. A cough monitoring system using a contact microphone and an accelerometer attached to the participant's suprasternal (jugular) notch was developed in [19]. The participants moved around their homes while the cough audio and vibration was recorded. A similar ambulatory cough monitoring system, using an accelerometer attached to the skin of the participant's suprasternal notch using a bioclusive transparent dressing, was developed in [20]. Here, the recorded signal is transmitted to a receiver carried in a pocket or attached to a belt. Two accelerometers, one placed on the abdomen and the second on a belt wrapped at dorsal region, have been used to measure cough rate after cross-correlation of the two sensor signals [21]. Regression analysis, carried out on both audio and accelerometer signals gathered from 50 children, was able to achieve 97.8% specificity and 98.8% sensitivity when the accelerometer was placed in the centre of the abdomen between the navel and sternal notch [22]. Finally, multiple sensors, including ECG, thermistor, chest belt, accelerometer and audio microphones were used for cough detection in [23].

However, attaching an accelerometer to the patient's body is inconvenient and intrusive. We propose the monitoring of coughing based on the signals captured by the on-board accelerometer of an inexpensive consumer smartphone firmly attached to the patient's bed, as shown in Figure 1. This eliminates the need to wear a measuring equipment. The work presented here extends our previous study [24] by using three additional classifiers in the cough detection process and by comparing the performance between the proposed accelerometer-based classifiers and the baseline systems that classify audio signals of the same cough events. Such audio-based cough detection systems have been reported to discriminate between coughing and other sounds with areas under the ROC curve (AUCs) as high as 0.96 [7] and specificity as high as 99% [25]. Although we have found that the audio-based cough detection still outperforms the accelerometer-based detection, we demonstrate that the difference in the performance is narrow, as our best 50-layer residual architecture (Resnet50) based cough detector achieves an AUC of 0.996 for audio-based and 0.989 for accelerometer-based detection respectively. Thus we present an automatic non-invasive accelerometer-based cough detection system which can be used in long-term monitoring of patient's recovery process.

The structure of the reminder of this paper is as follows. Section 2 describes data collection while Section 3 details the features we extract from this data. The classifiers we use for ex-

perimentation are explained in Section 4 and the classification process itself is elaborated in Section 5. The results are presented in Section 6 and discussed in Section 7. Finally, Section 8 concludes the paper.

2. Dataset Preparation

2.1. Data collection

Data has been collected at a small 24h TB clinic near Cape Town, South Africa, which can accommodate approximately 10 staff and 30 patients. The clinic contains 8 wards and each ward has four beds, thus four patients at one time can be monitored inside a ward. The overall motivation of this study was to develop a practical method of automatic cough monitoring for the patients in this clinic, so that the progress of the recovery process can be monitored.

Figure 1 shows the recording setup, where an enclosure housing an inexpensive consumer smartphone is firmly attached to the back of the headboard of each bed in a ward. An Android application, developed specifically for this study, monitors the accelerometer and the audio signals. The on-board smartphone accelerometer has a sampling frequency of 100Hz. Although this sensor provides tri-axial measurements, we record only the vector magnitude. A BOYA BY-MM1 external microphone was used to capture audio signals (visible in Figure 1) and it has the sampling rate of 22.05 kHz. Using a simple energy detector, activity on either the acceleration or the audio channels triggers the simultaneous recording of both. This results in a dataset consisting of a sequence of non-overlapping tome intervals during which both acceleration and audio have been recorded.
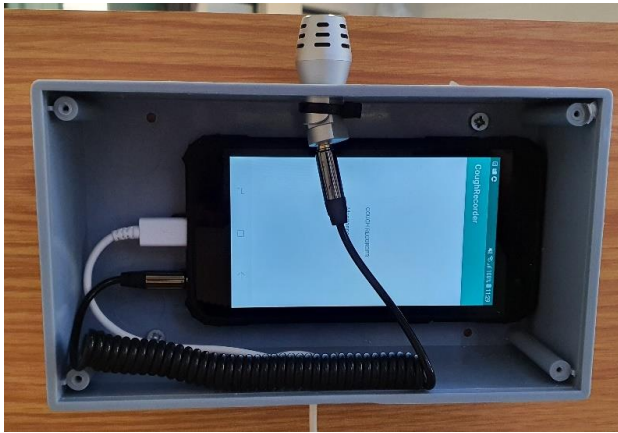
A Samsung Galaxy J4 smartphone with an inbuilt accelerometer connected to an external BOYA BY-MM1 cardioid microphone by a 3.5 mm audio jack

Figure 1: *Recording Equipment: A plastic enclosure housing an inexpensive smartphone running data gathering software is attached behind the headboard of each bed. The acceleration signal from the on-board accelerometer as well as the audio signal from the external microphone are monitored. Recording is triggered if activity is detected in either of these two signals.*

2.2. Data annotation

A large volume of both the audio and accelerometer data have been captured by using this energy-threshold-based detection for both audio and acceleration signals. Ceiling-mounted cameras recorded continuous video to assist with the data annotation process. The audio signals and the video recordings allowed the presence or absence of a cough in an event to be unambiguously confirmed during manual annotation. In the remainder of this paper, we will define an 'event' to be any interval of activity in either the accelerometer or the audio signals.

The non-cough events are generated mostly due to the patients getting in and out of the bed, moving while on the bed, sneezing and throat-clearing. Examples of the accelerometer magnitude signals for a cough event and a non-cough event (in this case moving on the bed) are shown in Figure 2. The spectrogram representations of these two signals are shown in Figure 3. Manual annotation was performed using the multimedia software tool ELAN [26], which allowed easy consolidation of the accelerometer, audio and video signals for accurate manual labelling.
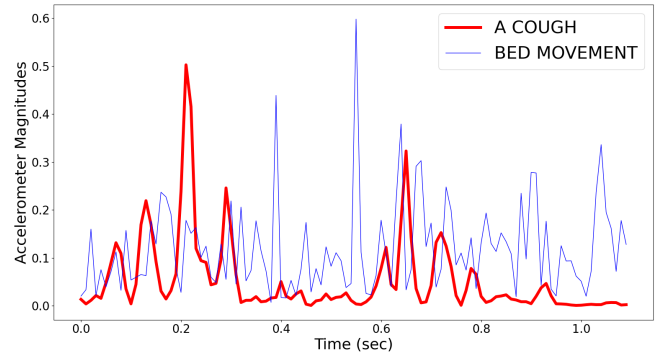

Figure 2: *The accelerometer magnitudes for a cough event (red) and a non-cough event (blue). In this case, the non-cough event was the patient moving on the bed.*

2.3. Final Dataset

The final dataset, summarised in Table 1, contains approximately 6000 coughs and 68000 non-coughs events from 14 adult male patients. Cough events are on average 1.90 sec long, with a standard deviation of 0.26 sec. Non-cough events are on average 1.70 sec long, with a standard deviation of 0.24 sec. The total lengths of the cough and non-cough events are 11397.60 sec (3.16 hours) and 115928.12 sec (32.20 hours) respectively. No other information regarding patients are recorded due to the ethical constraints of the study. This dataset was used to train and evaluate six classifiers, explained in Section 4 within a leave-one-out cross-validation framework, described in Section 5.

2.4. Dataset balancing

According to Table 1, cough events are outnumbered by the non-cough events in our dataset. This imbalance can affect the machine learning classifiers detrimentally [27, 28]. So, we have applied synthetic minority oversampling technique (SMOTE) to create new synthetic samples of the minor class instead of for example oversampling randomly [29, 30] while training the classifiers. This way we have addressed this class imbalance
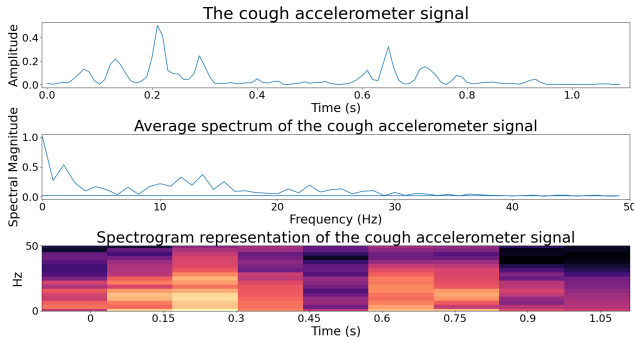
(a) *Accelerometer signal of the cough event.*
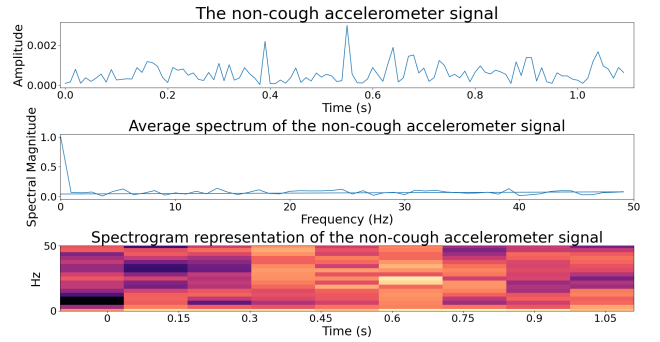
(b) *Accelerometer signal of the non-cough (bed-movement) event.*
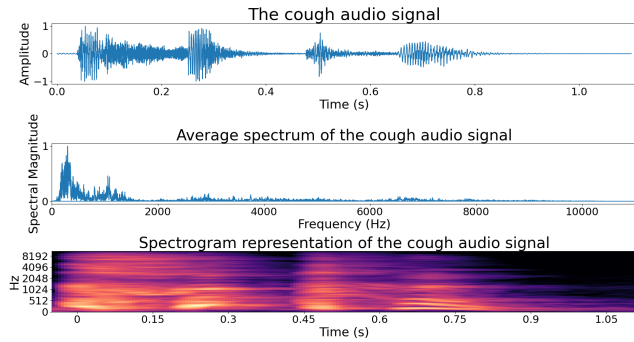
(c) *Audio signal of the cough event.*
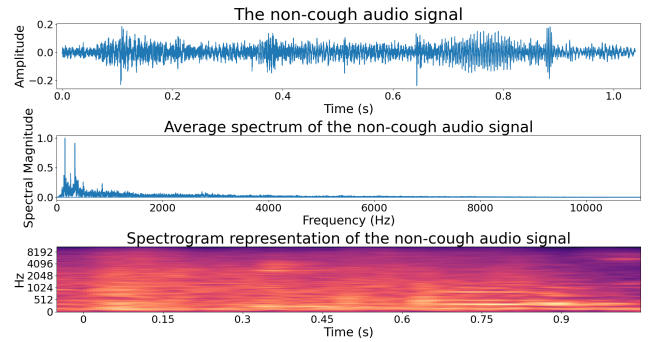
(d) *Audio signal of the non-cough (bed-movement) event.*

Figure 3: ***Spectrogram representation of the cough and non-cough events shown in Figure 2:*** *The cough event is shown on the left and the non-cough event (bed-movement) on the right. The accelerometer and audio signals are shown at the top and bottom row respectively. The audio signal has a higher sampling rate and thus contains more frequency and time-domain information than accelerometer measurements.*

Table 1: **Ground Truth Dataset Summary:** *'PATIENTS': list of the patients; 'COUGHS': number of confirmed cough events; 'NON COUGHS': number of confirmed events that are not coughs; 'COUGH TIME': total amount of time (in sec) for cough events; 'NON-COUGH TIME': total amount of time (in sec) for non-cough events.*

PATIENTS	COUGHS	NON COUGHS	COUGH TIME	NON-COUGH TIME
Patient 1	88	973	169.16	1660.67
Patient 2	63	1111	117.67	1891.92
Patient 3	469	11025	893.91	18797.32
Patient 4	109	9151	204.06	15596.71
Patient 5	97	7826	188.26	13344.98
Patient 6	192	12437	360.72	21197.35
Patient 7	436	14053	825.23	23953.15
Patient 8	368	2977	702.05	5077.89
Patient 9	2816	3856	5345.27	6569.32
Patient 10	649	2579	1236.84	4400.42
Patient 11	205	527	391.42	901.38
Patient 12	213	323	402.61	547.62
Patient 13	213	712	401.61	1211.75
Patient 14	82	455	158.77	777.64
TOTAL	**6000**	**68005**	**11397.6**	**115928.12**

for both the accelerometer and audio events. SMOTE has previously been successfully applied to cough detection and classification based on audio recordings [9, 24, 31].

3. Feature Extraction

The feature extraction process is illustrated for both accelerometer and audio signal in Figure 4.

3.1. Accelerometer features

Power spectrum, root mean square (RMS) value, kurtosis, moving averages and crest factor are extracted from the accelerometer magnitude samples. No de-noising has been applied prior to the feature extraction process. Power spectra [32] has been used to represent sensor data for input to classifiers, including neural networks, in several studies [33–36]. RMS [37] values from the sensor data have also been found to be useful features [38, 39]. The kurtosis has also been useful for machine learning applications as it indicates the prevalence of higher amplitudes [40]. Moving averages indicate the smoothed evolution of a signal over a time period and have been found to be useful features for sensor analysis [41]. Finally, the crest factor measures the ratio of the peak and the RMS signal amplitudes and have also been found to help machine learning prediction [42] including deep learning [43].

3.2. Audio features

Features such as mel-frequency cepstral coefficients (MFCCs), zero crossing rate (ZCR) and kurtosis are extracted from the audio signal. MFCCs are successfully used as features in audio
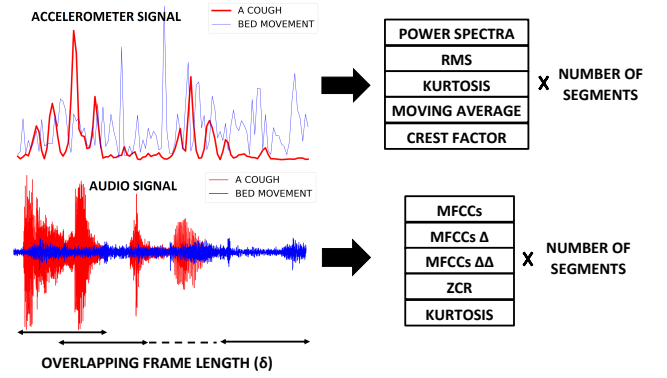

Figure 4: **Feature extraction for both accelerometer (top) and audio (bottom) signals:** *Both acceleration and audio signals of the events, shown in Figure 2 and 3, are split into a fixed number of overlapping frames. The length and number of these frames are Ψ and C for accelerometer signal & $\mathcal{F}$ and $\mathcal{S}$ for audio signal. For accelerometer measurements, the power spectrum, RMS, kurtosis, moving averages and crest factors of each frame are extracted. For audio signals, the MFCCs, MFCC velocity (Δ), MFCC acceleration ($\Delta\Delta$), ZCR and kurtosis are extracted. For the acceleration signal, this results in a feature matrix with dimensions $(C, \frac{\Psi}{2} + 5)$ while for the audio signal it generates a feature matrix with dimensions $(\mathcal{S}, 3\mathcal{M} + 2)$ where $\mathcal{M}$ is the number of MFCCs.*

analysis and especially in automatic speech recognition [44,45]. They can differentiate dry coughs from wet coughs [46] and also classify tuberculosis [47] and COVID-19 coughs [9, 48]. We have used the traditional MFCC extraction method considering higher resolution MFCCs along with the velocity (first-order difference, Δ) and acceleration (second-order difference, $\Delta\Delta$) as adding these has shown classifier improvement in the past [49]. The ZCR [50] is the number of times a signal changes its sign within a frame, and indicates the variability present in the signal. Finally, the kurtosis [51] indicates the prevalence of higher amplitudes in the samples of an audio signal. These features have been extracted by using the hyperparameters described in Table 2 for all cough and non-cough audio events.

3.3. Extraction Process

The features are extracted in a way that preserves the information regarding the beginning and the end of an event to allow time-domain patterns in the recordings to be discovered while maintaining the fixed input dimensionality, which is expected by the deep neural architectures such as a convolutional neural network (CNN).

For accelerometer signal, the frame length (Ψ) and number of segments (C) have been used as the feature extraction hyperparameters, shown in Table 2. The input feature matrix, fed to the classifiers mentioned in Section 4, has the dimension of $(C, \frac{\Psi}{2} + 5)$ whereas power spectra have $(\frac{\Psi}{2} + 1)$ coefficients.

From every event in audio signal, we extract a fixed number of features ($\mathcal{S}$) by distributing the fixed-length analysis frames ($\mathcal{F}$) uniformly over the time-interval of the cough. The extracted feature matrix from the audio always has the dimension of $(\mathcal{S}, 3\mathcal{M} + 2)$ for $\mathcal{M}$ number of MFCCs along with $\mathcal{M}$ number of velocity (Δ) and $\mathcal{M}$ number of acceleration ($\Delta\Delta$), as illustrated in Figure 4.

The frame skips are noted as δ in Figure 4. We divide the

number of samples in an event by the number of segments and take the next positive integer. For a 1.2 sec long audio event, the length of frame skip in samples is $\left\lceil \frac{1.2 \times 22050}{100} \right\rceil = \left\lceil \frac{26460}{100} \right\rceil = 265$ samples, as the audio sampling rate is 22.05 kHz.

Frame length (Ψ) used to extract features from acceleration signal is shorter than the frame length ($\mathcal{F}$) used to extract features from audio in this study (Table 2) and also traditionally [52]. This is because the accelerometer in the smartphone has a lower sampling rate of 100 Hz than the microphone 22.05 kHz (both shown in Figure 1) and longer frames lead to deteriorated performance as the signal properties can no longer be assumed to be stationary [53].

In contrast with the more conventionally applied fixed, non-overlapping frame rates, this way of extracting features ensures that the entire event is captured within a fixed number of frames, allowing especially the CNN to discover more useful temporal patterns and provide better classification performance. This particular method of feature extraction has also shown promising result in classifying COVID-19 coughs, breath and speech [9, 48].

4. Classifier Training

We have trained and evaluated six machine learning classifiers on both audio and accelerometer signal. Table 3 lists the classifier hyperparameters that were optimised during leave-one-out cross-validation.

First, we establish the baseline results by training and evaluating three shallow classifiers such as logistic regression (LR), support vector machine (SVM) and multilayer perceptron (MLP). Then, we improve the cough detection performance by implementing three deep neural network (DNN) classifiers such as CNN, long short-term memory (LSTM) and Resnet50.

LR models have outperformed other more complex classifiers such as classification trees, random forests, SVM in some clinical prediction tasks [3, 54, 55]. The gradient descent weight regularisation as well as lasso ($l1$ penalty) and ridge ($l2$ penalty) estimators [56, 57] were the hyperparameters, listed in Table 3, optimised inside the nested cross-validation during training. SVM classifiers have also performed well in both detecting [58, 59] and classifying [60] cough events in the past. The independent term in kernel functions is the hyperparameter optimised for the SVM classifier. An MLP, consisting multiple layers of neurons [61], is capable of learning non-linear relationships. It has produced promising results in discriminating influenza coughs from other coughs [62] in the past. MLP has also been applied to classify TB coughs [47, 59] and detect coughs in general [25, 63]. The penalty ratios, along with the number of neurons are used as the hyperparameters, optimised using the leave-one-out cross-validation process (Figure 7 and Section 5).

A CNN is a popular deep neural network architecture, primarily used in image classification [65], such as face recognition [66]. It has also performed well in classifying COVID-19 coughs, breath and speech [9, 48]. The CNN architecture [67, 68], shown in Figure 5, contains α_1 2D convolutional layers with kernel size α_2 and rectified linear units as activation functions. A dropout rate α_3 has been applied along with max-pooling, followed by α_4 dense layers with rectified linear units as activation functions, followed by another 8 dense layers, also with rectified linear units as activation functions. An LSTM model is a type of recurrent neural network which remembers previously-seen inputs when making its classification
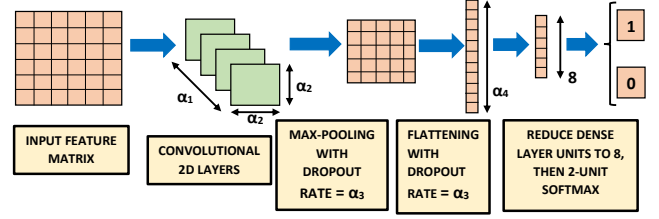

Figure 5: *CNN Classifier, trained and evaluated using leave-one-out cross-validation [64] on 14 patients. The results are shown in Table 4 and 5 for feature extraction hyperparameters mentioned in Table 2.*

Figure 6: *LSTM classifier, trained and evaluated using leave-one-out cross-validation [64] on 14 patients. The results are shown in Table 4 and 5 for feature extraction hyperparameters mentioned in Table 2.*
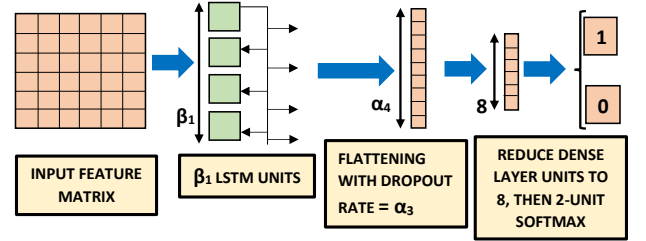
decision [69]. It has been successfully used in automatic cough detection [7, 24], and also in other types of acoustic event detection [70, 71] including COVID-19 coughs etc. [9, 48]. The hyperparameters optimised for the LSTM classifier [72] are mentioned in Table 3 and visually explained in Figure 6. The LSTM classifier, shown in Figure 6, contains β_1 LSTM units with rectified linear units as activation functions and a dropout rate α_3. Then α_4 dense layers have been applied with rectified linear units as activation functions, followed by another 8 dense layers also with rectified linear units as activation functions. For both CNN and LSTM classifiers, a final softmax function produces one output for a cough event (i.e. 1) and the other for a non-cough event (i.e. 0), shown in Figure 5 and 6. Features are fed into these two classifiers in batch size of ξ_1 for ξ_2 number of epochs. The 50-layer residual network (Resnet50) architecture (Table 1 of [73]) we trained and evaluated has a very deep architecture that contains skip layers and has performed even better than existing deep architectures such as VGGNet on image classification tasks on the dataset such as ILSVRC, the CIFAR10 dataset and the COCO object detection dataset [74]. This architecture has also performed the best in detecting COVID-19 signatures in coughs, breaths and speech [9, 48]. Due to extreme computation load, we have used the default Resnet50 structure mentioned in Table 1 of [73].

5. Classification Process

5.1. Hyperparameter optimisation

Hyperparameters for both the classifiers and feature extraction are optimised inside the leave-one-out cross-validation process and are listed in Table 2 and 3. Different phases of an event carries important information and our special way of feature

Table 2: ***Feature extraction hyperparameters for both accelerometer and audio signals.*** *For accelerometer, 16, 32, 64 samples i.e. 160, 320 and 640 msec long frames overlap in such a way that the number of these frames i.e. segments (5 and 10) are the same for all events in our dataset. Similarly for audio signals, MFCCs are varied between 13 and 65 & frames are varied between 256 samples (11.61 msec) and 4096 samples (185.76 msec) in such a way that the number of these extracted frames are varied between 50 to 150, fixed for4 all events in out dataset.*

FEATURE EXTRACTION HYPERPARAMETERS		
Accelerometer Hyperparameters		**Values**
Frame (Ψ)	Frame-length in samples, used to extract features	2^k where $k = 4, 5, 6$
Segments (C)	Number of frames extracted from the entire event	5, 10
Audio Hyperparameters		**Values**
MFCC ($\mathcal{M}$)	Number of lower-order MFCCs to keep	$13 \times k$, where $k = 1, \cdots, 5$
Frame ($\mathcal{F}$)	Frame-length in samples, used to extract features	2^k where $k = 8, \cdots, 12$
Segments ($\mathcal{S}$)	Number of frames extracted from the entire event	$10 \times k$, where $k = 5, 7, 10, 12, 15$

Table 3: ***Classifier hyperparameters***, *optimised using the leave-one-patient-out cross-validation.*

Hyperparameters	**Classifier**	**Range**
Regularisation strength (γ_1)	LR, SVM	10^i where, $i = -7, -6, \ldots, 6, 7$
$l1$ penalty (γ_2)	LR	0 to 1 in steps of 0.05
$l2$ penalty (γ_3)	LR, MLP	0 to 1 in steps of 0.05
Kernel coefficient (γ_4)	SVM	10^i where, $i = -7, -6, \ldots, 6, 7$
No. of neurons (γ_5)	MLP	10 to 100 in steps of 10
Batch size (ξ_1)	CNN & LSTM	2^k where $k = 6, 7, 8$
No. of epochs (ξ_2)	CNN & LSTM	10 to 250 in steps of 20
No. of convolutional filters (α_1)	CNN	3×2^k where $k = 3, 4, 5$
kernel size (α_2)	CNN	2 and 3
Dropout rate (α_3)	CNN & LSTM	0.1 to 0.5 in steps of 0.2
Dense layer size (α_4)	CNN & LSTM	2^k where $k = 4, 5$
LSTM units (β_1)	LSTM	2^k where $k = 6, 7, 8$
Learning rate (β_2)	LSTM	10^k where $k = -2, -3, -4$

extraction preserves the time-domain information. By varying the frame lengths and number of frames to extract, these information was varied. The spectral resolution was also varied by varying the number of lower order MFCCs to keep from the audio signal.

5.2. Cross-validation

All six classifiers have been trained and evaluated by using a leave-one-patient-out cross-validation scheme [64], as explained in Figure 7. Our dataset contains 14 patients and by using this cross-validation scheme we make the best use of our dataset, as patient's weight, coughing intensity, distance from the microphone can affect the accelerometer and audio signals and we were not allowed to collect those vital information due to the ethical constraints.

The Figure 7 shows that one patient is left out from 14 patients to be used for later independent testing. Then another patient is removed from the remaining 13 patients to be used as the development set where the hyperparameters, listed in Table 3,

are to be optimised. AUC has always been the optimisation criterion in this cross-validation. This entire procedure is repeated until all patients are used as an independent test sets in the outer loop. The final performance is evaluated by calculating and averaging AUC over these outer loops. The hyperparameters producing the highest AUC over these outer test sets are noted as the 'best hyperparameters' in Table 4 and 5. Performances produced by each classifier for each set of hyperparameters are noted by 'ID' in these tables.

6. Results

6.1. Accelerometer-based cough detection

Table 4 lists the performance achieved by all six classifiers for the hyperparameters mentioned in Table 2. The results from shallow classifiers are shown in systems C1 to C18 and from deep classifiers are shown in systems C19 to C36 in Table 4. These results are the averages over the 14 leave-one-patient-out testing partitions in the outer loop of cross-validation.
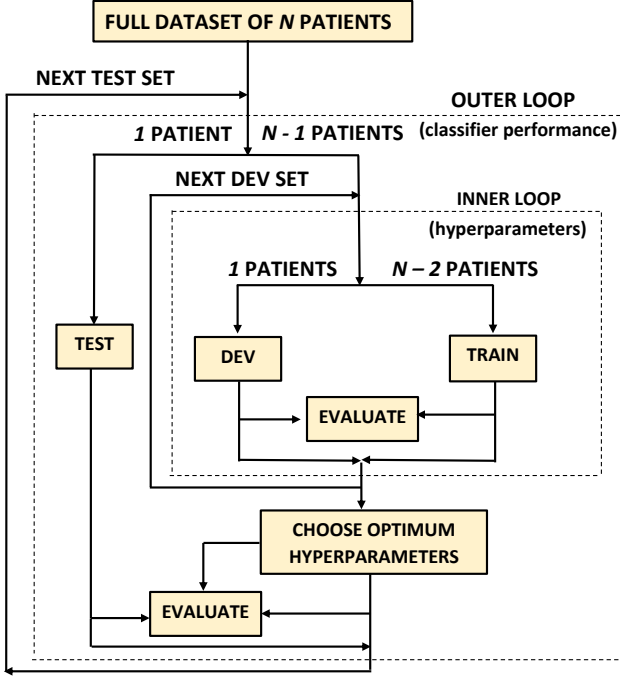
Figure 7: *Leave-one-out cross-validation used to train and evaluate all six classifiers. Here,* $N = 14$ *(Table 1). The development set (DEV) consisting 1 patient has been used to optimise the hyperparameters while training on the TRAIN set, consisted of 12 patients. The final evaluation of the classifiers in terms of the AUC occurs on the TEST set, consisting 1 patient.*

The shallow classifiers have provided the baseline classification performance. Table 4 shows that best performance is achieved by an LR classifier with an AUC of 0.8135 along with σ_{AUC} of 0.003, specificity of 81.42%, sensitivity of 81.28% and an accuracy of 81.35% (system C4). The SVM has produced an AUC of 0.8252 with σ_{AUC} of 0.003 and 80.91% specificity, 84.11% sensitivity and 82.51% accuracy while using ten 32 samples long frames (system C10). However, the AUC of 0.8587, accuracy of 85.67%, specificity of 84.47% and sensitivity of 86.89% have been achieved from the MLP classifier with 40 neurons and $l2$ penalty ratio of 0.7 using five 64 sample long frames (system C17) and this is the highest AUC achieved by a shallow classifier.

For the DNN classifiers, the lowest AUC of 0.9243 has been achieved from a CNN classifier in system C19 in Table 4. Table 4 also shows that the best-performing CNN uses 10 number of 64 samples (640 msec) long frames to achieve an AUC of 0.9499, accuracy of 85.82%, specificity of 80.91% and sensitivity of 90.73% (system C24). The optimal LSTM classifier achieves the slightly higher AUC of 0.9572 when using a frame length of 32 samples (320 msec) and 10 of such frames (system C28). However, the best performance is achieved by the Resnet50 architecture, with an AUC of 0.9888 after 50 epochs from 10 number of 32 samples (320 msec) long frames along with 96.71% accuracy, 94.09% specificity and 99.33% sensitivity (system C34).

Deep architectures have produced a higher AUCs and lower σ_{AUC} than the shallow classifiers on accelerometer-based classification task. Figure 8 shows the mean ROC curves for the optimal LR, SVM, MLP, CNN, LSTM and Resnet50, whose
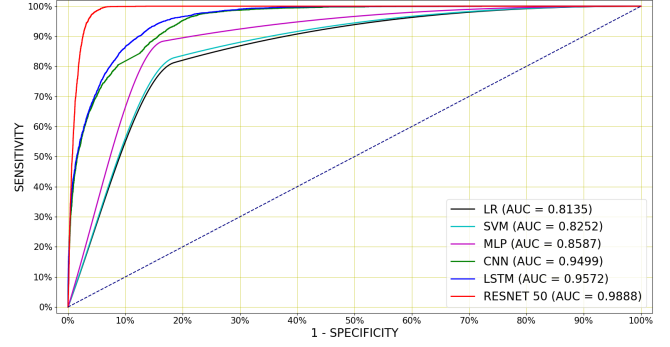

Figure 8: *Mean ROC curves for accelerometer-based cough detection, for the best performing classifiers whose hyperparameters are mentioned in Table 3. The best performance has been achieved from a Resnet50, which outperforms the LSTM and CNN over a wide range of operating points and has achieved the AUC of 0.9888 and the accuracy of 96.71%.*

configurations are shown in Table 4, where the means were calculated over the 14 cross-validation folds. The Resnet50 classifier is superior to all other classifiers over a wide range of operating points (Figure 8).

6.2. Audio-based cough detection

To place the performance of the accelerometer-based cough detection presented in the previous section into perspective, we have performed a matching set of experiments, this time using the audio signals to perform audio-based cough detection. These experiments are based on precisely the same events as the acceleration experiments, since our corpus contains both audio and acceleration signals for each.

Table 5 shows the best-three configurations for each of the six classifier architectures in Section 4. Again the results indicate that shallow classifiers (LR, SVM and MLP) achieve good classification scores.

LR achieved the highest AUC of 0.9129 with σ_{AUC} of 0.003 when using 26 MFCCs, 512 sample long frames and extracting 100 frames (system D1). The system has also generated the specificity of 87.52%, sensitivity of 87.71% and an accuracy of 87.61%. The SVM achieved an AUC of 0.9066 with σ_{AUC} of 0.003 for 26 MFCCs, 1024 sample long frames and extracting 120 frames (system D4). The system has also generated the specificity of 86.75%, sensitivity of 86.91% and an accuracy of 86.83%. An MLP has has produced the highest AUC of 0.9254 with σ_{AUC} of 0.002 for 39 MFCCs, 2048 sample long frames and extracting 120 frames (system D7). The system has also generated the specificity of 89.47%, sensitivity of 90.10% and an accuracy of 89.78%. This is the best performance achieved by the shallow classifiers.

Again, the DNN classifiers have outperformed the shallow classifiers by a large margin. The best LSTM classifier has produced the highest AUC of 0.9932 with σ_{AUC} of 0.002 for 26 MFCCs, 1024 sample long frames and extracting 70 frames (system D10). The system has also generated the specificity of 94.57%, sensitivity of 96.59% and an accuracy of 95.58%. The best CNN classifier has produced the highest AUC of 0.9944 with σ_{AUC} of 0.002 while features were extracted for 26 MFCCs, 1024 sample long frames and extracting 100 frames (system D13). The system has also generated the specificity of 93.24%, sensitivity of 97.88% and an accuracy of 95.56%.

Table 4: ***Accelerometer-based cough detection results.*** *The values are averaged over 14 cross-validation folds. DNN classifiers have outperformed the shallow classifiers by a wide margin and a Resnet50 produces the highest AUC of 0.9888.*

Classifier	ID	Best Feature Hyperparameters	Best Classifier Hyperparameters (Optimised inside nested cross-validation)	Spec	Sens	Acc	AUC	σ_{AUC}
				Performance				
LR	C1	$\Psi = 16, C = 5$	$\gamma_1 = 10^{-4}, \gamma_2 = 0.35, \gamma_3 = 0.65$	80.41%	80.28%	80.35%	0.8055	0.003
	C2	$\Psi = 16, C = 10$	$\gamma_1 = 10^{-2}, \gamma_2 = 0.55, \gamma_3 = 0.45$	80.25%	80.08%	80.16%	0.8058	0.003
	C3	$\Psi = 32, C = 5$	$\gamma_1 = 10^{2}, \gamma_2 = 0.2, \gamma_3 = 0.8$	80.39%	80.55%	80.47%	0.8072	0.003
	C4	$\Psi = 32, C = 10$	$\gamma_1 = 10^{-3}, \gamma_2 = 0.4, \gamma_3 = 0.6$	*81.42%*	*81.28%*	*81.35%*	*0.8135*	*0.003*
	C5	$\Psi = 64, C = 5$	$\gamma_1 = 10^{-1}, \gamma_2 = 0.25, \gamma_3 = 0.75$	80.22%	80.41%	80.31%	0.8119	0.003
	C6	$\Psi = 64, C = 10$	$\gamma_1 = 10^{-2}, \gamma_2 = 0.75, \gamma_3 = 0.25$	80.16%	80.32%	80.24%	0.8124	0.003
SVM	C7	$\Psi = 16, C = 5$	$\gamma_1 = 10^{3}, \gamma_4 = 10^{-3}$	80.71%	82.91%	81.81%	0.8202	0.003
	C8	$\Psi = 16, C = 10$	$\gamma_1 = 10^{-2}, \gamma_4 = 10^{2}$	80.22%	82.94%	81.58%	0.8248	0.003
	C9	$\Psi = 32, C = 5$	$\gamma_1 = 10^{3}, \gamma_4 = 10^{-2}$	80.41%	82.97%	81.69%	0.8212	0.003
	C10	$\Psi = 32, C = 10$	$\gamma_1 = 10^{-1}, \gamma_4 = 10^{-1}$	*80.91%*	*84.11%*	*82.51%*	*0.8252*	*0.003*
	C11	$\Psi = 64, C = 5$	$\gamma_1 = 10^{-4}, \gamma_4 = 10^{-3}$	80.28%	84.35%	82.31%	0.8245	0.003
	C12	$\Psi = 64, C = 10$	$\gamma_1 = 10^{2}, \gamma_4 = 10^{-4}$	80.55%	82.78%	81.68%	0.8251	0.003
MLP	C13	$\Psi = 16, C = 5$	$\gamma_3 = 0.55, \gamma_5 = 30$	82.37%	86.95%	84.68%	0.8507	0.003
	C14	$\Psi = 16, C = 10$	$\gamma_3 = 0.45, \gamma_5 = 50$	83.24%	87.08%	85.16%	0.8558	0.003
	C15	$\Psi = 32, C = 5$	$\gamma_3 = 0.35, \gamma_5 = 70$	83.55%	87.41%	85.47%	0.8552	0.003
	C16	$\Psi = 32, C = 10$	$\gamma_3 = 0.4, \gamma_5 = 20$	82.18%	86.05%	84.12%	0.8424	0.003
	C17	$\Psi = 64, C = 5$	$\gamma_3 = 0.7, \gamma_5 = 40$	*84.47%*	*86.89%*	*85.67%*	*0.8587*	*0.003*
	C18	$\Psi = 64, C = 10$	$\gamma_3 = 0.35, \gamma_5 = 30$	83.45%	86.84%	84.64%	0.8499	0.003
CNN	C19	$\Psi = 16, C = 5$	$\alpha_1=48, \alpha_2=2, \alpha_3=0.1, \alpha_4=32, \xi_1=128, \xi_2=210$	83.47%	85.62%	84.55%	0.9243	0.002
	C20	$\Psi = 16, C = 10$	$\alpha_1=24, \alpha_2=2, \alpha_3=0.3, \alpha_4=32, \xi_1=256, \xi_2=110$	83.76%	87.56%	85.66%	0.9358	0.002
	C21	$\Psi = 32, C = 5$	$\alpha_1=96, \alpha_2=2, \alpha_3=0.3, \alpha_4=32, \xi_1=128, \xi_2=150$	76.98%	91.96%	84.47%	0.9272	0.002
	C22	$\Psi = 32, C = 10$	$\alpha_1=48, \alpha_2=2, \alpha_3=0.3, \alpha_4=16, \xi_1=256, \xi_2=110$	84.09%	86.41%	85.25%	0.9324	0.002
	C23	$\Psi = 64, C = 5$	$\alpha_1=48, \alpha_2=2, \alpha_3=0.5, \alpha_4=32, \xi_1=256, \xi_2=230$	85.47%	87.15%	86.31%	0.9339	0.002
	C24	$\Psi = 64, C = 10$	*$\alpha_1=96, \alpha_2=2, \alpha_3=0.3, \alpha_4=32, \xi_1=128, \xi_2=170$*	*80.91%*	*90.73%*	*85.82%*	*0.9499*	*0.002*
LSTM	C25	$\Psi = 16, C = 5$	$\beta_1 = 128, \beta_2 = 0.0001, \alpha_3 = 0.3, \alpha_4 = 32, \xi_1 = 256, \xi_2 = 210$	84.34%	90.82%	87.58%	0.9444	0.002
	C26	$\Psi = 16, C = 10$	$\beta_1 = 128, \beta_2 = 0.01, \alpha_3 = 0.1, \alpha_4 = 32, \xi_1 = 128, \xi_2 = 110$	85.37%	91.27%	88.32%	0.9504	0.002
	C27	$\Psi = 32, C = 5$	$\beta_1 = 256, \beta_2 = 0.001, \alpha_3 = 0.3, \alpha_4 = 16, \xi_1 = 128, \xi_2 = 130$	79.92%	94.31%	87.11%	0.9457	0.002
	C28	$\Psi = 32, C = 10$	*$\beta_1 = 128, \beta_2 = 0.001, \alpha_3 = 0.1, \alpha_4 = 32, \xi_1 = 256, \xi_2 = 150$*	*86.41%*	*92.05%*	*89.21%*	*0.9572*	*0.002*
	C29	$\Psi = 64, C = 5$	$\beta_1 = 256, \beta_2 = 0.001, \alpha_3 = 0.3, \alpha_4 = 16, \xi_1 = 128, \xi_2 = 190$	84.57%	92.79%	88.68%	0.954	0.002
	C30	$\Psi = 64, C = 10$	$\beta_1 = 128, \beta_2 = 0.01, \alpha_3 = 0.5, \alpha_4 = 32, \xi_1 = 256, \xi_2 = 230$	86.21%	89.13%	87.66%	0.9489	0.002
Resnet50	C31	$\Psi = 16, C = 5$	Default Resnet50 (Table 1 in [73])	93.81%	97.09%	95.43%	0.9802	0.002
	C32	$\Psi = 16, C = 10$	,,	94.12%	98.58%	96.35%	0.9812	0.002
	C33	$\Psi = 32, C = 5$	,,	94.29%	98.79%	96.54%	0.9810	0.002
	C34	$\Psi = 32, C = 10$	,,	***94.09%***	***99.33%***	***96.71%***	***0.9888***	***0.002***
	C35	$\Psi = 64, C = 5$	,,	94.71%	98.23%	96.35%	0.9854	0.002
	C36	$\Psi = 64, C = 10$	,,	95.07%	97.89%	96.46%	0.9884	0.002

However, the highest AUC of 0.9957 has been achieved again from a Resnet50 classifier with a σ_{AUC} of 0.001 for 26 MFCCs, 1024 sample (i.e. 46.44 msec) long frames and extracting 100 frames from the entire event (system D16). This system has also achieved a specificity of 96.74% and a sensitivity of 99.5% along with the accuracy of 98.13%.

Again, deep architectures have produced a higher AUCs and lower σ_{AUC} than the shallow classifiers on audio-based classification task. These best results for audio-based classification are shown in Figure 9. Table 5 also indicate that the number of MFCCs has been varied between 13 and 65, although the best performance was achieved by using 26 and 39 MFCCs. Using the frame length of 1024 and extracting 100 frames from the events has provided the best performance for most of the classifiers.
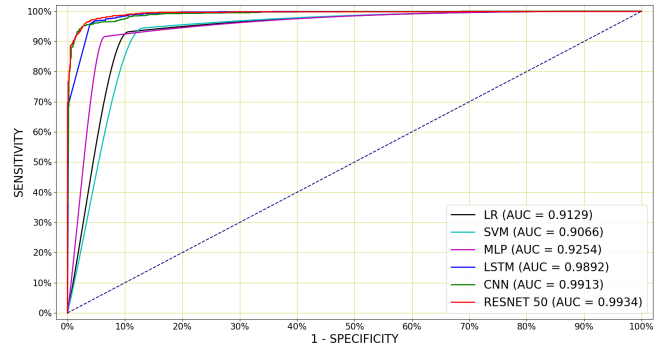

Figure 9: ***Mean ROC curves for audio-based cough detection****, for the best performing classifiers whose hyperparameters are mentioned in Table 3. The best performance has been achieved from a Resnet50 and similar performances have been achieved from a CNN and LSTM. The best Resnet50 produces the AUC of 0.9957 and the accuracy of 98.13%.*

Table 5: ***Audio-based cough detection results.*** *The values are averaged over 14 cross-validation folds and the best-three performances of each classifier are shown. All classifiers have performed well in detecting coughs but DNN classifiers have performed particularly well and their performances are very close to each other.*

Classifier	ID	Best Feature Hyperparameters	Best Classifier Hyperparameters (Optimised inside nested cross-validation)	Performance				
				Spec	Sens	Acc	AUC	σ_{AUC}
LR	D1	$\mathcal{M}=26, \mathcal{F}=512, \mathcal{S}=100$	$\gamma_1=10^{-3}, \gamma_2=0.25, \gamma_3=0.75$	87.52%	87.71%	87.61%	0.9129	0.003
	D2	$\mathcal{M}=39, \mathcal{F}=1024, \mathcal{S}=70$	$\gamma_1=10^2, \gamma_2=0.4, \gamma_3=0.6$	87.31%	87.41%	87.36%	0.9358	0.003
	D3	$\mathcal{M}=26, \mathcal{F}=1024, \mathcal{S}=100$	$\gamma_1=10^{-4}, \gamma_2=0.55, \gamma_3=0.45$	87.14%	87.28%	87.21%	0.9272	0.003
SVM	D4	$\mathcal{M}=26, \mathcal{F}=1024, \mathcal{S}=120$	$\gamma_1=10^{-2}, \gamma_4=10^3$	86.75%	86.91%	86.83%	0.9066	0.003
	D5	$\mathcal{M}=26, \mathcal{F}=512, \mathcal{S}=100$	$\gamma_1=10^2, \gamma_4=10^{-3}$	86.61%	86.68%	86.64%	0.9058	0.003
	D6	$\mathcal{M}=39, \mathcal{F}=1024, \mathcal{S}=100$	$\gamma_1=10^3, \gamma_4=10^{-3}$	86.40%	86.54%	86.47%	0.9017	0.003
MLP	D7	$\mathcal{M}=39, \mathcal{F}=2048, \mathcal{S}=120$	$\gamma_3=0.35, \gamma_5=30$	89.47%	90.10%	89.78%	0.9254	0.002
	D8	$\mathcal{M}=26, \mathcal{F}=1024, \mathcal{S}=70$	$\gamma_3=0.4, \gamma_5=50$	89.55%	89.76%	89.66%	0.9214	0.003
	D9	$\mathcal{M}=39, \mathcal{F}=1024, \mathcal{S}=100$	$\gamma_3=0.6, \gamma_5=40$	88.78%	89.04%	88.91%	0.9205	0.003
LSTM	D10	$\mathcal{M}=26, \mathcal{F}=1024, \mathcal{S}=70$	$\beta_1=128, \beta_2=0.001, \alpha_3=0.3, \alpha_4=32, \xi_1=256, \xi_2=210$	94.57%	96.59%	95.58%	0.9932	0.002
	D11	$\mathcal{M}=39, \mathcal{F}=1024, \mathcal{S}=100$	$\beta_1=128, \beta_2=0.001, \alpha_3=0.3, \alpha_4=16, \xi_1=256, \xi_2=130$	94.21%	94.21%	96.43%	0.9904	0.002
	D12	$\mathcal{M}=26, \mathcal{F}=2048, \mathcal{S}=120$	$\beta_1=128, \beta_2=0.01, \alpha_3=0.3, \alpha_4=32, \xi_1=128, \xi_2=170$	93.95%	96.25%	95.10%	0.9857	0.002
CNN	D13	$\mathcal{M}=26, \mathcal{F}=1024, \mathcal{S}=100$	$\alpha_1=48, \alpha_2=2, \alpha_3=0.3, \alpha_4=32, \xi_1=256, \xi_2=90$	93.24%	97.88%	95.56%	0.9944	0.002
	D14	$\mathcal{M}=13, \mathcal{F}=512, \mathcal{S}=70$	$\alpha_1=24, \alpha_2=2, \alpha_3=0.3, \alpha_4=16, \xi_1=256, \xi_2=170$	92.18%	98.74%	95.46%	0.9891	0.002
	D15	$\mathcal{M}=39, \mathcal{F}=2048, \mathcal{S}=120$	$\alpha_1=48, \alpha_2=2, \alpha_3=0.1, \alpha_4=32, \xi_1=256, \xi_2=130$	92.78%	97.56%	95.17%	0.9872	0.002
Resnet50	D16	$\mathcal{M}=26, \mathcal{F}=1024, \mathcal{S}=100$	Default Resnet50 (Table 1 in [73])	**96.74%**	**99.50%**	**98.13%**	**0.9957**	**0.001**
	D17	$\mathcal{M}=39, \mathcal{F}=1024, \mathcal{S}=70$	,,	96.55%	96.95%	96.75%	0.9912	0.001
	D18	$\mathcal{M}=39, \mathcal{F}=512, \mathcal{S}=100$	,,	94.70%	96.46%	95.58%	0.9891	0.002

7. Discussion

The results shown in Table 4 and 5 indicate that audio-based cough detection is consistently more accurate than accelerometer-based classification. However it is interesting to note that the performances offered by the two alternatives are fairly close. In fact, the deep architectures like Resnet50 offer almost equal performance for audio-based and accelerometer-based cough detection. It also seems that the CNN and LSTM find it easier to classify cough events based on audio rather than accelerometer signal. We postulate that this is due to the limited range of time and frequency information contained in accelerometer data, which is in turn due to the lower accelerometer sampling rate.

For acceleration signals, the extraction of 10 frames each with a length of 640 ms produced the best result. For the audio, the extraction of 100 frames each with a length of 46.44 ms provided the optimal performance. These audio frame lengths are close to those traditionally used for feature extraction in automatic speech recognition. We also note that the performances of the deep classifiers are consistently better than those offered by the baseline shallow classifiers for both types of signals. Although the datasets differ, our system also appears to improve on recent work using the accelerometer integrated into a smartwatch [75] and detecting cough audio among other audio events such as sneeze, speech, noise etc. [7].

8. Conclusion and Future Work

An automatic non-invasive machine learning based cough detector is able to accurately discriminate between the accelerometer and audio signals due to coughing and due to other movements as captured by a consumer smartphone attached to a patient's bed.

We have trained and evaluated six classifiers including three shallow classifiers such as logistic regression (LR), support vector machine (SVM) and multilayer perceptron (MLP) and three deep neural network (DNN) classifiers such as convolutional neural networks (CNN), long-short-term-memory (LSTM) networks, and a 50-layer residual-based neural network architecture (Resnet50). A specially-compiled corpus of manually-annotated acceleration and audio events, including approximately 6000 cough and 68000 non-cough events such as sneezing, throat-clearing and getting in and out of the bed, gathered from 14 adult male patients in a small TB clinic was used to train and evaluate these classifiers by using a leave-one-out cross-validation scheme. For accelerometer-based classification, the best system uses a Resnet50 architecture and produces an AUC of 0.9888 as well as a 96.71% accuracy, 94.09% specificity and 99.33% sensitivity while features were extracted from ten 32 sample (320 msec) long frames. This demonstrates that it is possible to discriminate between cough events and other non-cough events by using very deep architectures such as a Resnet50; based in signals gathered from an accelerometer that is not attached to the patient's body, but rather to the headboard of the patient's bed.

We have also compared this accelerometer-based cough detection with audio-based cough detection for the same cough and non-cough events. For audio-based classification, the best result has also been achieved from a Resnet50 with the highest AUC of 0.9957. This shows that the accelerometer-based cough detection is almost equally accurate as audio-based classification while using very deep architectures such as a Resnet50. Shallow classifiers and DNN such as CNN and LSTM however perform better in classifying cough events on audio signals rather than accelerometer signals, as audio signal carries more dynamic and diverse frequency content.

Accelerometer-based detection of cough events has successfully been considered before due to its lower sampling rates and lesser demand of high processing power, however only by using sensors worn by the subjects, which is intrusive and can be inconvenient in some respects. This study shows that excellent discrimination is also possible when the sensor is attached

to the patient's bed, thus providing a less intrusive and cumbersome solution. Furthermore, since the use of the acceleration signal avoids the need to gather audio, privacy is inherently protected. Therefore, the use of a bed-mounted accelerometer inside an inexpensive consumer smartphone may represent a more convenient, cost-effective and readily accepted method of long-term patient cough monitoring.

In the future, we will be attempting to optimise some of the Resnet50 metaparameters and fuse both audio and accelerometer signal to achieve higher specificity and accuracy in cough detection. We are also in the process of applying the proposed system in an automatic non-invasive cough monitoring system. We also note, the manually annotated cough events sometimes contains multiple bursts of cough onsets and we are currently investigating automatic methods that allow such bursts within a cough event to be identified.

9. Acknowledgements

10. Conflict of interest

The authors declare that they have no conflict of interest.

11. References

[1] J Korpáš, J Sadloňová, and M Vrabec, "Analysis of the cough sound: an overview," *Pulmonary Pharmacology*, vol. 9, no. 5-6, pp. 261–268, 1996.

[2] J Knocikova, J Korpas, M Vrabec, and M Javorka, "Wavelet analysis of voluntary cough sound in patients with respiratory diseases," *Journal of Physiology and Pharmacology*, vol. 59, no. Suppl 6, pp. 331–40, 2008.

[3] GHR Botha, G Theron, RM Warren, M Klopper, K Dheda, PD Van Helden, and TR Niesler, "Detection of tuberculosis by automatic cough sound analysis," *Physiological Measurement*, vol. 39, no. 4, pp. 045005, 2018.

[4] Mahmood Al-khassaweneh and Ra'ed Bani Abdelrahman, "A signal processing approach for the diagnosis of asthma from cough sounds," *Journal of Medical Engineering & Technology*, vol. 37, no. 3, pp. 165–171, 2013.

[5] Renard Xaviero Adhi Pramono, Syed Anas Imtiaz, and Esther Rodriguez-Villegas, "A cough-based algorithm for automatic diagnosis of pertussis," *PloS one*, vol. 11, no. 9, pp. e0162128, 2016.

[6] Angelo Carfì, Roberto Bernabei, Francesco Landi, et al., "Persistent symptoms in patients after acute COVID-19," *JAMA*, vol. 324, no. 6, pp. 603–605, 2020.

[7] Igor DS Miranda, Andreas H Diacon, and Thomas R Niesler, "A comparative study of features for acoustic cough detection using deep architectures," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 2601–2605.

[8] Jordi Laguarta, Ferran Hueto, and Brian Subirana, "COVID-19 Artificial Intelligence Diagnosis using only Cough Recordings," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 275–281, 2020.

[9] Madhurananda Pahar, Marisa Klopper, Robin Warren, and Thomas Niesler, "COVID-19 cough classification using machine learning and global smartphone recordings," *Computers in Biology and Medicine*, vol. 135, pp. 104572, 2021.

[10] Francine Chen, John Adcock, and Shruti Krishnagiri, "Audio privacy: reducing speech intelligibility while preserving environmental sounds," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 733–736.

[11] Yu-Chih Tung and Kang G Shin, "Exploiting Sound Masking for Audio Privacy in Smartphones," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 257–268.

[12] Stephen Xia and Xiaofan Jiang, "PAMS: Improving Privacy in Audio-Based Mobile Systems," in *Proceedings of the 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, 2020, pp. 41–47.

[13] Daniyal Liaqat, Ebrahim Nemati, Mahbubur Rahman, and Jilong Kuang, "A method for preserving privacy during audio recordings by filtering speech," in *2017 IEEE Life Sciences Conference (LSC)*. IEEE, 2017, pp. 79–82.

[14] Tracey J Mehigan, "Harnessing accelerometer technology for inclusive mobile learning," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2009, pp. 1–2.

[15] Helia Mohammadi, Ali-Akbar Samadani, Catriona Steele, and Tom Chau, "Automatic discrimination between cough and non-cough accelerometry signal artefacts," *Biomedical Signal Processing and Control*, vol. 52, pp. 394–402, 2019.

[16] Paul Munyard, Carolyn Busst, Ron Logan-Sinclair, and Andrew Bush, "A new device for ambulatory cough recording," *Pediatric Pulmonology*, vol. 18, no. 3, pp. 178–186, 1994.

[17] Michael Coyle, P Alexander Derchak, and Lance Jonathan Myers, "Systems and methods for monitoring cough," June 1 2010, US Patent 7,727,161.

[18] Jingqi Fan, German Comina, Robert Gilman, Jose Lopez, and Brian H Tracey, "Cough monitoring for pulmonary tuberculosis using combined microphone/accelerometer measurements," *The Journal of the Acoustical Society of America*, vol. 135, no. 4, pp. 2268–2268, 2014.

[19] Lucy Pavesi, Subbu Subburaj, and Kerri Porter-Shaw, "Application and validation of a computerized cough acquisition system for objective monitoring of acute cough: a meta-analysis," *Chest*, vol. 120, no. 4, pp. 1121–1128, 2001.

[20] Ian M Paul, Kitman Wai, Steven J Jewell, Michele L Shaffer, and Vasundara V Varadan, "Evaluation of a new self-contained, ambulatory, objective cough monitor," *Cough*, vol. 2, no. 1, pp. 7, 2006.

[21] Johnny Yat Ming Chan, Stephen Anthony Tunnell, and Joshua Aaron Lawrence Jacobs, "Systems, methods and kits for measuring cough and respiratory rate using an accelerometer," Sept. 4 2014, US Patent App. 13/783,257.

[22] Kota Hirai, Hideyuki Tabata, Mariko Hirayama, Tohru Kobayashi, Yasumasa Oh, and Hiroyuki Mochizuki, "A new method for objectively evaluating childhood nocturnal cough," *Pediatric Pulmonology*, vol. 50, no. 5, pp. 460–468, 2015.

[23] Thomas Drugman, Jerome Urbain, Nathalie Bauwens, Ricardo Chessini, Carlos Valderrama, Patrick Lebecque, and Thierry Dutoit, "Objective Study of Sensor Relevance for Automatic Cough Detection," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 699–707, 2013.

[24] Madhurananda Pahar, Igor Miranda, Andreas Diacon, and Thomas Niesler, "Deep Neural Network based Cough Detection using Bed-mounted Accelerometer Measurements," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8002–8006.

[25] Justice Amoh and Kofi Odame, "DeepCough: A deep convolutional neural network in a wearable cough detection system," in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2015, pp. 1–4.

[26] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes, "ELAN: a professional framework for multimodality research," in *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006.

[27] Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th International Conference on Machine learning*, 2007, pp. 935–942.

[28] Bartosz Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[29] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "SMOTE: synthetic minority oversampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[30] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[31] Anthony Windmon, Mona Minakshi, Pratool Bharti, Sriram Chellappan, Marcia Johansson, Bradlee A Jenkins, and Ponrathi R Athilingam, "Tussiswatch: A smart-phone system to identify cough episodes as early symptoms of chronic obstructive pulmonary disease and congestive heart failure," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1566–1573, 2018.

[32] Christopher Bingham, M Godfrey, and J Tukey, "Modern techniques of power spectrum estimation," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 56–66, 1967.

[33] Tzyy-Ping Jung, Scott Makeig, Magnus Stensmo, and Terrence J Sejnowski, "Estimating alertness from the eeg power spectrum," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 1, pp. 60–69, 1997.

[34] Lutfiye Durak and Orhan Arikan, "Short-time fourier transform: two fundamental properties and an optimal implementation," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1231–1242, 2003.

[35] RK Sinha, "Artificial neural network detects changes in electroencephalogram power spectrum of different sleep-wake states in an animal model of heat stress," *Medical and Biological Engineering and Computing*, vol. 41, no. 5, pp. 595–600, 2003.

[36] Bo Liang, SD Iwnicki, and Yunshi Zhao, "Application of power spectrum, cepstrum, higher order spectrum and neural network analyses for induction motor fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 39, no. 1-2, pp. 342–360, 2013.

[37] Norman Levinson, "The wiener rms (root mean square) error criterion in filter design and prediction," *J. math. Phys*, vol. 25, no. 4, pp. 261–278, 1947.

[38] Robert L Lux, Christopher Todd Sower, Nancy Allen, Susan P Etheridge, Martin Tristani-Firouzi, and Elizabeth V Saarel, "The application of root mean square electrocardiography (rms ecg) for the detection of acquired and congenital long qt syndrome," *PloS one*, vol. 9, no. 1, pp. e85689, 2014.

[39] Jordon Gilmore, Monsur Islam, Josie Duncan, Rucha Natu, and Rodrigo Martinez-Duarte, "Assessing the importance of the root mean square (rms) value of different waveforms to determine the strength of a dielectrophoresis trapping force," *Electrophoresis*, vol. 38, no. 20, 2017.

[40] Xu Zhang, Paul E Barkhaus, William Zev Rymer, and Ping Zhou, "Machine learning for supporting diagnosis of amyotrophic lateral sclerosis using surface electromyogram," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 1, pp. 96–103, 2013.

[41] Xiaohui Yuan, Qingxiong Tan, Xiaohui Lei, Yanbin Yuan, and Xiaotao Wu, "Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine," *Energy*, vol. 129, pp. 122–137, 2017.

[42] Julien Lepine, Vincent Rouillard, and Michael Sek, "On the use of machine learning to detect shocks in road vehicle vibration signals," *Packaging Technology and Science*, vol. 30, no. 8, pp. 387–398, 2017.

[43] Lei Ren, Jin Cui, Yaqiang Sun, and Xuejun Cheng, "Multi-bearing remaining useful life collaborative prediction: A deep learning approach," *Journal of Manufacturing Systems*, vol. 43, pp. 248–256, 2017.

[44] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, and Kong-Pang Pun, "An efficient MFCC extraction method in speech recognition," in *IEEE International Symposium on Circuits and Systems*, 2006.

[45] Madhurananda Pahar and Leslie S Smith, "Coding and Decoding Speech using a Biologically Inspired Coding System," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 3025–3032.

[46] Hanieh Chatrzarrin, Amaya Arcelus, Rafik Goubran, and Frank Knoefel, "Feature extraction for the differentiation of dry and wet cough sounds," in *IEEE International Symposium on Medical Measurements and Applications*. IEEE, 2011.

[47] Madhurananda Pahar, Marisa Klopper, Byron Reeve, Grant Theron, Robin Warren, and Thomas Niesler, "Automatic Cough Classification for Tuberculosis Screening in a Real-World Environment," *arXiv preprint arXiv:2103.13300*, 2021.

[48] Madhurananda Pahar, Marisa Klopper, Robin Warren, and Thomas Niesler, "COVID-19 Detection in Cough, Breath and Speech using Deep Transfer Learning and Bottleneck Features," *arXiv preprint arXiv:2104.02477*, 2021.

[49] M. M. Azmy, "Feature extraction of heart sounds using velocity and acceleration of MFCCs based on support vector machines," in *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2017, pp. 1–4.

[50] RG Bachu, S Kopparthi, B Adapa, and Buket D Barkana, "Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy," pp. 279–282, 2010.

[51] Lawrence T DeCarlo, "On the meaning and use of kurtosis.," *Psychological Methods*, vol. 2, no. 3, pp. 292, 1997.

[52] Gen Takahashi, Takeshi Yamada, Shoji Makino, and Nobutaka Ono, "Acoustic scene classification using deep neural network and frame-concatenated acoustic feature," *Detection and Classification of Acoustic Scenes and Events*, 2016.

[53] Cyril Joder, Slim Essid, and Gaël Richard, "Temporal Integration for Audio Classification With Application to Musical Instrument Classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 174–186, 2009.

[54] Evangelia Christodoulou, Jie Ma, Gary S Collins, Ewout W Steyerberg, Jan Y Verbakel, and Ben Van Calster, "A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models," *Journal of Clinical Epidemiology*, vol. 110, pp. 12–22, 2019.

[55] Saskia Le Cessie and Johannes C Van Houwelingen, "Ridge estimators in logistic regression," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 41, no. 1, pp. 191–201, 1992.

[56] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 477–485.

[57] Hiroshi Yamashita and Hiroshi Yabe, "An interior point method with a primal-dual quadratic barrier penalty function for nonlinear optimization," *SIAM Journal on Optimization*, vol. 14, no. 2, pp. 479–499, 2003.

[58] Vikrant Bhateja, Ahmad Taquee, and Dilip Kumar Sharma, "Pre-Processing and Classification of Cough Sounds in Noisy Environment using SVM," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*. IEEE, 2019, pp. 822–826.

[59] Brian H Tracey, Germán Comina, Sandra Larson, Marjory Bravard, José W López, and Robert H Gilman, "Cough detection algorithm for monitoring patient recovery from pulmonary tuberculosis," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 6017–6020.

[60] Roneel V Sharan, Udantha R Abeyratne, Vinayak R Swarnkar, and Paul Porter, "Cough sound analysis for diagnosing croup in pediatric patients using biologically inspired features," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2017, pp. 4578–4581.

[61] H Taud and JF Mas, "Multilayer perceptron (MLP)," *Geomatic Approaches for Modeling Land Change Scenarios*, pp. 451–455, 2018.

[62] Lokanath Sarangi, Mihir Narayan Mohanty, and Srikanta Pattanayak, "Design of MLP Based Model for Analysis of Patient Suffering from Influenza," *Procedia Computer Science*, vol. 92, pp. 396–403, 2016.

[63] Jia-Ming Liu, Mingyu You, Zheng Wang, Guo-Zheng Li, Xianghuai Xu, and Zhongmin Qiu, "Cough detection using deep neural networks," in *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2014, pp. 560–563.

[64] Claude Sammut and Geoffrey I Webb, "Leave-one-out cross-validation," *Encyclopedia of Machine Learning*, pp. 600–601, 2010.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[66] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[67] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*. IEEE, 2017, pp. 1–6.

[68] Xingqun Qi, Tianhui Wang, and Jiaming Liu, "Comparison of support vector machine and softmax classifiers in computer vision," in *2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. IEEE, 2017, pp. 151–155.

[69] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[70] Erik Marchi, Fabio Vesperini, Felix Weninger, Florian Eyben, Stefano Squartini, and Björn Schuller, "Non-linear prediction with LSTM recurrent neural networks for acoustic novelty detection," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[71] Justice Amoh and Kofi Odame, "Deep neural networks for identifying cough sounds," *IEEE transactions on Biomedical Circuits and Systems*, vol. 10, no. 5, pp. 1003–1011, 2016.

[72] Alex Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, pp. 132306, 2020.

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[74] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[75] Daniyal Liaqat, Salaar Liaqat, Jun Lin Chen, Tina Sedaghat, Moshe Gabel, Frank Rudzicz, and Eyal de Lara, "Coughwatch: Real-World Cough Detection using Smartwatches," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 8333–8337.