Cover Your Cough: Detection of Respiratory Events with Confidence Using a Smartwatch

Khuong An Nguyen KHUONG.NGUYEN@RHUL.AC.UK
Zhiyuan Luo ZHIYUAN.LUO@RHUL.AC.UK
Computer Science Department, Royal Holloway, University of London, Surrey, United Kingdom

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Ralf Peeters

Abstract

Cough and sneeze are the most common means to spread respiratory diseases amongst humans. Existing approaches to detect coughing and sneezing events are either intrusive or do not provide any reliability measure. This paper offers a novel proposal to reliably and non-intrusively detect such events using a smartwatch as the underlying hardware, Conformal Prediction as the underlying software. We rigorously analysed the performances of our proposal with the Harvard ESC Environmental Sound dataset, and real coughing samples taken from a smartwatch in different ambient noises.

Keywords: Cough and sneeze detection; conformal prediction

1. Introduction

Human speech recognition is a well-studied subject, with robust implementations such as Google Voice and Apple Siri, which are capable of deciphering various human voices with near absolute accuracy in major languages. Nevertheless, non-speech body sounds (e.g. cough, sneeze) processing had not received major attentions, despite containing valuable information about the person's health conditions. Previous research in classifying respiratory sounds have one thing in common, that is the lack of reliability measure for the prediction. In addition, the sampling method is either intrusive by demanding continuous monitoring of the ambient sounds around the test object, which may include private conversations, or requiring the person to wear an unusual piece of hardware around the neck or stomach.

1.1. Paper's contributions

This paper offers two novel approaches to detecting coughing and sneezing events. Firstly, we propose the use of a smartwatch, an increasingly prevalent accessory, to capture the potential respiratory related sound just before it may happen by monitoring the activity of the in-built accelerometer. Secondly, we apply Conformal Prediction to analyse the recorded samples to reliably identify any coughing or sneezing events. In principle, the benefit of our approach are:

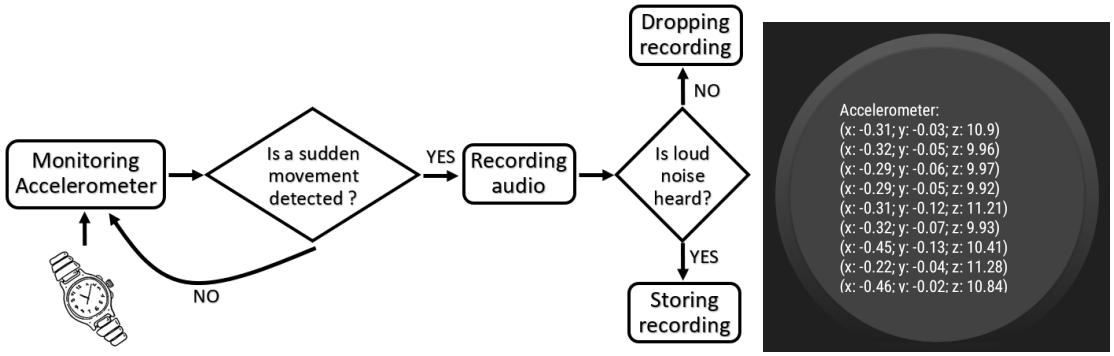
- The accelerometer is a low-power sensor, and is well-suited to monitor coughing or sneezing over prolonged time period.

- No continuous auditory recording is needed, which is well-known for being intrusive. The recording is triggered just before a potential coughing or sneezing event.
- Audio samples are captured in small segments, hence, reducing the chance of being polluted by ambient noises, and avoiding the need to isolate the coughing event from a long recording.
- The smartwatch is becoming a daily accessory for mainstream users.
- A confidence level is associated to each test sample, giving more information than just a simple yes/no detection in traditional approaches.

2. Recording potential coughing and sneezing events with a smartwatch

Previous coughs and sneezes detection techniques mostly fall into two categories. The first one is based on auditory recordings, by continuously monitoring the sound around a test object. These recordings are later screened by a trained expert or an algorithm to spot any coughing or sneezing events. The challenge is that the ambient noises may pollute the recording. Additionally, this approach is intrusive in the sense that all sound events, including conversations are recorded.

The second category is based on the sensor's signals. For instance, electrodes are placed on the abdominal muscles to detect any elevated expiratory pressure generated to make a cough or sneeze. This approach is inherently less intrusive than the first one, and is arguably more accurate, at the expense of the hassle of having to wear a sensor on the human body, and the effort to maintain such hardware.



(*a*) The decision process to trigger the auditory recording with a smartwatch.

(*b*) The Android watch app we developed to monitor the accelerometer.

Figure 1: Our proposed auditory recording process on the smartwatch.

The approach in this paper is a hybrid one, combining both sound recording and sensor signals. We use a smartwatch to record the ambient sound around the user, only if a swift hand movement is detected. Our assumption is that the user should quickly cover her mouth before coughing or sneezing, which serves as a trigger to start the auditory recording.

Although it may be more beneficial to train a model to detect this 'hand-covering-mouth' gesture, using an accelerometer dataset, we decided to simply initiate the recording after a fast hand movement for simplicity. The whole decision process is summarised in Figure 1(a). This process is executed by an Android app that we developed which runs on the watch to monitor the accelerometer and captures the ambient sounds (see Figure 1(b)). These recordings are kept locally on the watch and will later be transferred to a PC for analysis.

Nevertheless, the caveat is that the user may also move their hands quickly to perform other daily routines (e.g. picking up an item) which unwittingly triggers the recording process too. Thus, in the next section, we will explain our approach using machine learning to classify the correct respiratory events.

3. Classifying coughs and sneezes from auditory recordings with confidence machine learning

Our task can be formulated as: given a training database of labelled auditory recordings including coughing and sneezing samples, and a new un-labelled recording, we need to identify if a cough or a sneeze happens within this new recording. Because of the finite nature of the label set, this is a classification problem. Note that as we are only interested in knowing whether the test sample contains a cough, a sneeze or none, this problem may be viewed as a trinary classification, or binary classification if we group cough and sneeze into one class.

In principle, the challenges for our problem are:

- **The characteristics of the test sample may differ from that of the training ones.** While we can control the auditory training examples so that they have uniform length, the test sample may have various lengths or different recording conditions, which are beyond our control.

- **The auditory recording may carry more than just the main sound event.** Emotional tone, acoustical noise, speaking rate are just a few potential factors that may pollute or obfuscate the main content of a recording. *It is interesting to note that a person can never speak the same word in exactly the same vocal tone twice.* (Sodnik and Tomažič, 2015)

To tackle these challenges, we first extract the relevant auditory features from the recordings using Mel Frequency Coefficient, Chroma Feature Analysis, and Zero Crossing Rate. Then, we apply Dynamic Time Warping which stretches or compresses the audio sequences for matching. Finally, while previous classification-based methods simply stated yes or no, we will apply Conformal Prediction to provide a confident prediction for each test sample. The work-flow of our approach is depicted in Figure 2 and will be explained in detail in the next sections.

3.1. Extracting features from an auditory recording

The first step before training or predicting coughing samples is to extract the meaningful features from the recording. In principle, we want to identify the main acoustic components that are relevant to a coughing or sneezing event, and avoid irrelevant bits (e.g. ambient

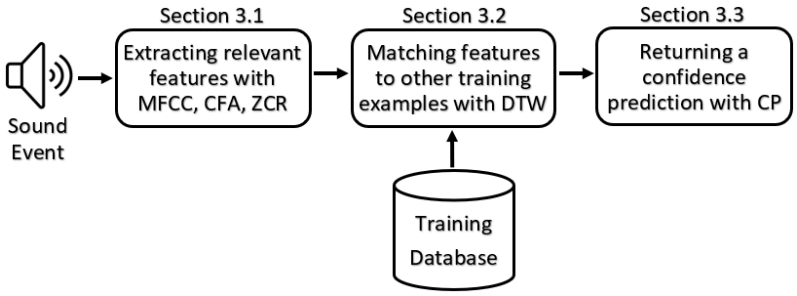


Figure 2: The workflow of our machine learning based sound event classification process.

noises). Additionally, this step will generate the same number of features for each training example, although the original recordings may have different lengths. For these purposes, we employed three different feature extracting techniques.

The first one is Mel Frequency Cepstral Coefficient (MFCC), which is a popular linguistic technique to extract the power spectrum of the recording (Mermelstein, 1976; Muda et al., 2010). In short, MFCC mimics human speech production and perception (i.e. the vocal shape of the tongue and teeth that determines how the sound is produced) and tries to eliminate other speaker dependent features embedded within the recording (e.g. the tone of speech).

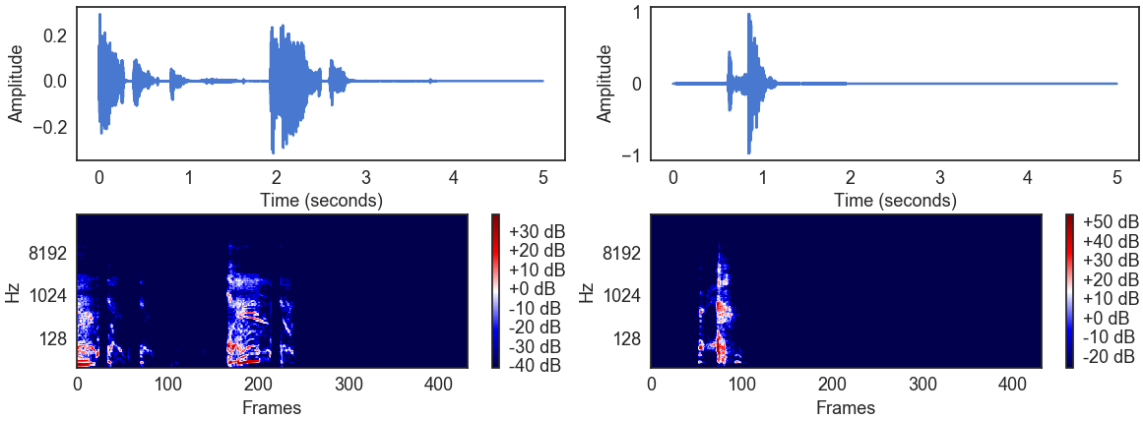
The second one is Chroma Feature Analysis (CFA), which was inspired by music notes (Müller and Ewert, 2011). In short, it projects the entire recording onto 12 bins corresponding to 12 distinct pitch classes. The reason is that music notes that are one octave apart are considered as similar, although their tones are different. CFA is useful in capturing the harmonic and melodic features of the recording, regardless of the type of instruments and timbre.

The third one is Zero Crossing Rate (ZCR), which observes if the audio signal is composed of high frequency contents (Saunders, 1996). ZCR is particularly useful to detect human conversations within a recording.

In summary, we chose the above three techniques in such a way that they complement each other in providing their own features to accurately describe an event within an auditory recording. Each technique is capable of extracting their own unique set of features from a given auditory recording, and will later be used to train our classification model. Figure 3 demonstrates an example of the features extracted from a coughing and a sneezing recording.

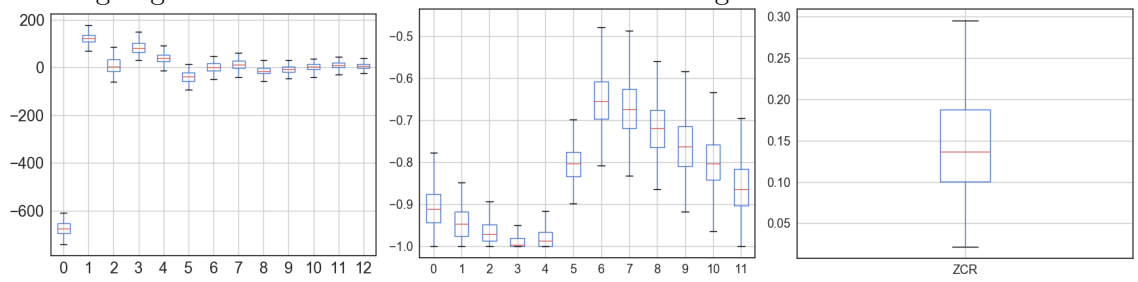
3.2. Matching features with Dynamic Time Warping

Given a set of features extracted from a new auditory sample, we need a means to compare this sample to other examples in our training dataset. While the previous step was tasked with extracting the relevant acoustic machine learning related information from a noisy recording, this step will deal with the mis-alignment of the recordings (e.g. a cough may be swift or spread out for several seconds). For this purpose, we employed Dynamic Time Warping (DTW) (Rabiner and Juang, 1993; Müller, 2007).



(*a*) The waveforms and spectrogram of a coughing event.

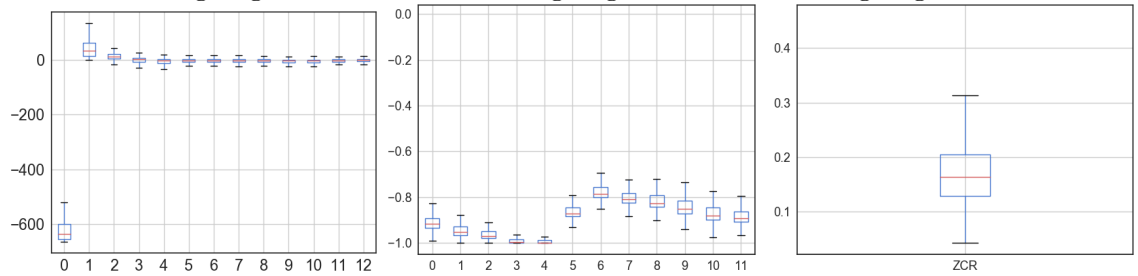
(*b*) The waveforms and spectrogram of a sneezing event.



(*c*) The distribution of the 13 MFCC coughing features.

(*d*) The distribution of the 12 CFA coughing bins.

(*e*) The distribution of ZCR coughing feature.



(*f*) The distribution of the 13 MFCC sneezing features.

(*g*) The distribution of the 12 CFA sneezing bins.

(*h*) The distribution of ZCR sneezing feature.

Figure 3: The visualisations of the coughing and sneezing events, perceived by human and machine. The waveforms and spectrograms are normally used by trained experts to spot the events, whereas the MFCC, CFA and ZCR diagrams are used by machine learning algorithms to classify the events.

The foremost benefit of DTW is that it can stretch the shorter recording to match the longer one, which is essential for our purpose because of different lengths of the auditory recordings. Secondly, DTW can match mis-aligned sequences, caused by various speak rates, by looking for the optimal warping path between them, whereas other distance-based

metrics such as Euclidean or Manhattan simply align the i^{th} point on the first sequence to another i^{th} point on the second sequence.

Without loss of generality, given two recordings $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_n)$, where m and n are the length of the sequences, a_i and b_i $(1 \leq i \leq (m, n))$ is a vector representing the features along the time series, DTW tries to find the optimal warped path of length $k : (p_1, q_1), \ldots, (p_k, q_k)$ that minimises $\sum_{i=1}^{k} |A(p_i) - B(q_i)|$.

DTW first constructs an m-by-n matrix M, where $M[i, j]$ is the distance between a_i and b_j. The final distance $M[m, n]$ represents the optimal distance between the two time series, which can be calculated recursively as follows.

$$M[i, j] = |a(i) - b(j)| + min \left\{ \begin{array}{l} \text{M[i-1, j]} \\ \text{M[i-1, j-1]} \\ \text{M[i, j-1]} \end{array} \right\} \tag{1}$$

with $i = 1 : m$ and $j = 1 : n$, and $M[1, 1] = |a(1) - b(1)|$.

There are two ways we may implement DTW for machine learning classification. The first one is by treating DTW as a distance metric for k-Nearest Neighbours to find the closest training examples to the test sample, which is similar to how the Euclidean distance, Manhattan distance are normally used. The second way is to implement DTW as an SVM kernel (Martin et al., 2016; Gudmundsson et al., 2008). A simple DTW-based kernel may be defined as $K(x, z) = DTW(x, z)$. It is worth mentioning that in our case, the training examples have equal number of features, which leads to higher chance of convergence, although the nature of DTW is positive definite and symmetric.

In the next section, we will explain how to incorporate Conformal Prediction into DTW to produce a confidence level for each prediction.

3.3. Providing the confidence level with Conformal Prediction

Intuitively, the concept of classification Conformal Prediction (CP) is testing all possible labels for a new sample, to observe how well it fits into the whole training dataset. In order to quantify the level of difference (or similarity) between each training example and the rest, a non-conformity function (which can be designed in any way we prefer) is applied to calculate a non-conformity score. In this paper, we will implement three underlying algorithms with CP, namely SVM with different kernels, k-Nearest Neighbours, and Random Forest. Based on this score, we can work out the p-value of each label as follows, which indicates the order statistics of the non-conformity score of the test object in the distribution of non-conformity scores defined by the new sample and the training examples (Vovk et al., 2005; Shafer and Vovk, 2008).

$$p(L) = \frac{\#|\{j = 1, \ldots, M + 1 : \alpha_j(L) \geq \alpha_{M+1}(L)\}|}{M + 1} \quad . \tag{2}$$

with M is the size of the training set, and L is the label.

The higher the p-value is, the better it indicates that the assumed label L helps the new sample T_{M+1} fit into the training dataset. In opposite, the lower the p-value is, which means α_{M+1} is much bigger than the majority of α_j, the stronger the indication is that the assumed label L makes this new sample an outlier.

Finally, if the user requires a single prediction, the label with the largest p-value is chosen as the predicted label. If the user prefers a prediction set, a confidence level $(1-\epsilon)$ within *0%* and *100%*, where ϵ is the significance level, is required. The labels with p-value greater than ϵ are included in the prediction set Γ^{ϵ}.

$$\Gamma^{\epsilon}(T_1, \ldots, T_{M+1}) = \{L | p(L) > \epsilon\} \quad . \tag{3}$$

The version of CP used in this paper is a variant called Inductive CP (ICP), that is intended for handling large datasets, such as the one used in this paper. ICP splits the training examples into two roughly equal sets - a 'proper training set' and a 'calibration set'. The training examples are randomly picked for each set to ascertain that all classes are well-represented in both sets.

In the first step, we may employ our preferred learning algorithm (e.g. SVM, k-Nearest Neighbours) to train a model M using the proper training set. In the second step, we calculate the non-conformity score α for each example in the calibration set. Finally, given a new sample, we go through all possible labels, and calculate its α in the same way as the previous step, as well as its p-value.

Clearly, the advantage of ICP is we need not re-calculating the non-conformity score α for the whole training set, for each possible label. The training model used to predict the label for the new sample stays the same.

4. Empirical results

Having outlined the theoretical discussions, we are now in a good position to present the results of our approach, executed on real-life datasets.

4.1. The datasets

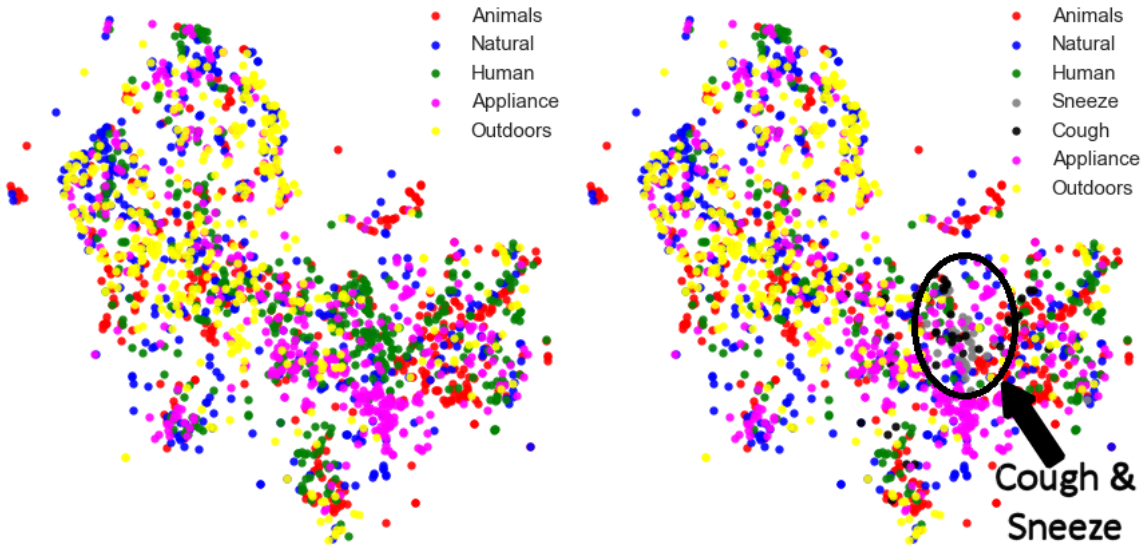
To train our classification model, we used the ESC Dataset for Environmental Sound Classification, released in 2015. It was one of the recent large-scale audio datasets, with over 2,000 training examples, covering 50 classes of real-life sound events. Each sample is rigorously examined and hand-labelled for correctness. This dataset is available publicly on Harvard Dataverse[1]. For visualisation purpose, we employed t-distributed Stochastic Neighbor Embedding, a popular tool to inspect the structure of high-dimensional data (see Figure 4).

To test our model, we independently record 40 coughing test samples on our Huawei W1 smartwatch in different ambient noise environments. We sampled another 40 sneezing recordings from Freesound.org. Note that the test recordings were deliberately sampled with different lengths to make the classification process more challenging. A summary of both datasets is presented in Table 1.

4.2. Evaluation criteria

To evaluate the overall efficiency of our approach, we will adapt four criteria, namely the precision, the recall, the accuracy, and the F_1 score. Their information is detailed in Ta-

1. https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YDEPUT - last accessed in March 2018.



(a) The 2D distribution of the 5 higher categories, showed certain overlapped areas amongst different categories which makes the classification challenging.

(b) However, the coughing and sneezing classes are fairly concentrated.

Figure 4: The visualisation of the whole ESC dataset with t-SNE.

Table 1: Summary of the datasets. The Smartwatch set contains only the coughing and sneezing samples and will mainly be used for testing the classification model trained by the ESC dataset.

	ESC set	**Smartwatch set**
Samples	2,000	80
Length of sample	5 seconds	5-10 seconds
Classes	50	2
Samples per class	40	40
Features	26	26
Recording source	Freesound.org	Smartwatch & Freesound.org
Recording frequency	44.1 kHz	44.1 kHz

ble 2. In the interest of evaluating the performance of CP, we also examine four additional information, for any given confidence level.

- Emptiness: how many test samples were returned empty (i.e. CP with the chosen underlying algorithm rejects all potential labels). This often happens when the given confidence level is low.

- Uncertainty: how many test samples contain more than one prediction. Ideally, we prefer as few predictions as possible (tight prediction region), while maintaining a high confidence level.
- Validity: after a certain number of test samples, are the overall predictions still valid ? (i.e. the number of times CP does not produce a prediction that contains the correct prediction should not exceed the specified significance level).

Table 2: The criteria used to evaluate the overall performance of each baseline algorithm. We will mostly focus on the Accuracy and the F_1 score as they represent a balance between Precision and Recall.

	Formula	**Explanation**
Precision	$\frac{\#TP}{\#TP + \#FP}$	the fraction of correctly positive predictions to the total positive predictions.
Recall	$\frac{\#TP}{\#TP + \#FN}$	the fraction of correct predictions to the actual positive predictions.
Accuracy	$\frac{\#TN + \#TP}{\#TN + \#TP + \#FN + \#FP}$	the fraction of test samples correctly predicted.
F_1 score	$\frac{2 * Recall * Precision}{Recall + Precision}$	this is a weighted average of the above precision and recall values.

Note: TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

It is worth reminding that our purpose is to provide reliable predictions in the form of confidence levels and to ensure that they remain valid, rather than to improve the usual predictions by non-CP algorithms.

4.3. Baseline performances with DTW

Firstly, we will train our classification model on the ESC dataset, and perform 5-fold cross-validation and grid-search with $k = \{1, 3, 5, 10\}$, $gamma = \{0.1, 0.2, 0.5\}$, $C = \{1, 5, 10\}$ to optimise its parameters and obtain some baseline results, before incorporating CP to get the confidence level. Figure 5 presents the mean accuracy of k-NN, Random Forest, and SVM with and without DTW under different parameters.

Our first impression was that, unfortunately, the impact of DTW was little, where 10-NN with DTW as the distance metric performed similarly to 10-NN with Euclidean metric at about 28% accuracy. The highest result belonged to SVM with DTW kernel at 42.2%, although SVM with a Linear kernel was just slightly below at 41.6%. The F1 score plot shows rather mixed results, where 10-NN excelled in certain classes and SVM

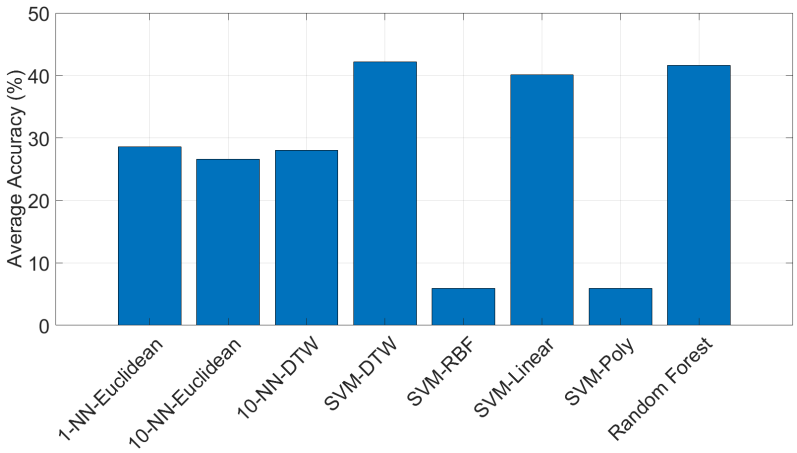


Figure 5: The mean accuracy of k-NN, SVM, and Random Forest, averaged over 5-fold cross-validation. SVM with DTW kernel achieved the best overall accuracy at 42.2%, however, SVM with a Linear kernel is just slightly behind at 41.6%.

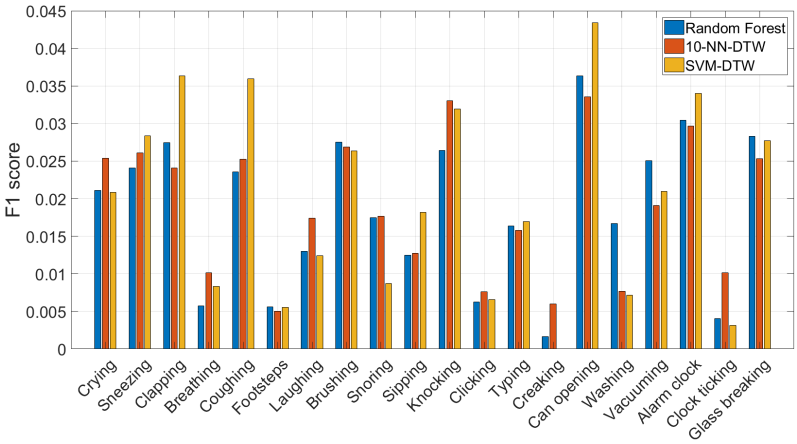


Figure 6: The F1 score plot of Random Forest, 10-NN-DTW, and SVM-DTW. The result was rather mixed with SVM-DTW excelled in some classes while Random Forest did in the others.

did in others (see Figure 6). A probable explanation for these results was that our feature extraction scheme did a great job in extracting the most relevant information from the auditory recording. Plus, all training recordings were exactly 5 second long, which nullifies the impact of DTW. Later on, we will use this training model to examine the a separate test sets with different recording lengths.

To gain a deeper understanding of the classification result on individual classes, we drew a confusion matrix of SVM-DTW for 20 classes, including all respiratory labels (see Figure 3). What is interesting from this result was that although the algorithm achieved

the highest classification accuracy within every class (i.e. the diagonal line), with up to 57.5% within the 'Sneezing' class itself for example, the 'Crying' class were classified as 2.5% (in the same Sneezing row), which indicates that some of their training examples may be similar.

4.4. Performance analysis with Conformal Prediction

This section analyses the performance of CP, in particular, ICP on the datasets.

To test ICP on the ESC dataset, we will split the dataset into five random folds. Two of them (800 samples) will be used as a proper training set to train a CP classification model. The other two (800 samples) as a calibration set to calibrate our model, and the remaining fold (400 samples) as the test set. Table 4 demonstrates the performance of ICP with k-NN, SVM, and Random Forest under different settings.

At a quick glance, it seems all algorithms performed impressively with much higher accuracy than the baseline performances earlier without CP. It is worth reminding that while it is true that the predictions returned by CP did include the correct label, it also includes other labels. In fact, when the confidence level was 95%, 1-NN with CP returned almost every single labels. Hence the 'uncertainty' level was reported as a high percentage too. Interestingly, 10-NN-DTW, SVM-DTW with CP did mange a few single predictions at the same 95% confidence level. This is also a major improvement from CP, that is the ability to return a set of predictions, rather than the usual single prediction from other algorithms. Ideally, we would like to achieve a high accuracy, but low uncertainty.

To have a deeper insight into the impact of the underlying algorithms on CP, we plot the histogram of the p-values of the correct label for all test samples. The figures display a near uniformly distributed trend for the p-values of the correct labels, whereas the opposite figure for the p-values of the wrong labels shows a half normal distribution trend with the majority of p-values concentrated around 0 (see Figure 7). This trend was expected and was important to maintain the validity of CP, as we will examine later on.

To inspect the validity of ICP, we have a closer look at the error rate across 100 confidence levels from 0% to 100%, averaged over 5 runs on 400 test samples (see Figure 8). The results indicated that ICP produced valid predictions for all confidence levels, subject to statistical fluctuations.

Now, to examine the performance of our approach on the smartwatch test set to observe how well our proposed system detects cough and sneeze events in challenging real life conditions, we will use the whole ESC dataset to train and calibrate a classification model, which will then be used to predict the test samples in the smartwatch set. Firstly, we will split the ESC dataset into two equal halves, one half as a proper training set used to train an ICP classification model, and the other half as a calibration set used to calibrate our model.

Our first impression was that since our training examples and test samples did not come from the same distribution, the independent and identically distributed (i.i.d) condition was violated, which was expressed by a much higher error rate in the results (see Table 5). Nevertheless, the advantage of DTW was noted with higher accuracy and lower error rate for 10-NN-DTW at 95% confidence, and lower uncertainly for SVM-DTW at 90% confidence.

Table 3: The confusion matrix of SVM-DTW for 20 classes including all respiratory events. Although the algorithm achieved the highest classification accuracy within every class (i.e. the diagonal line), some of the training examples were similar amongst classes.

	Cry	Sneeze	Clap	Breath	Cough	Foots	Laugh	Brush	Snore	Drink	Knock	Click	Type	Creak	Open	Wash	Vacuum	Alarm	Tick	Break
Crying baby	50.0%	2.5%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	0.0%
Sneezing	2.5%	57.5%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	0.0%	2.5%	5.0%	0.0%	0.0%	0.0%	0.0%	5.0%
Clapping	5.0%	0.0%	62.5%	0.0%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	2.5%	0.0%
Breathing	0.0%	2.5%	10.0%	7.5%	5.0%	2.5%	0.0%	2.5%	0.0%	2.5%	0.0%	0.0%	10.0%	7.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Coughing	0.0%	12.5%	0.0%	0.0%	55.0%	0.0%	2.5%	0.0%	0.0%	12.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%
Footsteps	0.0%	0.0%	2.5%	0.0%	2.5%	10.0%	2.5%	0.0%	0.0%	0.0%	5.0%	0.0%	17.5%	0.0%	0.0%	5.0%	0.0%	0.0%	2.5%	0.0%
Laughing	0.0%	2.5%	0.0%	0.0%	10.0%	5.0%	22.5%	0.0%	2.5%	5.0%	0.0%	0.0%	5.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Brushing teeth	2.5%	0.0%	0.0%	0.0%	0.0%	5.0%	0.0%	57.5%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%
Snoring	0.0%	0.0%	0.0%	2.5%	0.0%	2.5%	0.0%	0.0%	40.0%	2.5%	10.0%	2.5%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%
Drinking	0.0%	2.5%	0.0%	0.0%	15.0%	2.5%	2.5%	2.5%	7.5%	25.0%	7.5%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Door knock	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	0.0%	2.5%	5.0%	65.0%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Mouse click	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	5.0%	2.5%	7.5%	0.0%	12.5%	7.5%	0.0%	0.0%	12.5%	2.5%	0.0%	2.5%	0.0%	5.0%
Keyboard typing	0.0%	0.0%	2.5%	0.0%	2.5%	5.0%	0.0%	0.0%	5.0%	5.0%	7.5%	0.0%	37.5%	0.0%	0.0%	0.0%	0.0%	0.0%	5.0%	2.5%
Door creaks	7.5%	2.5%	7.5%	5.0%	0.0%	5.0%	2.5%	0.0%	2.5%	0.0%	0.0%	0.0%	5.0%	2.5%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%
Can opening	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	2.5%	0.0%	82.5%	0.0%	0.0%	0.0%	0.0%	7.5%
Washing machine	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	30.0%	2.5%	0.0%	0.0%	0.0%
Vacuum cleaner	0.0%	0.0%	7.5%	0.0%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	45.0%	0.0%	0.0%	0.0%
Clock alarm	7.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	0.0%	0.0%	0.0%	0.0%	57.5%	0.0%	2.5%
Clock tick	5.0%	10.0%	0.0%	0.0%	5.0%	5.0%	0.0%	2.5%	7.5%	0.0%	5.0%	0.0%	7.5%	2.5%	0.0%	0.0%	0.0%	0.0%	5.0%	2.5%
Glass breaking	0.0%	7.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	0.0%	0.0%	22.5%	0.0%	0.0%	0.0%	2.5%	57.5%

Table 4: The performance of ICP on the ESC dataset, averaged over 5 runs, with 400 test samples. Higher accuracy, lower uncertainty, lower error rate are generally desirable. The best results for each criterion are highlighted in bold.

(a) 95% confidence level, $\epsilon = 0.05$

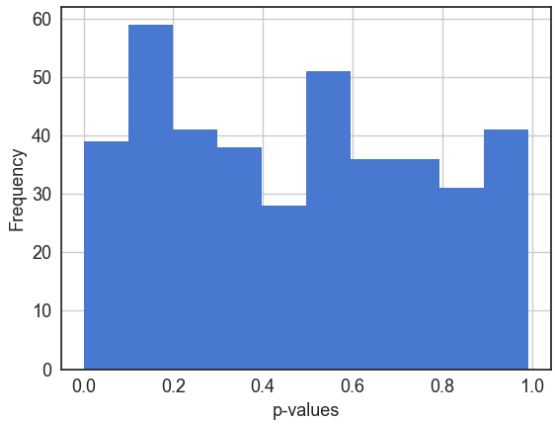
	Accuracy	**Uncertainty**	**Emptiness**	**Error rate**
1-NN-Euclidean	95.25%	100%	0%	4.75%
10-NN-Euclidean	94%	99.5%	0%	6%
10-NN-DTW	93.75%	98.9%	0%	6.25%
SVM-DTW	95.75%	**94.3%**	0%	4.25%
SVM-RBF	94.25%	100%	0%	5.75%
SVM-Linear	95.5%	94.8%	0%	4.5%
SVM-Poly	95.5%	99.2%	0%	4.5%
Random Forest	**97.25%**	98.2%	0%	**2.75%**

(b) 90% confidence level, $\epsilon = 0.1$

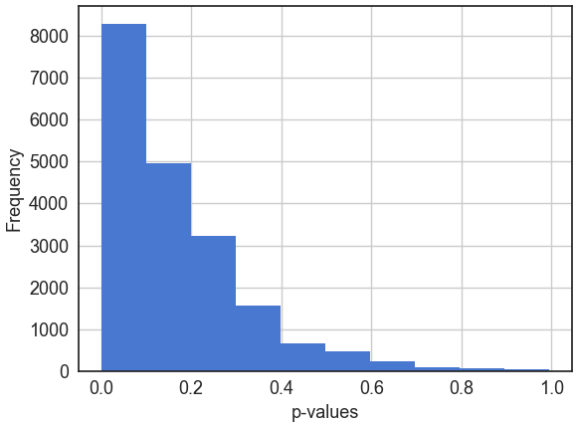
	Accuracy	**Uncertainty**	**Emptiness**	**Error rate**
1-NN-Euclidean	89.25%	100.0%	0%	10.75%
10-NN-Euclidean	87.75%	98.3%	0%	12.25%
10-NN-DTW	88.25%	99.4%	0%	11.75%
SVM-DTW	89%	**91.6%**	0%	11.0%
SVM-RBF	**90.5%**	100%	0%	**9.5%**
SVM-Linear	90.25%	92.8%	0%	9.75%
SVM-Poly	89.75%	97.5%	0%	10.25%
Random Forest	87.5%	94.9%	0%	12.5%

(c) 20% confidence level, $\epsilon = 0.8$. At this low confidence, the majority of prediction sets were empty.

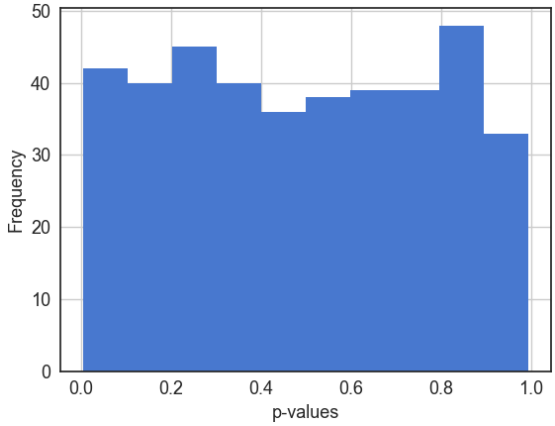
	Accuracy	**Uncertainty**	**Emptiness**	**Error rate**
1-NN-Euclidean	17.25%	0%	15%	82.75%
10-NN-Euclidean	14.25%	0%	35%	85.75%
10-NN-DTW	17%	0%	33.5%	83%
SVM-DTW	18.75%	0%	60.75%	81.25%
SVM-RBF	**19.25%**	100%	**0%**	**80.75%**
SVM-Linear	18.5%	0%	59.75%	81.5%
SVM-Poly	17.5%	0%	51.25%	82.5%
Random Forest	14.25%	0%	50.5%	85.75%



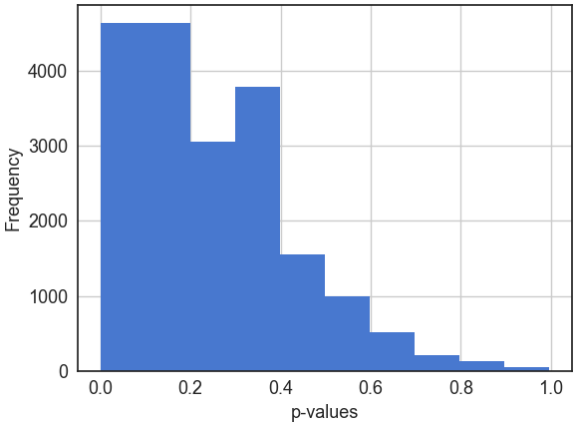
(*a*) The distribution of the p-values for correct labels with SVM-DTW.



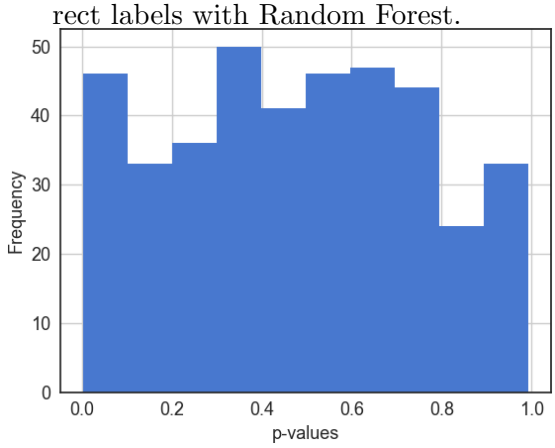
(*b*) The distribution of the p-values for wrong labels with SVM-DTW.



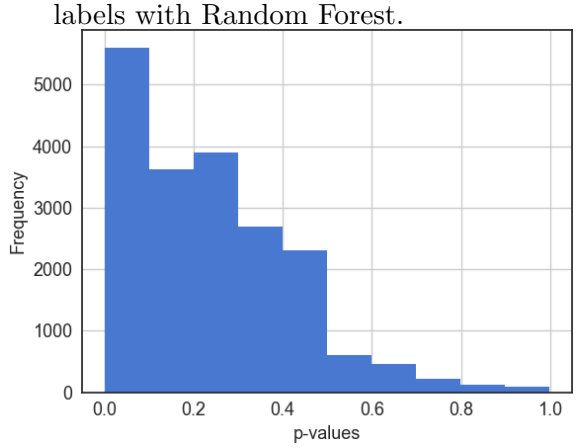
(*c*) The distribution of the p-values for correct labels with Random Forest.



(*d*) The distribution of the p-values for wrong labels with Random Forest.

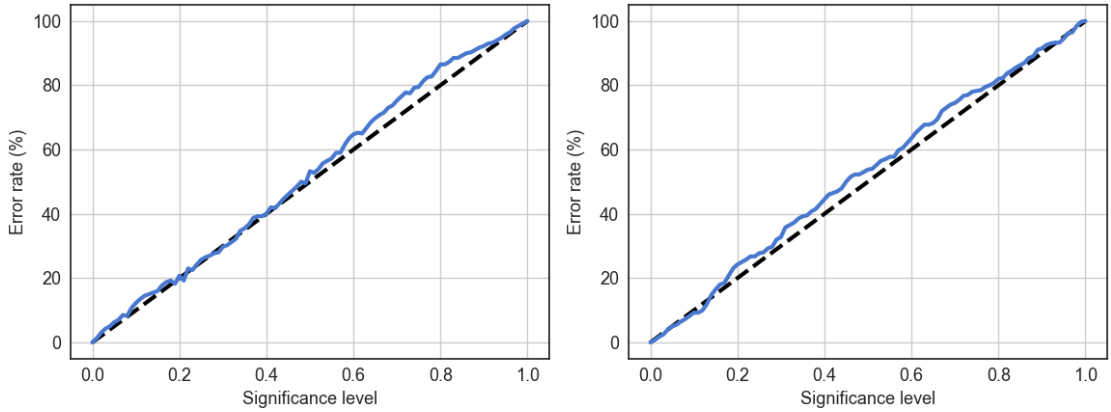


(*e*) The distribution of the p-values for correct labels with 10-NN-DTW.



(*f*) The distribution of the p-values for wrong labels with 10-NN-DTW.

Figure 7: The histogram of the p-values using different underlying algorithms. The left figures display a near uniformly distributed trend, whereas the right figures show a half normal distribution trend with the majority of p-values concentrated around 0. This trend demonstrates the ability to maintain the validity of CP to be examined later on.



(a) The error rate for 10-NN-DTW. (b) The error rate for SVM-DTW.

Figure 8: The validity of ICP averaged over 5 runs on 400 test samples of the ESC dataset. All 100 confidence levels were evaluated, which indicated that ICP produced valid predictions, subject to statistical fluctuations.

Table 5: The performance of ICP on the Smartwatch dataset with 80 test samples. Higher accuracy, lower uncertainty, lower error rate are generally desirable. The best results for each criterion were highlighted in bold. The result has higher error rate than the ESC dataset, mostly because of the low number of test samples and because the training and test samples did not come from the same distribution.

(a) 95% confidence level, $\epsilon = 0.05$

	Accuracy	**Uncertainty**	**Emptiness**	**Error rate**
1-NN-Euclidean	92.5%	100%	0%	7.5%
10-NN-Euclidean	96.25%	98.7%	0%	3.75%
10-NN-DTW	**98.75%**	96.2%	0%	**1.25%**
SVM-DTW	91.25%	**93.5%**	0%	8.75%
SVM-RBF	80%	100%	0%	20%
SVM-Linear	91.25%	100%	0%	10.0%
SVM-Poly	96.25%	99.2%	0%	3.75%
Random Forest	95%	98.7%	0%	5%

(b) 90% confidence level, $\epsilon = 0.1$

	Accuracy	**Uncertainty**	**Emptiness**	**Error rate**
1-NN-Euclidean	93.75%	100.0%	0%	6.25%
10-NN-Euclidean	88.75%	95.8%	0%	11.25%
10-NN-DTW	91.25%	98.6%	0%	8.75%
SVM-DTW	90%	**84.7%**	0%	10%
SVM-RBF	**96.25%**	100%	0%	**3.75%**
SVM-Linear	90%	87.5%	0%	10%
SVM-Poly	93.75%	89.3%	0%	6.25%
Random Forest	87.5%	94.9%	0%	12.5%

5. Related work

This section overviews some of the related work in the respiratory diseases detection domain.

One of the first approaches to combine sensor signals with ambient sounds were proposed by (Rahman et al., 2014; Sun et al., 2011; Larson et al., 2011; Matos et al., 2007). They all share one aspect in common, that is the use of bespoke microphone, attached to the subject for monitoring. Our approach improves on this aspect in the sense that we utilise the accelerometer in a ubiquitous device (i.e. the smartwatch). Our system intelligently ignores non-respiratory sounds, hence, making it less intrusive than other sound-based approaches.

Machine learning wise, the majority of previous approaches tried to predict the coughing and sneezing event also by training a model (Matos et al., 2006; Sun et al., 2015; Schröder et al., 2016; Amoh and Odame, 2016). Our approach improves on their ideas by using Conformal Prediction to associate each prediction with a confidence level to indicate how likely that a test sample is a real coughing or sneezing event.

6. Conclusion and further work

We have presented a novel approach to detecting coughing and sneezing events using a smartwatch and confidence based algorithms. Our proposal identified the challenges of different characteristics (i.e. recording length, speaking rate) of the training and testing samples, as well as the multiple events encapsulated within an auditory recording (i.e. acoustical noise, emotional tone). We first extracted the relevant features with MFCC, CFA, and ZCR, then used them to train a machine learning model. To match the test sample with the training examples, we employed Dynamic Time Warping which excelled at handling mis-aligned recordings. Lastly, we implemented Conformal Prediction to associate a confidence level for each test sample. We have evaluated our proposal on a large ESC Sound Environment dataset to present its efficiency and validity, as well as testing it on a real-life dataset sampled directly from a smartwatch to demonstrate its capability in classifying coughing and sneezing samples.

Nevertheless, what we have presented in the paper is not the end-story. The detection process may be improved further in two ways. Firstly, we may train a model to smartly initiate the auditory recording, by predicting if a hand gesture is truly to cover the mouth, instead of triggering the recording process every time a swift movement is detected in the current design. Secondly, we may incorporate more sensors already built in the smartwatch to aid the detection process. For example, some research suggested that the human heart could miss a beat when sneezing, which may be detected by a heart-rate sensor.

Acknowledgments

This work is partly funded by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 671555.

References

Justice Amoh and Kofi Odame. Deep neural networks for identifying cough sounds. *IEEE transactions on biomedical circuits and systems*, 10(5):1003–1011, 2016.

Steinn Gudmundsson, Thomas Philip Runarsson, and Sven Sigurdsson. Support vector machines and dynamic time warping for time series. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2772–2776. IEEE, 2008.

Eric C Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N Patel. Accurate and privacy preserving cough sensing using a low-cost microphone. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 375–384. ACM, 2011.

Stephanie Martin, Peter Brunner, Iñaki Iturrate, José del R Millán, Gerwin Schalk, Robert T Knight, and Brian N Pasley. Word pair classification during imagined speech using direct brain recordings. *Scientific Reports*, 6:25803, 2016.

Sergio Matos, Surinder S Birring, Ian D Pavord, and H Evans. Detection of cough signals in continuous audio recordings using hidden markov models. *IEEE Transactions on Biomedical Engineering*, 53(6):1078–1083, 2006.

Sergio Matos, Surinder S Birring, Ian D Pavord, and David H Evans. An automated system for 24-h monitoring of cough frequency: the leicester cough monitor. *IEEE Transactions on Biomedical Engineering*, 54(8):1472–1479, 2007.

Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

Meinard Müller and Sebastian Ewert. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011. hal-00727791, version 2-22 Oct 2012*. Citeseer, 2011.

Lawrence R Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.

Tauhidur Rahman, Alexander Travis Adams, Mi Zhang, Erin Cherry, Bobby Zhou, Huaishu Peng, and Tanzeem Choudhury. Bodybeat: a mobile system for sensing non-speech body sounds. In *MobiSys*, volume 14, pages 2–13, 2014.

John Saunders. Real-time discrimination of broadcast speech/music. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 993–996. IEEE, 1996.

Jens Schröder, Jorn Anemiiller, and Stefan Goetze. Classification of human cough signals using spectro-temporal gabor filterbank features. In *Acoustics, Speech and Signal*

Processing (ICASSP), 2016 IEEE International Conference on, pages 6455–6459. IEEE, 2016.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421, 2008.

Jaka Sodnik and Sašo Tomažič. *Spatial Auditory Human-Computer Interfaces*. Springer, 2015.

Xiao Sun, Zongqing Lu, Wenjie Hu, and Guohong Cao. Symdetector: detecting sound-related respiratory symptoms using smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 97–108. ACM, 2015.

Zheng Sun, Aveek Purohit, Kathleen Yang, Neha Pattan, Dan Siewiorek, Asim Smailagic, Ian Lane, and Pei Zhang. Coughloc: Location-aware indoor acoustic sensing for non-intrusive cough detection. In *International Workshop on Emerging Mobile Sensing Technologies, Systems, and Applications*, 2011.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.