POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master of Science in Biomedical Engineering

Department Of Electronics, Information and Bioengineering

Automatic Identification of Cough events in Audio Signals

Supervisor: Emilia AMBROSINI

Co-supervisor: Elena BARDI

Candidate:

Alessandro ZANINI

Identification number: 920434

Academic Year 2019-2020

RINGRAZIAMENTI

Ringrazio vivamente la Professoressa Ambrosini, che mi ha seguito in tutto lo svolgimento di questa tesi aiutandomi e risolvendo le varie complicazioni incontrate. Un altro grandissimo grazie è sicuramente per Elena Bardi che mi ha aiutato moltissimo nella scrittura della tesi e nel rispondere a tutti i miei dubbi. Infine, grazie alla mia famiglia, ai miei amici e a tutte le persone che mi sono state vicine e che mi hanno supportato durante il mio percorso di studi e lo svolgimento di questa tesi.

Contents

List of acronyms

AD Alzheimer's disease

AKV absolute kinematic velocity

AIM Advanced Informatics in Medicine

ANN Artficial Neural Network

AUC Area Under Curve

CNN Convolutional Neural Network

COPD Chronic obstructive pulmonary disease

DLN Deep Learning Networks

DNN Deep Neural Network

FN False negatives

FP False Positives

FPR False Positive Rate

FFT Fast Fourier Transform

GFCC Gammatone Frequency Cepstral Coefficient

GERD Gastroesophageal disease

HACC Hull Automated Cough Counter

HMM Hidden Markov Model

ICT Information and communications technology

KNN K Nearest Neighbour

LCM Leicester Cough Monitor

L-MFCC Liftered Mel Frequency Cepstral Coefficient

LPCC Linear predictive cepstral coefficients

LSTM Long Short Term Memory Neural Network

MCI Mild Cognitive Impairment

MFCC Mel Frequency Cepstral Coefficient

MRMR Minimum redundancy maximum relevance

NLU Natural language processing

PCA Principal Component Analysis

PD Parkinson Disease

PDF Probability Density Function

PNN Probabilistic neural network

PSD Power spectral density

RBF Radial basis function

RF Random Forest

RLSFN Random Least Squares Feed-Forward Network

RNN Recurrent Neural Network

ROC Receveing Operator Curve

ROI Regions of Interest

STFT Short Time Fourier Transform

SVM Support Vector Machine

TN True Negatives

TP True Positives

TPR True Positive Rate

UCLA University of California Los Angeles

UHC Universal health coverage

UTC Urge-to-cough

VAS Visual Analog Scoring

VGG Visual Geometry Group

WHO World Health Organization

Abstract

Introduction

The world population life expectancy is increasing every year, especially in Western countries, together with the demand of assistance and monitoring from the healthcare system towards elderly people. Moreover, this year, the Covid-19 pandemic highlighted the limited availability of accommodation and specialized personnel in healthcare structures. Furthermore, the need to maintain patients far from the hospitals, when possible, and to give care in remote to avoid the contagious spread, strongly demonstrated the importance of telemedicine and dislocation of care solutions. Mobile-health technologies, which exploits smartphones to monitor health conditions, can improve and facilitate the job of doctors and nurses. This is possible principally because m-health technologies are cheap, easy to use and allow continuous control of the users situation. The implementation of automatic algorithms on smartphones makes possible to detect from voice analysis conditions such as aging, cognitive impairments, stress, emotions like sadness and depression, which together allow to generally assess people quality of life. In particular, the Covid-19 pandemic focused the world attention to respiratory tract in general and, more specifically, to the virus symptoms: cough, fever, tiredness and loss of taste and smell. In this context, this work was focused on the assessment of the status of the respiratory tract through the count of cough events. Specifically, the aim was to develop a classification algorithm to identify cough events based on acous-

tic voice features, automatically extracted from audio recordings. The performance of different classifiers, both classical machine learning methods (Random Forest with Bagging, Support Vector Machine, K-Nearest Neighbours) and deep machine learning methods (Neural Network with Long Short-Term Memory), were compared. The capability of the classifier to correctly identify cough events from speech signals and environmental noises was analyzed on 313 cough sounds, manually labelled, 313 words and 156 ambient sounds. This project should be considered as a preliminary work towards the development of an application for smartphones, able to count the number of cough events per day in a transparent and unobtrusive manner.

State of the art

In order to perform speech analysis, many studies applied features extraction and used automatic classifiers. Speech analysis can help detect cognitive pathologies and conditions that negatively affect the general quality of life of an individual. Among all the possible applications of speech analysis some of the most related with health and aging are assessing stress and some other conditions like dementia, Parkinson and loneliness that strongly affect elderly people.

In order to assess the health of the respiratory tract, cough event detection can be useful. Cough is a forceful explosive expiration of air from the airways. Usually, cough could be a symptom of cold, flu, pneumonia, asthma or chronic bronchitis. The average duration of a cough event is between 300 ms and 650 ms. When the airflow is obstructed, the event can reach a 20 kHz frequency, but usually high-energy peaks are around 400 Hz and 1500 Hz. Cough can be detected mechanically with devices applied on the chest wall, that register its movements, or with audio based systems can be easily implemented in smartphones, making it possible to monitor the users every time and everywhere with very low costs. Going deeper into the cough detection framework, most of the times, predictors are extracted from an audio, that was previously divided in frames, and are then used to train a classifier. Many different predictors have been used in this field. For example, Pramono et al. [1] extracted temporal features, such as the zero crossing rate, energy features, such as the energy level, and also spectral

features, like spectral Kurtosis, spectral slope, spectral centroid and so on. Other studies, like the one conducted by Miranda et al [2], used spectral predictors, comparing the simple Short Time Fourier Transform Coefficient with Mel spectrogram. The Mel spectrogram was used also by Barata et al. [3] together with the Mel frequency cepstral coefficients, derived by the Mel spectrogram using a discrete cosine transform. MFCCs were used also by many other research groups, Di Perna [4], Swarnkar [5] and Tracey [6] to cite some. Pramono et al. [7], in another study, explored the performance of the Linear predictive coding coefficient, which is a time domain feature, and tonality index. Some more structured predictors have been used by Monge et al [8], who used Local Hu moments, which are advanced averages of the image or audio intensities, invariant to translation, scale, and rotation.

After the features extraction, the predictors are used to train a classifier, that categorizes the data into classes and can distinguish a cough event from other sounds. To perform this task many algorithms can be used. Logistic regression classifier was used by Pramono et al [1] to perform cough detection, achieving a sensitivity of 92%. Support vector machine, another classic machine learning method, was used by Di Perna [4] to detect chronic obstructive pulmonary patients. They were able to detect cough with value of Area Under Receiver Operating Curve (ROC) of 0.916. Also Liu et al. [9] and Tracey et al. [6] used Support Vector Machines (SVMs) to classify cough events achieving respectively 91% and 81% sensitivity. Lucio et al. [10] used K nearest neighbour (KNN) obtaining 87% sensitivity in detecting cough events. Monge-Alvarez [8] also used KNN but inside a decision tree method. Other works that used trees are Larson, Lee et al. [11] who used a random forest classifier, achieving an average sensitivity of 92%. The same method was used by Breiman et al. [12] who achieved a sensitivity equal to 92%. Amrulloh et al. [13] used Artficial Neural Networks (ANNs) classification to develop a cough detector achieving sensitivity of 93%. More advanced ANN can be used, such as probabilistic neural network (PNN) [14] or Convolutional Neural Network (CNN). The latter was used by Tracey et al. [6] in cooperation with SVM, allowing to obtain 81% sensitivity. Miranda et al. [2] used and compared CNN, Deep Neural Network and Long Short Term Memory Neural Network showing that performance varies with different frame and segment lengths configurations. Finally,

there is the Hidden Markov Model (HMM) used by Matos et al. [15] that is a key spotting technique, in which the idea is to detect occurrences of keywords in a continuous speech recording, this approach can be fit for cough as well.
An important issue in every audio recording application, and so for cough detection as well, is to protect the privacy of the users. The main criticalities are found in the possible recognition of the speaker from voice characteristics (pitch, formants), in the reconstruction of a private conversation or of personal life activities during the day and in the recognition of a person's gender, accent, nationality and emotional state [16].

Materials And Methods

Audio samples were collected from online audio datasets; specifically, from Freesound[1] and Commands Speech[2]. A total of 313 cough event samples were extracted from Freesound dataset and labelled manually with the Matlab AudioLabeler App. Then, 156 random sounds, such as music or environmental sounds, were taken from Freesound together with 313 random words from Speech Commands database. Two different datasets were created. One, called "Speech & Cough", was built with 313 cough events and 313 words. The second one, called "Complete" was built to contain 313 cough sounds, 157 random words and 156 ambient sounds from Freesound. All the algorithms and functions were implemented in Matlab R2020b (MathWorks, Inc. USA) with the Audio Toolbox[3] and the Deep Learning Toolbox[4]. After the datasets creation, the audio were downsampled to 16kHz in order to have the same frequency for all audio samples derived from different datasets. Since the amplitude of the audio signals is affected by inter-subject variability and by the different distances of the subjects from the recorder, an amplitude standardization was applied in order to make the audios comparable. Then, a feature extractor object was created and used to extract 56 predictors from the audio samples. First, the feature extractor subdivides the audio in frames of 64 ms duration with 50% overlap. The features extracted were Mel spectrogram coefficients, Mel Frequency Cepstral Coefficent, pitch, spectral Kurtosis, spectral

[1]The project and the dataset are available at: https://freesound.org
[2]The dataset is available from https://storage.googleapis.com/download.tensorflow.org data/speechcommandsv0.01.tar.gz. Copyright Google 2017.
[3]Toolbox documentation avaible at: https://ch.mathworks.com/help/deeplearning/index.html
[4]Toolbox documentation avaible at: https://ch.mathworks.com/products/audio.html

centroid, spectral crest, spectral entropy, spectral flatness, spectral flux, spectral roll off, spectral slope and harmonic ratio. Then, data was evaluated with a Lilliefors test to assess normality. If the features were normal, a t-test for independent samples was used. If the features were not normal ($p<0.05$), a Wilcoxon rank sum test. Both tests were applied to compare each feature between the two classes ("cough" assigned to "1" and "no cough" assigned to "0"). After this preliminary statistical analysis, a minimum redundancy maximum relevance (MRMR) algorithm was applied to rank the predictors according to a trade off between predictors importance and mutual information. Based on this algorithm, the 15 most relevant features were selected.
The following classifiers were compared in terms of their performance to automatically identify cough events: Random Forest with Bagging, K Nearest Neighbour, Support Vector Machine and Long Short Term Memory Neural Network. The classifiers were trained in four modalities. The first two modality consisted in using the "Speech & Cough" dataset using, on one side, all the 56 predictors and, on the other side, just the 15 features selected by MRMR. The other two modalities consisted in using the "Complete" dataset again with all the predictors in one case, and with the selected ones in the other case. The training and testing performance metrics assessment was done using a 10-fold cross validation. This allowed to average the results, thus obtaining more robust performance indicators. These performance metrics used are accuracy, precision and recall. The classifiers capacity of prediction with fresh data was made with the test set and evaluated in terms of accuracy, precision and recall. Moreover, the results were visualized into ROC curves. The area under the curve (AUC) is a performance index. Finally, a Kruskal Wallis test was used to assess whether an algorithm actually performed significantly better than another. The statistical significance between training the classifiers with all the features or with just a subset was explored, as well as the significance between using the two different datasets to train the algorithms. Finally, the significance across the classifiers was evaluated, in order to assess whether a classifier was performing better than another.

Results and Discussions

The statistical analysis carried out on the predictors showed that they did not follow

a normal distributions. The Wilcoxon test highlighted significant difference for all the predictors between the two classes "cough" and "no cough", except for three predictors of the "Complete" dataset. The MRMR algorithm selected the 15 most relevant features. In particular, for the "Complete" dataset, the three features with the highest predictor importance score were found to be "Mel4", "MFCC9" and "spectralKurtosis", while in the "Speech & Cough" dataset, the best features we found to be "Mel32", "MFCC12" and "MFCC8".

All the classifiers performance metrics were computed as the median, with the corresponding interquartile range, of the single values obtained during the 10-fold cross validation. An example is shown in table 1. The learner that generally showed the best performance was LSTM. It performed classification with a sensitivity of 92.4 ± 3.3% when tested with the "Complete" dataset, considering all the 56 features. When the "Speech & Cough" dataset was used, LSTM performed even better, achieving a sensitivity of 98.2 ± 8.1%. And also Pramono et al [1], Larson et al. [11] and Breiman et al. [12] that all obtained a sensitivity equal to 92%. The two datasets results were compared to see whether, using a cleaner dataset such as "Speech & Cough", without noises, could improve the classifier performances with respect to the "Complete" dataset. The statistical analysis showed that for KNN, when 56 features were used, the behaviour was opposite than expected, with a larger accuracy for the "Complete" dataset, 86.1 ± 3% against 79.8 ± 3.3% of the "Speech & Cough" dataset, the p-value was lower than 0.001. In the SVM case, when using the 15 selected features, the choice of the dataset showed again that the "Complete" dataset gave more information to the classifier resulting in a better performance. In general, for the rest of the cases, the presence of noises did not affect the results demonstrating that the classifiers are robust to noise and sounds different from speech. Finally, we compared the performance of each algorithm considering all features or just a subset and this analysis showed that the features selection performed with MRMR generally did not influence the results, meaning that the chosen predictors correctly stored most of the information. On the other side, feature selection did not improve the classifiers' performances. The only exception occurred for KNN, for the "Complete" dataset, which was negatively effected by feature selection. The precision was decreased from 82.6 ±6.2 % to 66.3 ± 2.3%

when a subset of features were considered, the p-value for the precision was lower than 0.001. All the algorithms' performances and significance, when comparing all the algorithms between each other, for the "Complete" dataset using all the 56 feature, are shown in table 1. Finally, the ROC curves of the 4 algorithms for the same training

Table 1: *Comparison between the different classification algorithms in terms of accuracy, precision and recall. Classifiers were trained on the "Complete" dataset, considering all 56 features*

Metrics	Classifiers				P values					
	RF	**KNN**	**SVM**	**LSTM**	**RF-KNN**	**RF-SVM**	**RF-LSTM**	**KNN-SVM**	**KNN-LSTM**	**SVM-LSTM**
Accuracy	88.2 ±1.7%	86.1 ±3%	87.6 ±1.3%	94.4 ±2.8%	0.151	0.915	0.068	0.465	**0.001***	**0.010***
Precision	76.3 ±9%	82.6 ±6.2	78.5 ±14.3%	96.8 ±2.4%	0.752	0.992	**0.001***	0.894	**0.014***	**0.001***
Recall	82.5 ±8.5%	75.5 ±10.5%	80.5 ±14.3%	92.4 ±3.3%	0.152	0.984	**0.035***	0.302	**0.001***	**0.012***

modality as in Table 1 ("Complete" dataset, 56 features) are shown in Figure 1. Using AUC it is possible to confirm that LSTM achieved the best performance, followed by RF, SVM and KNN, which obtained the worst performances.

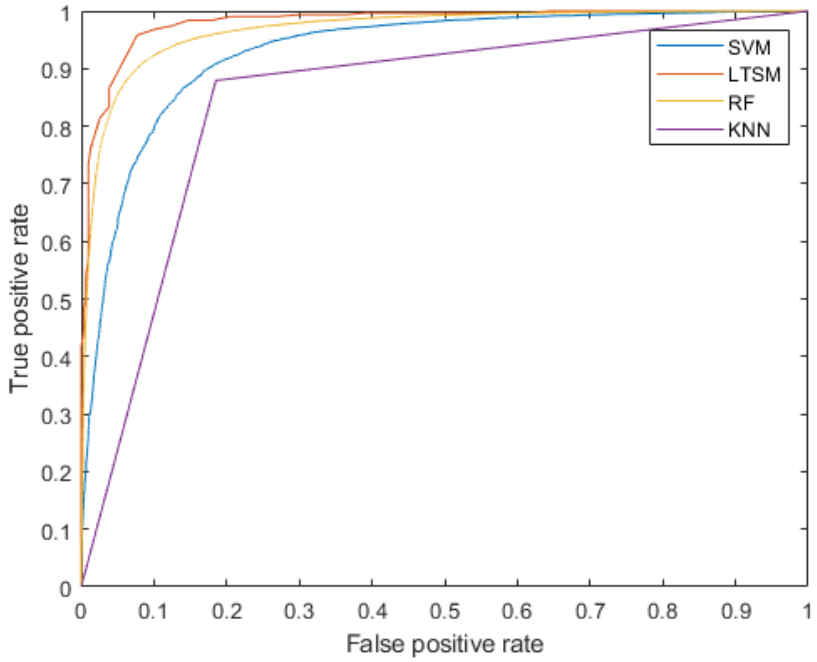

Figure 1: *ROC Curve: "Complete" Dataset, 56 Features*

Conclusions

In conclusion, the LSTM was the best classifier. The LSTM performance metrics outperformed other studies results, such as Liu et al. [9] and Tracey et al. [6] Lucio

et al. [10] which achieved, with different methods, respectively 91%, 81% and 87% sensitivity in detecting cough events. All algorithms were robust also to the presence of environmental noise, indeed the classification performance did not decrease when noise was added in the dataset. Features selection resulted to be not a mandatory step, especially for the LSTM neural network. Anyway, because it did not decrease the performance, it could be applied to reduce the dimensionality of the problem. This could indeed help speeding up calculations, in view of a future development of a smartphone application based on the best classifier, LSTM. Specifically, the trained LSTM classifier could be used for a real time classification of cough events. The smartphone application would automatically extract voice features in real time during phone conversations or during the day and count the number of cough events per day in a continuous and not intrusive way, for the purpose of assessing the respiratory tract health.

Sommario

Introduzione

L'aspettativa di vita della popolazione mondiale aumenta ogni anno, soprattutto nei paesi occidentali, assieme al bisogno di assistenza e monitoraggio nei confronti degli anziani da parte del sistema sanitario. Inoltre, quest'anno, la pandemia da Covid-19 ha evidenziato la limitata disponibilità di letti e personale specializzato nelle strutture sanitarie. Inoltre, la necessità di mantenere i pazienti lontani dagli ospedali, quando possibile, e di prestare assistenza a distanza per evitare la diffusione del contagio, ha fortemente dimostrato l'importanza della telemedicina e delle soluzioni per dislocare l'assistenza ai pazienti. Le mobile-technologies, che sfruttano gli smartphone per monitorare le condizioni di salute, possono migliorare e facilitare il lavoro di medici e infermieri. Questo è possibile principalmente perché le tecnologie m-health sono economiche, facili da usare e permettono un controllo continuo della condizione degli utenti. L'implementazione di algoritmi automatici sugli smartphone ha permesso di rilevare dal linguaggio parlato molte informazioni. Per esempio, ha aiutato a rilevare condizioni quali invecchiamento, disturbi cognitivi, stress, emozioni come tristezza e depressione. Nell'insieme queste condizioni consentono di valutare la qualità di vita generale degli utenti. In particolare, la pandemia di Covid-19 ha focalizzato l'attenzione mondiale sulle vie respiratorie in generale e, più nello specifico, sui sintomi del virus: tosse, febbre, stanchezza e la perdita del gusto e dell'olfatto. In questo contesto, lo

scopo di questo lavoro è quello di sviluppare un modello di classificazione per valutare la salute respiratoria e la qualità della vita degli utenti attraverso il conteggio degli eventi di tosse. L'obiettivo principale di questo lavoro era quello di valutare e confrontare diversi classificatori atti a rilevare eventi di tosse all'interno di audio contenenti parole e suoni ambientali. Gli algoritmi confrontati in questo lavoro erano sia metodi di machine learning classici (Random Forest con Bagging, Support Vector Machine, K-Nearest Neighbors) che metodi di deep machine learning (Neural Network con Long Short-Term Memory). La capacità del classificatore di identificare correttamente gli eventi di tosse da segnali vocali e rumori ambientali è stata analizzata su 313 suoni di tosse, etichettati manualmente, 313 parole e 156 suoni ambientali. Questo progetto di tesi deve essere considerato come un lavoro preliminare allo sviluppo di un'applicazione per smartphone, in grado di contare il numero di eventi di tosse al giorno in modo trasparente e non invasivo.

Stato dell'arte

Nello stato dell'arte dell'analisi del parlato, molti studi hanno applicato l'estrazione di feature e hanno utilizzato classifier automatici. L'analisi del parlato può aiutare a rilevare patologie cognitive e condizioni che influenzano negativamente la qualità generale della vita di un individuo. Tra tutte le possibili applicazioni dell'analisi del linguaggio alcune delle più legate alla salute e all'invecchiamento sono la valutazione dello stress e alcune altre condizioni come la demenza, il Parkinson e la solitudine, che influenzano fortemente anziani.

Per valutare la salute delle vie respiratorie, può essere utile il rilevamento degli eventi di tosse. La tosse è una forte espirazione esplosiva di aria emessa dalle vie aeree. Può essere un sintomo di raffreddore, influenza, polmonite, asma o bronchite cronica, per esempio. La tosse può essere rilevata meccanicamente con dispositivi applicati sulla parete toracica, che ne registrano i movimenti oppure usando dei sistemi audio based. I sistemi audio based possono essere facilmente implementati negli smartphone, rendendo possibile il monitoraggio dei pazienti in ogni momento e luogo con costi molto contenuti. Approfondendo il processo di identificazione della tosse, di solito, i predittori vengono estratti da un audio, precedentemente suddiviso in frames, e ven-

gono quindi utilizzati per addestrare un classificatore. Per identificare la tosse sono stati utilizzati molti predittori diversi. Ad esempio, Pramono et al. [1] hanno estratto feature temporali, come lo zero crossing, feature energetiche, come il livello di energia, e anche caratteristiche spettrali, come spectral Kurtosis, spectral Slope, spectral Centroid. Altri studi, come quello condotto da Miranda et al [2], hanno utilizzato predittori spettrali, confrontando i coefficienti della Short Time Fourier Transform con lo spettrogramma di Mel. Lo spettrogramma di Mel è stato utilizzato anche da Barata et al. [3] insieme ai Mel Frequency Cepstral Coefficient, derivati dallo spettrogramma di Mel utilizzando una trasformata coseno discreta. Gli MFCC sono stati usati anche da molti altri gruppi di ricerca, Di Perna [4], Swarnkar [5] e Tracey [6] per citarne alcuni. Pramono et al. [7], in un altro studio, ha esplorato le prestazioni del Linear predictive coding coefficient, che è una feature nel dominio del tempo e del Tonality Index. Alcuni predittori più complessi sono stati utilizzati da Monge et al [8], che hanno utilizzato i Local Hu Moments, che sono elaborazioni avanzate delle intensità dell'immagine o dell'audio, e sono degli invarianti a traslazione, scala e rotazione.

Dopo l'estrazione delle feature, i predittori vengono utilizzati per addestrare un classificatore, che classifica i dati in classi e può distinguere un evento di tosse da altri suoni. Per eseguire questa operazione, è possibile utilizzare molti algoritmi. La regressione logistica è stata utilizzata da Pramono et al [1] per rilevare la tosse ottenendo una sensibilità del 92%. Le Support vector machines (SVM), sono un altro metodo classico di machine learning, utilizzate ad esempio da Di Perna [4] per rilevare i pazienti con polmonite ostruttiva cronica. Liu et al. [9] e Tracey et al. [6] hanno utilizzato le Support Vector Machines per classificare gli eventi di tosse ottenendo rispettivamente il 91% e l'81% di sensibilità. Lucio et al. [10] hanno usato K nearest neighbour ottenendo una sensibilità dell'87% nel rilevare i colpi di tosse. Anche Monge-Alvarez et al. [8] hanno usato le KNN ma all'interno di un metodo che usa gli alberi decisionali (decision tree). Altri lavori che hanno utilizzato alberi sono Larson, Lee et al. [11], che hanno utilizzato un Random Forest classifier e hanno ottenuto una sensibilità media del 92 %. Lo stesso metodo è stato utilizzato da Breiman et al. [12] raggiungendo una sensibilità pari al 92%. Amrulloh et al. [13] hanno utilizzato un Artifical Neural Network (ANN) per sviluppare un rilevatore di tosse con una sensibilità del 93%. È

possibile utilizzare ANN più avanzate, come Probabilistic Neural Network (PNN) [14] o Current Neural Network (CNN). Quest'ultima è stata utilizzata da Tracey et al. [6] assieme alle SVM, consentendo di ottenere una sensibilità dell'81 %. Miranda et al. [2] hanno utilizzato e confrontato Current Neural Network, Deep Neural Network e Long Short Term Memory Neural Network, dimostrando che le prestazioni variano a seconda delle diverse configurazioni di frame e lunghezza dei segmenti. Infine, l'Hidden Markov Model (HMM), utilizzato da Matos et al. [15], è una tecnica di keyspotting, in cui l'idea è di rilevare le occorrenze di parole chiave in una registrazione continua del parlato. Questo approccio può essere adattato anche per la tosse.
Un problema importante nell'analisi di registrazione audio, e quindi anche per il rilevamento della tosse, è proteggere la privacy degli utenti. Le principali criticità si riscontrano nell'eventuale riconoscimento di chi parla dalle caratteristiche vocali (intonazione, formanti), nella ricostruzione di una conversazione privata o di attività private e nel riconoscimento del genere, dell'accento, della nazionalità e dell'emotività di una persona [16].

Materiali e Metodi

I campioni audio sono stati raccolti da dataset di audio online. In particolare, da Freesound [5] e Commands Speech[6]. 313 campioni di colpi di tosse sono stati estratti dall'audio dataset Freesound ed etichettati manualmente con l'app Matlab AudioLabeler. 156 suoni casuali, come musica o suoni ambientali, sono stati presi da Freesound insieme a 313 parole casuali prese da Speech Commands. Sono stati creati due diversi dataset. Uno, chiamato "Speech & Cough", è stato costruito con 313 eventi di tosse e 313 parole. Il secondo, chiamato "Complete", è stato costruito con 313 colpi di tosse, 157 parole casuali e 156 suoni ambientali di Freesound. Tutti gli algoritmi e le funzioni sono stati imlementati su Matlab R2020b (MathWorks, Inc. USA) usando anche l'Audio Toolbox[7] e il Deep Learning Toolbox[8]. Dopo la creazione dei dataset, l'audio è stato sottocampionato a 16 kHz in modo da avere la stessa frequenza tra i campioni

[5]Progetto e dataset sono disponibili al sito: https://freesound.org
[6]Il dataset è disponibile su: https://storage.googleapis.com/download.tensorflow.org data/speechcommandsv0.01.tar.gz. Copyright Google 2017.
[7]Documentazione del Toolbox disponibile su: https://ch.mathworks.com/help/deeplearning/index.html
[8]Documentazione del Toolbox disponibile su: https://ch.mathworks.com/products/audio.html

audio dei diversi set di dati. Poiché l'ampiezza dei segnali audio è influenzata dalla variabilità inter-soggetto e dalle possibili distanze diverse dal registratore quando una persona parla, è stata applicata una standardizzazione dell'ampiezza per rendere gli audio confrontabili. Un feature extractor object è stato creato su Matlab e utilizzato per estrarre 56 feature dai campioni audio. Innanzitutto, il feature extractor suddivide l'audio in frames della durata di 64 ms che si sovrappongono del 50%. Esempi dei predittori estratti sono i coefficienti dello spettrogramma di Mel, gli MFCC, il pitch (tono), Spectral Kurtosis, spectral centroid, spectral crest, spectral entropy, spectral flatness, spectral flux, spectral roll off, spectral slope and harmonic ratio. Successivamente, i dati sono stati valutati con un test di Lilliefors per valutarne la normalità. Se le feature erano normali è stato utilizzato un t-test per campioni indipendenti. Se le caratteristiche non erano normali ($p < 0{,}05$), è stato utilizzato un Wilcoxon rank sum test. Entrambi i test sono stati applicati per confrontare ciascuna feature tra le due classi ("tosse" assegnata a "1" e "no tosse" assegnata a "0"). Dopo questa valutazione, è stato applicato un algoritmo di minimum redundancy maximum relevance (MRMR) per classificare i predittori in base a un compromesso tra l'importanza dei predittori e la loro mutua informazione. Le 15 migliori features sono state selezionate.

I classificatori valutati sono stati Random Forest, K Nearest Neighbor, Support Vector Machine e Long Short Term Memory Neural Network. I classificatori sono stati addestrati in quattro modalità. Le prime due consistevano nell'utilizzo del dataset "Speech & Cough " utilizzando, da un lato, tutte le 56 feature e, dall'altro, solo le 15 feature selezionate da MRMR. Le altre due modalità consistevano nell'usare il dataset " Complete " con tutti i predittori in un caso e con quelli selezionati nell'altro. La valutazione delle metriche di performance dell'addestramento e del test è stata effettuata utilizzando una 10-fold cross validation. Questo ha permesso di ottenere la mediana dei risultati, avendo così degli indicatori di performance più robusti. Le metriche di performance usate sono accuratezza, precisione e sensitività. La capacità di previsione dei classifier con nuovi dati è stata effettuata con il test set e valutata con accuracy, precision e recall. Inoltre, i risultati sono stati visualizzati con delle curve ROC. Infine, è stato utilizzato un test di Kruskal Wallis per valutare se un algoritmo avesse classificato significativamente meglio di un altro. È stata esplorata la significatività statistica tra

l'addestramento dei classificatori con tutte le feature o con un sottoinsieme di feature. Poi, la significatività tra l'utilizzo dei due diversi dataset per addestrare gli algoritmi. Infine, la significatività tra i classificatori, al fine di valutare se un classificatore aveva prestazioni migliori di un altro.

Risultati e Discussioni

L'analisi statistica dei predittori ha mostrato che non seguono una distribuzione normale. Il test di Wilcoxon ha evidenziato una differenza significativa per tutti i predittori tra le due classi "cough" e "no cough", ad eccezione di tre predittori del dataset "Complete" che non sono risultati significativamente differenti tra le classi. L'algoritmo MRMR ha selezionato le migliori 15 feature. In particolare, per il dataset "Complete", le tre feature con il valore della capacità di predizione più alto individuate sono state " Mel4 ", " MFCC9 " e " SpectralKurtosis ", mentre in " Speech & Cough ", le migliori caratteristiche trovate sono " Mel32 ", " MFCC12 " e " MFCC8 ".
Tutte le metriche di performance dei classificatori sono state calcolate come mediana, con l'intervallo interquartile corrispondente,dai singoli valori ottenuti durante la 10 cross validation. Un esempio è mostrato in tabella 2. Il classificatore che generalmente ha mostrato le migliori prestazioni è stato LSTM. Ha eseguito la classificazione con una sensibilità di 92,4 ± 3,3 % quando testato con il set di dati "Complete", utilizzando tutte le 56 feature. Quando è stato utilizzato il dataset " Speech & Cough ", LSTM ha ottenuto dei risultati persino migliori, 98,2 ± 8,1 %. Le performance di LSTM hanno superato i risultati di altri studi, come Liu et al. [9] e Tracey et al. [6] Lucio et al. [10] che hanno raggiunto, con metodi differenti, rispettivamente il 91%, l'81% e l'87% di sensibilità nel rilevare i colpi di tosse. E anche Pramono et al. [1], Larson et al. [11] e Breiman et al. [12] hanno ottenuto tutti una sensibilità pari al 92 %. I risultati dei due dataset sono stati confrontati per vedere se, l'utilizzo un dataset più semplice come " Speech & Cough ", quindi senza rumori, possa migliorare le prestazioni del classificatore rispetto al dataset " Complete ". I risultati del p-value hanno mostrato che per KNN, quando vengono utilizzate 56 feature, il comportamento si è rivelato opposto al previsto, cioè con una una maggior precision con il dataset " Complete", 86,1 ± 3 % contro 79,8 ± 3,3% del dataset " Speech & Cough ", con p-value minore di 0.001. Nel

caso delle SVM, quando si utilizzano le 15 feature selezionate, la scelta del dataset ha mostrato ancora una volta che il set di dati "Complete" ha fornito più informazioni al classificatore con un conseguente miglioramento delle prestazioni. In generale, per il resto dei casi, la presenza di rumori non ha influenzato i risultati dimostrando che i classificatori sono robusti al rumore e ai suoni diversi dal parlato. Gli altri risultati hanno mostrato che la feature selection eseguita con MRMR in generale non ha influenzato i risultati, il che significa che i predittori scelti hanno memorizzato correttamente la maggior parte dell'informazione. D'altro canto, la selezione delle feature non ha migliorato le prestazioni dei classificatori. L'unica eccezione è quella di KNN, che per il dataset "Complete", è stato influenzato negativamente dalla feature selection. La precision è passata da 82,6 ± 6,2 % a 66,3 ± 2,3 % quando le features sono diminuite. Tutte le performance e la significatività degli algoritmi, quando vengono confrontati tutti gli algoritmi tra di loro, per il dataset "Complete" utilizzando tutte le 56 feature, sono mostrati nella tabella 2. Infine, gli algoritmi sono stati reppresentati tramite una

Tabella 2: *Confronto tra gli algoritmi: "Complete" dataset, 56 features*

Metrics	Classifiers				P values					
	RF	**KNN**	**SVM**	**LSTM**	**RF-KNN**	**RF-SVM**	**RF-LSTM**	**KNN-SVM**	**KNN-LSTM**	**SVM-LSTM**
Accuracy	88.2 ±1.7%	86.1 ±3%	87.6 ±1.3%	94.4 ±2.8%	0.151	0.915	0.068	0.465	**0.001***	**0.010***
Precision	76.3 ±9%	82.6 ±6.2	78.5 ±14.3%	96.8 ±2.4%	0.752	0.992	**0.001***	0.894	**0.014***	**0.001***
Recall	82.5 ±8.5%	75.5 ±10.5%	80.5 ±14.3%	92.4 ±3.3%	0.152	0.984	**0.035***	0.302	**0.001***	**0.012***

curva ROC (Receiver Operating Curve) per poter confrontare visivamente le diverse performance. Il caso più rappresentativo è quello con il dataset "Complete" e con tutte le 56 features utilizzate, mostrato in figura 2. Poichè l'area sotto la curva (AUC) è un indice di performance, è possibile osservare che LSTM è il miglior classificatore, seguito da RF, SVM e KNN, che ha ottenuto le peggiori prestazioni.

Conclusioni

In conclusione, la rete neurale LSTM è stato il miglior classificatore. Le metriche di prestazione della LSTM hanno superato i risultati di altri studi, come Liu et al. [9], Tracey et al. [6] e Lucio et al. [10] che hanno raggiunto, con metodi differenti, rispettivamente il 91%, l'81 % e l'87 % di sensibilità nel rilevare eventi di tosse. Tutti i metodi hanno mostrato robustezza sia con un dataset con solo tosse e parole sia con

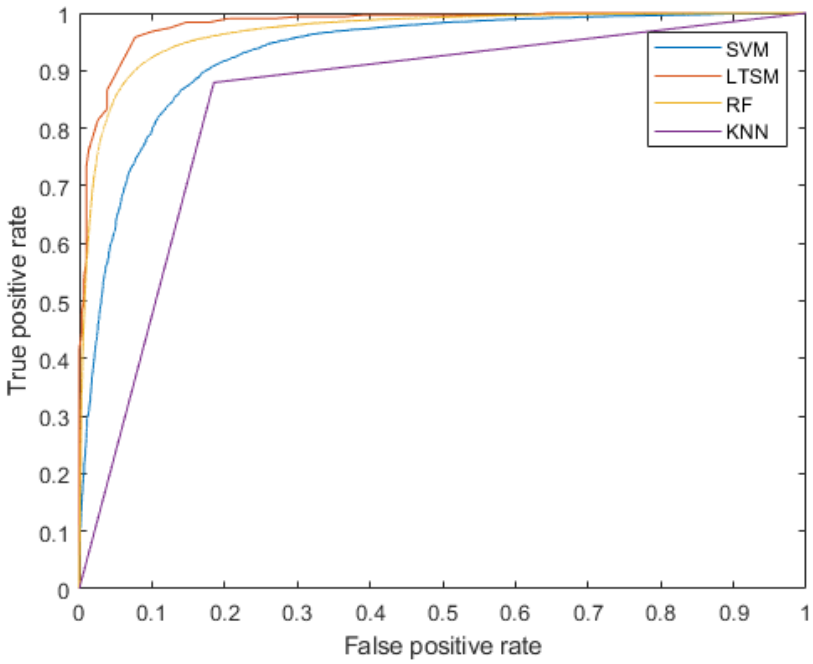

Figura 2: *Curva ROC: "Complete" Dataset, 56 Features*

uno contenente tosse, parlato e rumori esterni. La feature selection non si è rivelata un passaggio obbligatorio, soprattutto per la rete neurale LSTM. In ogni caso, poiché non diminuiva le prestazioni, può essere applicata per ridurre la dimensionalità del problema. Utile per diminuire i tempi di calcolo. Il miglior classificatore è risultato essere LSTM e quindi è la scelta indicata per eventuali sviluppi futuri. Questi risultati potrebbero essere sfruttati per un ulteriore lavoro in cui il classificatore LSTM addestrato può essere utilizzato per una classificazione in tempo reale dei colpi di tosse. Il metodo finale potrebbe essere implementato in un'applicazione per smartphone che possa estrarre automaticamente le feature vocali in tempo reale durante le conversazioni telefoniche o in background durante la giornata. Dalle feature estratte, potrebbe poi contare il numero di eventi di tosse al giorno in modo continuo e non invasivo, con la finalità di valutare lo stato di salute delle vie respiratorie.

Chapter 1

Introduction

The constant increase in life expectancy [17], poses important challenges to researches, whose goal is to improve not only life duration but quality of life as well. As shown in figure 1.1, the Western population life expectancy at birth increases every year. This trend shows that the number of elderly people increases as well, together with their need of assistance from the healthcare system, specifically, the need and demand for care due to the physiological and/or pathological decline of physical and cognitive functions. To face this increasing demand, there is a rapidly developing interest in the

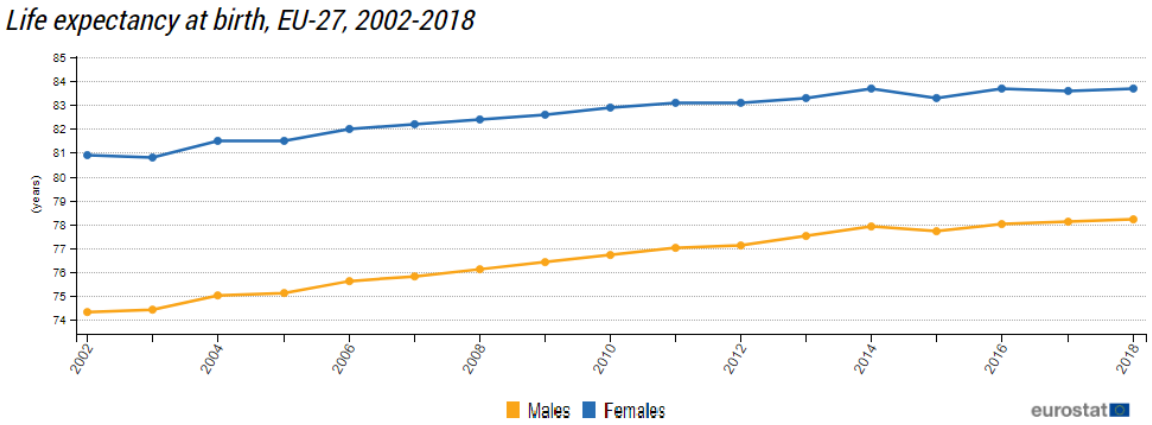

Figure 1.1: *Life Expectancy at birth Europe-27 -Source Eurostat.*

use of information and communications technology (ICT) for the non constrained and remote monitoring of physiological and health variables at home. Evidence for this interest are major research programs in Europe, initiated under the general umbrella

of the Advanced Informatics in Medicine (AIM) program, which has as a major theme the application of Telematics Systems in Health Care [18]. Omnipresent available ICT, such as smartphones or wearable devices, can be used to unobtrusively monitor patients' health condition as these devices are everywhere and almost always carried by individuals [19]. In this direction, the mobile-health (m-health) technologies are also important. The World Health Organization (WHO) defined m-Health as "medical and public health practice supported by mobile devices, such as mobile phones, patient monitoring devices, personal digital assistants , and other wireless devices" [20]. m-Health can contribute to achieving universal health coverage (UHC) through making services available to remote populations and communities and providing mechanisms for data exchange on patients and services. It can be used to increase access to and provision of health services in areas where there is little infrastructure to support the internet (or other technologies) or traditional health services, but where mobile communications technology infrastructure has been prioritized. Moreover, supplying the technology for mobile communications is cheaper than providing in-person services. The use of mobile devices has been increasing exponentially internationally – from 2.2 billion global mobile phone subscriptions (82 per 100 inhabitants) in 2005 to more than 7 billion (120 per 100 inhabitants) by the end of 2015 [21]. Therefore, their use in healthcare is surely a good way to improve general health on the global scale. Moreover, m-Health is valuable also in high income countries because it permits to be always connected with physicians and sanitary personnel, at any distance.

Nowadays, healthcare providers are facing growing challenges in patient diagnostics and information gathering due to in-adequate infrastructure and insufficient number of health-care professionals to perform such operations. In fact, the traditional way of patient information gathering is largely based on text and dependent on self-report surveys and infrequent doctor consultations. Using mobile device based on distributed sensors can improve remote diagnostics and healthcare information gathering [22]. In particular, this year, the Covid-19 pandemic highlighted the limited availability of accommodation and specialized personnel in healthcare structures. Furthermore, the need to maintain patients far from the hospitals, when possible, and to give care in remote to avoid the contagious spread, strongly demonstrated the importance of

telemedicine and dislocation of care solutions. Mobile-health technologies, which exploits smartphones to monitor health conditions, can improve and facilitate the job of doctors and nurses. This is possible principally because m-health technologies are cheap, easy to use and allow continuous control of the patients situation. Mobile devices permit a cheap, easy and possibly all time recording of voice and respiratory sounds. Furthermore, due to the great spread of mobile phones and home assistant devices that allows human-machine interaction, it is more and more common extracting relevant information from speech other than just the message itself. For example, it is possible to extract the speaker identity, ethnicity, emotion and age so that it can be used to improve customer services and offer tailored products [23].

From an healthcare point of view, speech can be useful in finding out patients aging, cognitive impairments, stress, emotions, such as sadness and depression, and finally can give a general idea of their quality of life, for example from their social interaction and from the symptoms recorded during the day. Such symptoms may include tiredness during speaking, number of cough events, number of sneezing, number of pauses. Speech can be used to guess people age just using neural networks, Fedorova et al. [24] were able to find out the age with a maximum mean absolute error of 6.35 years. Ultimately, on the wave of Covid-19 pandemic many studies focused on the respiratory tract. For example, speech analysis can be used to detect respiratory sounds in order to have a general overview of a person life quality and respiratory health. Respiratory sound analysis can be made with smartphones so that it would not only guarantee scalability and cost-efficiency, but would also provide a reproducible measurement of the respiratory activity [3]. A basic indicator of respiratory tract health it's the number of cough events, occurrence of a typical respiratory symptom, over time, whose value is supposed to be low, almost zero times for day, in a healthy subject. Also, cough is one of the most frequent symptoms among various pathologies and it impacts negatively the daily life. It can reveal respiratory disorders in all population but it especially impacts the categories most at risk, like seniors or persons with respiratory diseases, hence it's a good index of general health. Cough is controlled through the cerebral cortex. Hence, a cerebrovascular disease can block the control of the neural pathway and result in an elevated threshold for the cough reflex. For this reason, cough reflex

sensitivity can be used as indicator of dementia [25]. Decreased cough reflex sensitivity has been established, also, as one of the risk factors for the onset of aspiration pneumonia in elderly people [26].

In this context, this work aim is to generally assess breathing quality of a subject during his daily life using a classification algorithm that identifies the cough events from audio recordings. Moreover, this work tests and validates those different classification algorithms. This preliminary study focused on studying which are the optimal choices of audio features, machine learning classifiers and segments length to assess the cough detection. The features investigated are spectral formants present in literature: Mel Spectrogram, Mel Formant Cepstral Coefficients and other spectrals features. Then, the algorithms, both classical (Random Forest with Bagging, Support Vector Machine, K-Nearest Neighbours) and deep machine learning method (Neural Network with Long Short-Term Memory), were evaluated. The algorithms were trained and validated on voice recordings downloaded from two online datasets: Freesound[1] and Commands Speech[2]. This work has to be considered a preliminary work towards the development of an application for smartphone, able to count the number of cough events per day in a transparent and unobtrusive manner.

[1]The project and the dataset are available at: https://freesound.org

[2]The dataset is available from https://storage.googleapis.com/download.tensorflow.org data/speechcommandsv0.01.tar.gz. Copyright Google 2017.

Chapter 2

State of the art

2.1 Speech analysis in quality of life assessment

As briefly introduced in Chapter 1, speech analysis made by machine learning algorithms can help detect cognitive pathologies and conditions that negatively affect the general quality of life of an individual. Among all the possible applications of speech analysis some of the most related with health and aging are assessing stress, since it is a condition that influences every aspect of the life of a person, and some other conditions like dementia, Parkinson and loneliness that strongly affect elderly people. Finally, it is important to underline how cough is connected with dementia and general every day health. In the following sections the main applications of speech analysis will be briefly discussed.

2.1.1 Stress

Stress and its management are important indicators of general health, at every age. It is a serious problem that can adversely affect different life situations and also carry a wide range of health-related diseases, including cardiovascular disease, cerebrovascular disease, diabetes, and immune deficiencies [27]. Over the past decades the effect of stress onto the speech production has been well studied. Respiration is strongly correlated with some emotional situations. For example, if a subject is in a stressful

situation, the respiration rate increases. Usually, this brings to an increase of subglottal pressure during speech, and consequently the pitch (or fundamental frequency of the signal) increases during the voiced section [28]. Moreover, a respiration rate increase can cause a shorter duration of speech between breaths which affects the speech articulation rate.

Kurniawan et al. [29] used several subsets to find stress correlated information in speech, including smoothed energy computed using the overlapping time frames, voiced and unvoiced speech, pitch, Mel Frequency Cepstral Coefficients (MFCCs) that represent human audio perception and Relative Spectral Transform - Perceptual Linear Perception [30]. All these features were shown to be robust to static noise [29]. Pathak et al. [16] simply trained an Support Vector Machine extracting MFCC from a small dataset with audios from stressing situations like police emergencies or after traumatic event such COVID-19 pandemic, acid attack survivors speech, other judicial victims of various cases. They obtained a good 83.4% accuracy using just 100 samples indicating speech as a viable indicator for detecting stressing scenarios. Simantiraki et al. [31] proposed a new set of features to detect stress based on the spectral tilt of the glottal source and of the speech signal itself. They used the three most probable Probability Density Function (PDF) of the spectral slopes from the glottal source and from the speech signal, obtained on a word level.

2.1.2 Fatigue and sleepiness

Sleepiness is an important state which affects safety, performance and comfort [32]. Sleepiness levels can be simply monitored thanks to the variations in signal bandwidth, since perceived timbre, dullness and muffling effects in speech change according to the sleepiness level, resulting in different perceptual bandwidths for non sleepy and sleepy audio. Greeley et al. [33] in a preliminary study confirmed the existence of a dependence between formant frequencies and fatigue. They used the Mel frequency cepstrum coefficients, together with their first and second derivatives, to obtain the formants frequencies. MFCCs are coefficients obtained dividing the audio sample in windows, in which a Discrete Fourier Transform is applied to acquire a periodogram. Then, a Mel filter is used to transform from Hz scale to Mel scale and finally, the

MFCCS are extracted from the spectrogram applying a Discrete Time Cosine transform. This final step is made to uncorrelate the Mel filter values. They found that certain formant frequencies are more related to sleep latency tests results, the gold standard for sleepiness testing, which calculates the time necessary for a subject to fall asleep [34]. Hence, it is useful to analyze the MFCCs of different phonemes and determine the phonemes that show good variations in the spectral domain, this means that these phonemes are a more accurate fatigue indicator. It's possible to observe that the correlation between Trial 1 (12 hrs awake) and Trial 10 (39 hrs awake) is higher (0.82) when compared to the correlation between Trial 1 and Trial 21 (78 hrs awake) (0.19). This is an indication that the MFCC components change as the subject gets increasingly fatigued [35], suggesting that speech can also be used for fatigue assessment. Gunsel et al. [36] used cepstral features obtained from a Short Time Fourier Transform to train a Support Vector Machine classifier that distinguishes between sleepy and well-rested subjects.

2.1.3 Loneliness

Recently, there has been a rise in suicide rates and opioid use, lost of productivity, health care costs and mortality in the United States due to loneliness. Lonely individuals show an increased risk of cognitive decline and development of dementia, which negatively impacts the economy [37] [38]. Additionally, the Covid-19 pandemic and the resulting lockdowns have increased the amount of time people have spent in solitude, worsening this situation [39]. For example, this year, the 85% of residents living in an independent senior housing community in San Diego reported moderate to severe levels of loneliness [40]. Language processing techniques and machine learning models can predict loneliness. In Badal et. al project [41], natural language processing (NLU) is used to perform an unbiased quantitative assessment of expressed emotion and sentiment, that has been compared with the UCLA Loneliness Scale, that is the most commonly used measure of loneliness [42]. Their study [41] focused on 80 independent senior living residents aged between 66 and 94 years, with a mean age of 83 years. Natural language patterns and machine learning were used to systematically examine long interviews from many individuals and explore how some speech features can help to

detect emotions like sadness that consequently may indicate loneliness. One advantage of using an Artificial Intelligence (AI) is that it is not open to bias and lack consistency as humans naturally are. The tool developed is powerful because it does not use only a dictionary-based approach – for example searching for specific words that reflect fear – but it looks at specific patterns across the words used in the responses. They made audio interviews with questions created with the purpose to assess loneliness, then all the material was transcripted in order to permit an examination with natural language processing tools such as IBM's Watson Natural Language Understanding (WNLU) software, that quantifies sentiment and expressed emotions [43]. WNLU uses deep learning to extract metadata from keywords, categories, sentiment, emotion and syntax. Sentiment (positive and negative) is represented as a number (continuous range between -1.0 and 1.0), indicating if the speaker is in disagreement or agreement with the current context of the conversation, for example a question from the interviewer. Emotion is instead represented by a five-tuple of sentiments (sadness, joy, fear, disgust, anger) with values ranging from 0.0 to 1.0 in proportion with the predicted strength of each emotion [44]. They mapped the distribution of emotions expressed in the responses to their questions. The respondents who acknowledged feeling lonely were more likely to express sadness in their responses. Expression of sentiment and other emotions (disgust, anger, joy, fear) did not differ between lonely versus non-lonely groups, as it is visible in fig 2.1.

The algorithm could predict loneliness with 94% precision when compared against

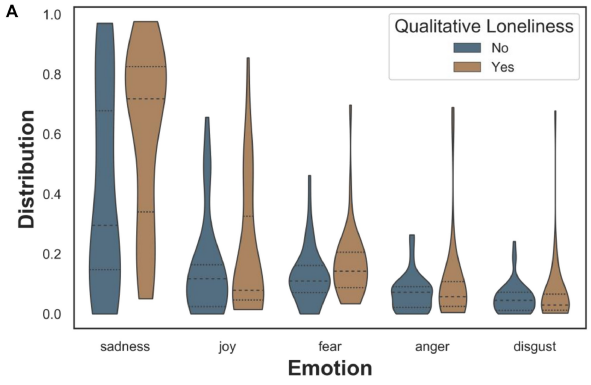

Figure 2.1: *Emotional composition of response to Question 1 ("Do you ever feel lonely and if so, how often?") by (A) Qualitative loneliness. Dashed lines in the middle of distribution indicate median (second quartile) and dotted lines indicates first and third quartiles in the distribution [41].*

the 'quantitative model, the UCLA Loneliness Scale compiled by the participants that is considered perfectly accurate.

2.1.4 Parkinson Disease

In Gomez-Vilda et al. work [45], it is described how the articulation dynamics in speech is correlated with the kinematic behavior of the jaw-tongue biomechanical system, encoded as a probability distribution of an absolute joint velocity. This distribution may be used in detecting and grading speech from patients affected by neurodegenerative illnesses, such as Parkinson Disease (PD). The correlation exists because vowel positions are related to formant descriptors, that are acoustic properties, and so, through formant kinematics, it is possible to estimate articulation dynamics (Carmona et al. [46]). Phonation is controlled by laryngeal nerves (vagus), whereas articulation depends mainly on jaw, lingual and facial nerves. PD tremors depend on neuromotor pathways and so they can affect phonation and articulation. Specifically, the Authors calculated from the first and second MFCC formant the absolute kinematic velocity (AKV), whose probability density function is a good marker to establish differentiation between stable and unstable articulation induced by PD. A dataset of sustained vowels recorded from Parkinson Disease patients was compared with similar recordings from normal subjects. The AKV probability density of the jaw-tongue system was extracted from each utterance. A Random Least Squares Feed-Forward Network (RLSFN) was used as a binary classifier in a leave-one-out strategy. As result, they obtained an error rates under 0.6% (Accuracy over 99.4%) [45].

2.1.5 Dementia

Speech parameters can also help to detect more specific conditions connected with aging, like dementia. In this case, it is useful to assess Mild Cognitive Impairment (MCI), a condition that often precedes dementia. MCI is a condition characterized by cognitive impairments that are visible on clinical exams or formal cognitive testing, but that are not yet producing a clinically significant impairment in daily functioning [47]. Additionally, due to the fact that Alzheimer's disease (AD) is part of a diffuse

cognitive disorganization and language impairment, MCI is one of the first symptoms that are developed in this condition. Caielli [48] has shown that voice breaks, speech rate, phonation, syllables duration and pauses duration are valid predictors for discriminating people with a slightly compromised cognitive function from seniors with an preserved cognitive function. Therefore, speech can be considered an important source of information to detect, assess and diagnose early dementia. In fact, dementia affects speech at the linguistic level (what is said) and paralinguistic level (how it is said). In more detail, temporal parameters of speech, such as longer hesitation time and lower speech rates, are indicator of the first phase of dementia [49]. People with AD tend to decrease speech fluency, by increasing the number and decreasing the length of voice segments [50] as well as by increasing the number of word finding pauses and the percentage of voiceless segments [51]. Another interesting way to detect dementia is using the cough reflex sensitivity. This parameter is estimated as the number of induced coughs after inhalation of a solution of physiological saline with increasing concentration of citric acid. The acid elicits the cough reflex and the sensitivity is estimated by the lowest concentration of citric acid that is able to arouse two or more coughs and the lowest concentration of citric acid that is able to arouse five or more coughs [52]. Cough reflex sensitivity and urge-to-cough deterioration can be used as indicators of dementia with Lewy bodies development [25]. This is possible because cough is controlled via the nucleus of the solitary tract in the brainstem and through the cerebral cortex. Urge-to-cough (UTC) is the sensation that initiates and inhibits the reflexive cough and is controlled by cerebral cortex as well. So, the occurrence of a cerebrovascular disease could impede the control of the modified pathway from the upper cough centre to the cough reflex arch, resulting in an elevated threshold for the cough reflex evoked by citric acid inhalation. For this reason, cough reflex sensitivity and Urge To Cough (UTC) are negatively correlated with cognitive function.

2.2 Respiratory System

An overview about the respiratory system structure and about how speech and cough are generated from it is needed. The respiratory system is composed by many different

parts that work together to permit breathing. The airways goal is to deliver air to the lungs. The airways are a made of different parts, which are shown in figure 2.2. They start with mouth and nose, that are openings from which the air is pulled from outside into the respiratory system. The sinuses are the hollow areas between the bones in the head that help regulate the temperature and humidity of the air inhaled. The pharynx, or throat is the tube that delivers air from mouth and nose to the trachea (windpipe), which is the structure that connects throat and lungs. Then, the bronchial tubes from the trachea divide the air flux into the two lungs. The lungs are the two organs that remove oxygen from the air and pass it into the blood. Then, the carbon dioxide is taken from the blood and passed into the air. This exchange of oxygen and carbon dioxide takes place in the alveoli, tiny air sacs in the lungs. From the lungs, the bloodstream delivers oxygen to all the organs and other tissues and the air breathed out expels carbon dioxide, water vapor and other waste gases [53]. Muscles and bones

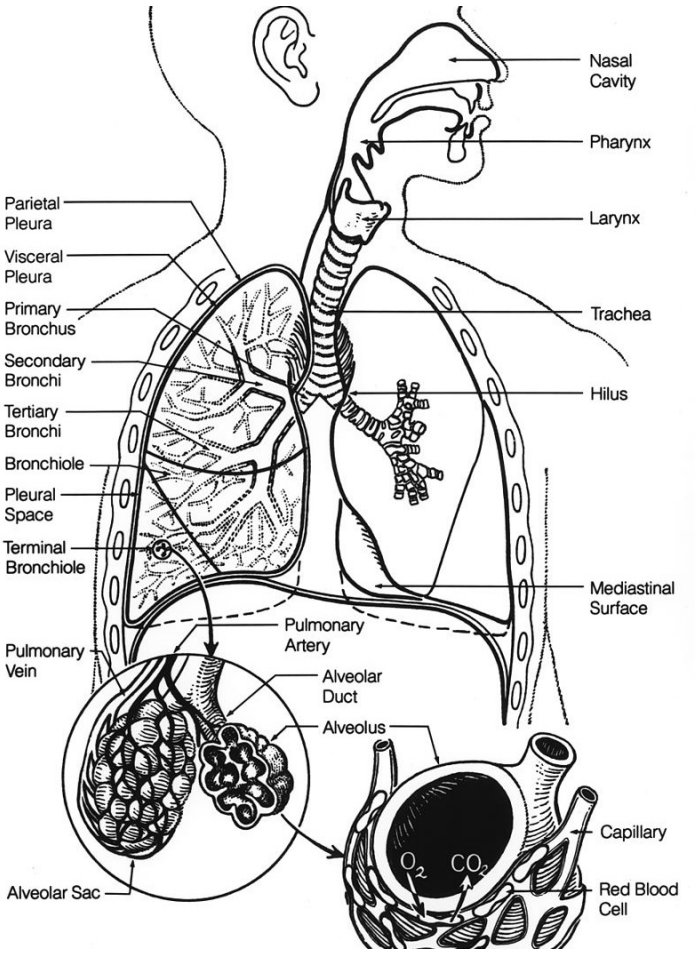

Figure 2.2: *Respiratory System [54].*

help move the air inhaled into and out of the lungs. The most important muscle in the respiratory system is the diaphragm, that pulls the air in and out. During inhalation, the diaphragm and the ribs muscles contract enlarging the thoracical cavity respectively in vertical and in horizontal dimension. The enlarged cavity creates a pressure fall and the air flows through the airways until the lungs. During exhalation the diaphragm and intercostal muscles relax. Then, the chest and abdomen thanks to their anatomical elasticity return to a resting position and the air flows out.
Disorders of the respiratory system can be classified into several groups:

- Airway obstructive conditions: emphysema, bronchitis, asthma...
- Pulmonary restrictive conditions: fibrosis, sarcoidosis, alveolar damage, pleural effusion...
- Vascular diseases: pulmonary edema, pulmonary embolism, pulmonary hypertension...
- Infectious, environmental and other "diseases": pneumonia, tuberculosis, asbestosis, particulate pollutants...
- Severe acute respiratory syndromes, such as COVID-19 [55].

Other functions carried out by the respiratory system, other than breathing, are protection, with chemical and mechanical defenses, and speaking. To protect the body from external elements present in the air, there is a chemical response. The respiratory epithelium can secrete a variety of molecules that help defend the lungs. These include secretory immunoglobulins, collectins, defensins and other peptides and proteases. These secretions can act directly as antimicrobials to help keep the airway free of infections. Chemokines and cytokines are also secreted to recruit the traditional immune cells to the site of infections.
There is also a mechanical response to defend the body from external elements, cough reflex and sneezing are induced by the irritation of nerve endings in the nasal passages or in the airways. These responses cause air to be expelled forcefully from the trachea or the nose, respectively. In this manner, irritants caught in the mucus are expelled from the respiratory tract or moved to the mouth where they can be swallowed [56].

Speaking and phonation are allowed by the movement of air through the larynx, pharynx and mouth. In particular, the vibration of air flowing across the larynx, where vocal cords are located, results in sound. Hence, the respiratory system, in particular the vocal tract, is at the basis of speech communication. The mechanism behind speech and the cough reflex are explained in detail in the following subsections, subsection 2.2.1 and subsection 2.2.2.

2.2.1 Vocalization

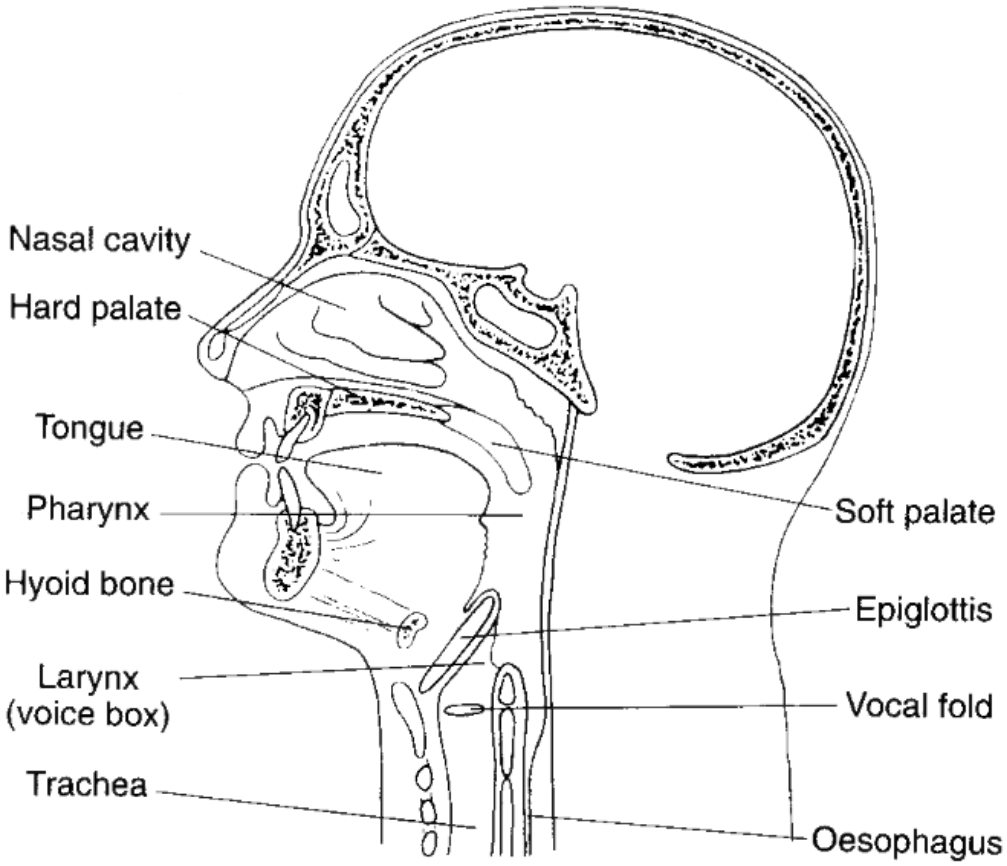

Figure 2.3: *Vocal Tract [57].*

The human vocal tract system starts from the vocal cords, or glottis, and ends at the lips. A general representation of the vocal tract is shown in figure 2.3. The nasal tract is the region between nostrils, nose holes and velum, which is near nasal pharynx. The vocal tract acts as a resonator and as a filter for the sounds produced by the vocal folds. Speech production starts with the entrance of air into the lungs, which

is known as normal breathing. At this step, sounds or speech are not produced yet. The vocal cords are then tensed and, While the air inside the lungs is expelled, the vocal cords vibrate due to the air pressure in the glottis. The glottis converts the air into quasi-periodic pulses by opening and closing. The quasi-periodic pulses are then frequency-shaped while passing through the pharynx, mouth cavity and nasal cavity. Finally, the sound is determined by the positions of the jaw, tongue, velum, lips and mouth [58]. More in detail, the sound is created thanks to muscles that can change the size and the shape of the vocal tract. Every time the size and shape change anywhere along the vocal tract, the change has an effect on the original sound which is filtered or amplified. A speech sentence usually is made of a part that carries information and a silent or noisy part, which does not contain verbal information. The verbal part can be divided into voiced and unvoiced speech. The first consists mainly of vowel sounds produced by the closing and the opening of the vocal cords adjusting the tension and modifying the air passing from the glottis. The latter is instead generated by forcing the air through a constriction in the vocal tract to produce turbulence. The ability to discriminate these three parts is fundamental in speech signal analysis [48].

2.2.2 Cough Reflex

As mentioned in section 2.2 cough is a normal reflex action of the body to clear the throat and the respiratory tract from foreign substances or irritants. During coughing, contraction of the smooth muscles in the airway walls narrows the trachea. This increases the expired airflow rate so that any irritant particle or mucus is removed. Cough may affect the subject life during all day, with an annoying and potentially painful effect. Moreover, it is one of the most frequent symptoms for many pathologies. Chronic cough, which is defined as the cough that persists for more than six weeks, is usually a symptom of chronic bronchitis, asthma, allergy, chronic obstructive pulmonary disease (COPD), gastroesophageal disease (GERD). Acute cough, lasting from few days or to two-three weeks, instead, may be caused by cold or upper respiratory tract infections like flu or pneumonia [59]. Counting the number of cough events over time, can be a valid index of the respiratory tract condition. A reliable evaluation of cough is therefore needed so that the severity of cough in a particular patient and the effectiveness

of treatment can be assessed. This cough severity test has so far relied mainly on subjective measures, such as cough reflex sensitivity, explained in subsection 2.1.5, and on the patient's perception of the symptoms, assessed by cough analogue scores, quality of life questionnaires, cough symptom scores and patient's diaries. Cough can be used as an index to detect dementia but also to assess the general health of the respiratory tract.

The cough reflex starts from an irritation of the afferent cough receptors in the airways. Efferent nerve impulses stimulate the diaphragm, the intercostal muscles and the larynx to produce the forceful explosive expiration of cough. Once triggered, the cough reflex consists of four phases [60]:

1. Inspiratory phase, which consists in an initial deep inspiration and glottal closure.

2. Compressive phase, consisting in a contraction of the expiratory muscles against the closed glottis.

3. Expulsive phase, in which there is a sudden glottis opening with an explosive expiration and fast expiratory airflows.

4. Wheezing phase, in which a wheeze or "voiced" sound is emitted.

Previous studies show that the glottis closure and expulsive phase can be repeated several times without any inspiration [11].

2.2.3 Cough duration and frequency

The average duration of a cough is approximately between 300 ms and 650 ms. Obstructed airflow can produce noises with frequencies up to 20 kHz but cough characteristic and most energetic sounds are associated with frequencies below 10 kHz. Therefore, a way to extract the main characteristics of cough sounds consists in downsampling the signals to 22.05 kHz after applying an anti-aliasing filter. Its spectrum exhibits a high-energy peak around 400 Hz and a secondary peak between 1000 and 1500 Hz as shown in figure2.4, in the left spectrogram the first peak is highlighted in the white box 1 and the second peak in the box 2 [3].

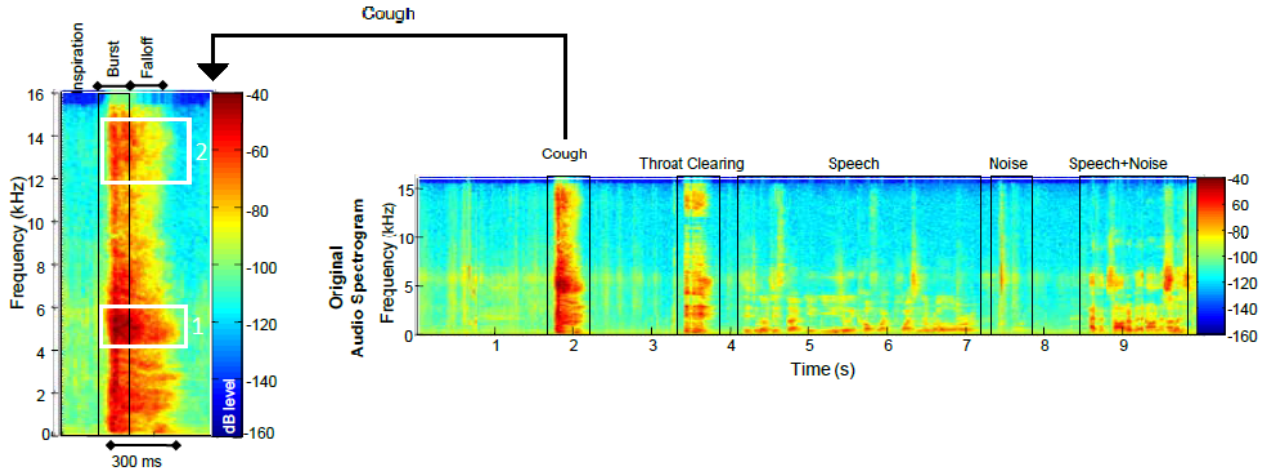

Figure 2.4: *Cough Spectrograms, adapted from Larson et al. [11]. On the left, an example of cough spectrogram is shown. At the right, an example of spectrogram of cough and non-cough audio sounds is shown.*

2.3 General Cough Detection

The most common technique for estimating cough frequency is to have patients self-reports using numeric scoring (0-5) or Visual Analog Scoring (VAS), where the subject is asked to mark on a 100 mm scale between 'no cough' and 'the worst cough severity' [61]. However, numerous studies have shown self-report to be highly inaccurate. The number of cough events a patient self-reports has been shown to be significantly influenced by other factors such as patient's mood, vigilance, and expectations so that his/her perception of the cough severity is not objective [62]. Cough diaries comprise questions relating to cough frequency but they correlate poorly with objectively measured cough frequency [63]. Moreover, patients cannot accurately track trends in their cough frequency from hour to hour, or when they are asleep. For these reasons, more objective methods have been developed for counting cough events. Many of them require expensive and bulky equipment, like the chest wall device developed by Kraman et Al. [64], or require paid annotators to listen to recordings and manually annotate cough sounds [62]. Starting from the 1950s, researchers developed methods for measuring thoracic pressure changes (airflow from the mouth) in order to obtain unbiased measures of cough frequency [65]. Semi autonomous methods exist: an example is the "Overnight Cough Monitoring System", which consists in attaching an air-coupled microphone to the chest wall or over the trachea when the participant is sleeping [66].

Kraman et al. [64] created an accelerometer-based system placed on the participant's chest wall, but required someone to manually count coughs based on the visualization of the acquired data. VitaloJAK [67] is a commercial product that uses a piezoelectric sensor attached to the chest wall to detect cough events. The reported true positive rate was 91.3-99.5% after an initial calibration. The VivoMetrics Lifeshirt [68] is another commercial product that incorporates various physiological sensors to monitor breathing rate, heart rate, activity, posture, and skin temperature. It can be used to detect coughs adding an extra throat microphone to the existing sensors system. They reported true positive rate equal to 78.2% and false positive rate of 0.4%. Each of these methods have the same problems: users must wear specialized equipment and sensors on the chest wall or around their body that increase cost, make the system uncomfortable and limit the use of the system in ambulatory settings. Audio based systems, are instead easy to deploy in an ambulatory setting and can be made extremely low cost. These systems have become more accurate and less expensive over recent years. Barry et al. created a system called the Hull Automated Cough Counter (HACC) [69], using a microphone and a wearable recording device. Using as predictors mel-frequency cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCC). The LPCCs are obtained from a simple recursion of the LPCs, that are spectral envelopes of a digital signal of speech in compressed form, using the information of a linear predictive model. However, they recorded audio signals in an outpatient clinic for only one hour per person, which is a relatively controlled and noise-reduced environment. Similarly, Matos et al. created a system called the Leicester Cough Monitor (LCM) [15], which uses a microphone with a portable audio recorder. They used MFCCs (with derivatives) as features to a Hidden-Markov Model (HMM). Moreover, audio based system can be easy implemented in smartphones making possible monitoring the users every time and everywhere with very low costs.

2.4 Acoustic features

As seen in the previous Section 2.3, audio based systems use engineered methods and algorithms to extract speech and sounds features. This step permits to represent infor-

mation in a more effective and lower dimensional way with respect to the simple raw audio, facilitating the subsequent learning and generalization steps. Feature extraction is a method of constructing combinations of the variables to solve the problem of having a large datasets, the need of a high computation power and to avoid overfitting, while still describing the data with sufficient accuracy. Overfitting occurs when the classifier fits the training data too precisely and loses his generalization capacity. In the task of automatic detection of cough events, many features have been used in literature. One of the simplest and most used features is Short Time Fourier Transform (STFT). STFT is a Fourier transform used to extract the sinusoidal frequency and phase from local section of a signal changing over time [70]. In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length, which can be overlapped, and then compute the Fourier transform separately on each shorter segment, revealing the Fourier spectrum on each one. The plots of the changing spectra as a function of time is called spectrogram. Starting from STFTs it is possible to obtain the Mel spectrogram if a transformation from Hertz to Mel is made and a Mel filter Banks is applied subsequently, which is usually a set of 20-40 triangular filters. A deeper explanation of this feature can be found in subsection 3.2.1. Other features that can be obtained following these steps are the Mel Frequency Cepstral Coefficients (MFCCs), as already explained in subsection 2.1.2. Starting from the Mel spectrogram a Discrete Time Cosine Transform is applied in order to obtain the coefficients that, thanks to this transformation, became uncorrelated. MFCCs are a representation of the short-term power spectrum of a sound, based on the linear cosine transform of the log power spectrum on the Mel frequency scale. They are composed by twelve coefficients plus the frame's logarithmic energy. Usually, first and second derivatives of the coefficients are calculated as well and used as features. In the Liftered-MFCC a liftering on the raw MFCC weights is made using the following formula:

$$w_i = 1 + \frac{M}{2} sin\left(\frac{\pi_i}{M}\right) \tag{2.1}$$

Gammatone Frequency Cepstral Coefficient (GFCCs) are very similar to MFCCs, differing only in the type of filter banks applied: instead of Mel filter Gammatone, filter

banks are used. A gammatone filter is a linear filter described by an impulse response that is the product of a gamma distribution and sinusoidal tone (pure sound). The gammatone impulse response is visible in fig. 2.5. The main idea of the GFCCs is based

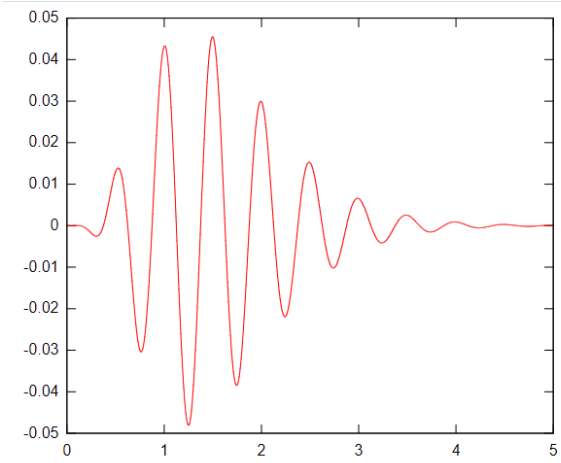

Figure 2.5: *Gammatone Filter Response. Author Roger Wilco*

on an auditory periphery model that imitates the human cochlear filtering mechanism. The auditory model is represented by a bank of Gammatone filters that decompose the input speech into a Time-Frequency representation.

Local Hu Moments are very advanced and engineered features that were initially used for Visual Pattern Recognition in images[71]. They are particular weighted averages of the image or audio intensities. Hu moments are derived from central moments and they are invariant with respect to translation, scale, and rotation.

These features have been applied in many works of which some significant examples are shown here. Barata et al. [3] implemented a new method using Mel Spectrogram. Monge et al. [8] used Local Hu Moments in their studies. Pramono et al. [7] preferred to use LPCC, Linear predictive coding coefficient, which is a time domain feature that can be used to predict the next signal value from the previous samples, described in section 2.3, with sensitivity 86,6% and specificity 99,42%. They used also Tonality Index, that is the arrangement of pitches and it's useful to distinguish speech from cough, and Spectral Flatness, which is a measure of how noise-like a sound is, obtaining a sensitivity of the 86,8% and a specificity of the 99,42%. Miranda et al. [2] used STFT, Mel scale filter banks and MFCC and L-MFCC. The Mel filter banks is calculated applying 40 triangular filters on a Mel-scale to the power spectrum to extract frequency bands.

STFT and MFB features provided the best cough detection accuracies. MFCC first and second derivatives did not prove to enhance performance and could be omitted, thereby reducing feature dimensionality. MFCC were used by Miranda et al. [2] and Di Perna et al. [4], as well. Other possible features, used for example by Pramono [1], are Zero-crossing rate that represents the frequency of sign-changes in a signal, the Crest Factor, which is the ratio of the value of a peak in a waveform relative to its root mean square (RMS) value and the Energy Level, which, in an audio frame, is calculated as the RMS of the frame. Since cough is an explosive sound, it has bursts of energy increase in short time for this reason Spectral Roll-Off (SRO) is useful in distinguishing sounds with different energy distributions. Spectral Skewness, which is a measure of the asymmetry in the power spectrum of a signal on its mean and it is useful in understanding if the power distribution in PSD estimate will have more density in lower or higher frequency. It is computed as the third moment divided by the cube of the standard deviation. Spectral Kurtosis Coefficient is a measure of the peak of the power spectrum and describes the shape of the probability distribution of the energy in the power spectrum of a signal. It is computed by using the fourth moment. Spectral centroid represents the equivalent of the center of mass in a spectrum and is computed as a weighted mean of the spectrum. Swarnkar et al. [5] used other spectral features such as formant frequencies, Kurtosis, and B–score together with MFCC features for cough detection.

2.5 Classification algorithms of cough events based on acoustic features

As already mentioned, cough detection has been under research and development since the 1950s [72]. Fig. 2.6 gives an overview of the state of art in automated cough detection algorithms [3].

In machine learning, classification is the process of categorizing a given set of data into classes. The goal is predicting the class of future given data points. The classes are often referred to as targets, labels or categories. In order to perform classification many methods have been used, which can be subdivided into two main categories: the

Author	Recording Device	Algorithm	Subjects (Coughs)	Cough Type	Sensitivity	Specificity
Coyle et al. 2005 (LifeShirt) [16]	Contact Mic. + Sensor Array	*no details available*	8 (3645)	Reflex	78.1%	99.6%
Barry et al. 2006 (HACC) [17]	Lapel Mic.	PNN	15 (2000)	Reflex	80%	96%
Birring et al. 2008 (LCM) [3]	Lapel Mic.	HMM	15 (1836)	Reflex	91%	99%
Vizel et al. 2010 (PulmoTrack) [18]	Piezoelectric Belt + Lapel & 2 Contact Mic.	*no details available*	12 (*no details available*)	Voluntary	96%	94%
Drugman et al. 2011 [19]	Contact Mic.	ANN	22 (2304)	Voluntary	94.7%	95%
Larson et al. 2011 [13]	Smartphone Built-In Mic.	RF	17 (2558)	Reflex	92%	99.5%
McGuiness et al. 2012 (VitaloJak) [20]	Piezo Sensor	Median Frequency Threshold	10 (*no details available*)	Reflex	97.5%	97.7%
Swarnkar et al. 2013 [21]	Matched Pair Low-Noise Mic.	NN	3 (342)	Reflex	93.44%	94.52%
Liu et al. 2014 [22]	Lapel Mic.	GMM-HMM & GMM-RBM	20 (> 2549)	Reflex	90.1%	88.6%
Amrulloh et al. 2015 [23]	Low-Noise Mic.	TDNN	24 (2090)	Reflex	93%	98%
Amoh et al. 2016 [24]	Wearable Sensor / Contact Mic.	CNN & RNN	14 (627)	Voluntary	87.7%	92.7%
Monge-Alvarez et al. 2018 [5]	(2x) Smartphone Built-In Mic.	*k*-NN	13 (*no details available*)	Reflex	88.51%	99.7%

Figure 2.6: *Classification Algorithms [3].*

HACC: Hull Automatic Cough Counter
LCM: Leicester Cough Monitor
ANN: Artifical Neural Network
GMM: Gaussian Mixture Model
TDNN: Temporal difference Neural Network
RNN: Recurrent Neural Network
PNN: Probabilistic Neural Network
HMM: Hidden Markov Model
RF: Random Forest
RBF: Radial Basis Function
CNN: Convolutional Neural Network
KNN: K Nearest Neighbour

classical classification methods and the more modern deep learning ones, that exploits neural networks. Most of the approaches developed are based on feature extraction. In conventional machine learning, features are typically selected and manipulated with the goal of reducing the data size to make the classification or solve the specific issue. Then, the features are used as input for the algorithm to train and make possible the learning of a pattern represented in the data, such as coughs. Some advantages of this approach are that a smaller data representation requires less memory, less computation power and lastly reduces the risk of over fitting the model on the training data, which consists in scarce generalization with new samples. This comes with the risk of losing valuable information and limiting the representation power of the learning algorithm. By contrast, deep learning, shifts the complexity of handcrafted feature engineering towards model optimization. Deep learning permits to use better computation power and data complexity [73]. For this reason, these architectures have increased their importance in various domains, such as speech recognition [74] and image classification [75]. Usually, they are based on artificial neural networks with multiple layers of nonlinear functions and information processing [76]. The difficulty in using deep learning is that deeper networks need the collecting of larger amounts of data from real subjects. The

main classification methods will be briefly explained in the following section, while a focus on the ones that have been adopted in this work will be given in the chapter 3.

2.5.1 Classical Algorithms

With the name of classical algorithms are considered the machine learning methods used before the arrival of deep neural networks. Some of the most used are Support Vector Machine, Logistic Regression, K-Nearest Neighbours, Decision trees, specially expanded with Random Forest or Bagging techniques. Among these methods, Principal Component Analysis (PCA) is often used, that is the computation of the principal components of the data, which is the direction that best fits the data while being orthogonal with the previous components and that explain the most variance in the data. PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point only onto the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible [77].
Support Vector Machine (SVM) algorithm has the objective to find a hyperplane in an N-dimensional space (N, the number of features) that distinctly classifies the data points [78]. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. The objective of SVM is to find a plane that has the maximum margin, i.e the maximum distance, between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane will be attributed to different classes. Support vectors are the data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, the margin of the classifier is maximized. The support vectors are the only points actually used to build the SVM's hyperplane. The loss function that helps maximizing the margin is hinge loss. The loss is 0 if the predicted value and the actual value are of the same sign. If they are not, the loss value is computed. It also adds a regularization parameter to the cost function. The objective of the regularization parameter is to balance the

margin maximization and loss. In SVMs the margin distance has to be maximized and the regularization part has to be minimized. In the regularization part a penalty cost is given for every Support vector inside the margins or misclassified [79]. Support Vector Machines algorithm has been used by Di Perna, Spina et al. [4] to detect cough in chronic obstructive pulmonary patients. In their work they detected cough (with value of Area Under ROC curve of 0.916) and COPD cough (with best value of 0.950 AUC), trying different implementations of the SVM algorithm. Liu et al. [9] used Gammatone Cepstral Coefficient features with SVM classification of 903 coughs from 4 subjects resulting in sensitivity and specificity of 91% and 95% respectively. Tracey et al. [6] developed an algorithm for cough detection to monitor patients' recovery from tuberculosis. They extracted MFCCs from the audio signals of 10 test subjects. These were used to detect coughs with a combination of artificial neural network (ANN) and support vector machine (SVM) classifiers achieving an overall sensitivity of 81%.

Logistic regression algorithm uses a linear equation with independent predictors to predict a value. It is an adaptation of the linear regression for classification[80]. Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent (x) and dependent (y) variable. Based on the given data points, it creates a line that best models the points. The line can be modelled based on the linear equation: $y = a_0 + a_1 * x$. The predicted value can be anywhere between negative infinity to positive infinity, hence, it is not optimal for classification. The outcome improves if the variable used is a class variable, 0-no, 1-yes. Therefore, a sigmoid function is used to compress the predicted value between 0 and 1. Pramono et al. [1] used a Logistic regression classifier to make cough detection, cough classification and whooping sound detection. The output provides a pertussis likelihood diagnosis. The performance of the proposed algorithm is evaluated using audio recordings from 38 patients. It can automatically detect individual cough sounds with 92% accuracy.

K-nearest neighbors is a non-parametric method used for classification and regression. The basic logic behind KNN is to explore its neighborhood, assume the test datapoint to be similar to training ones and derive the output. In order to make the prediction, k neighbors are used. In case of KNN classification, a majority voting is applied over the

k nearest datapoints whereas, in KNN regression, the mean of k nearest datapoints is calculated as the output. So, it's better to select odd numbers as k so that the voting can't produce an even result. KNN is a lazy learning model where the computations happens only runtime. There is no training involved in KNN. During testing, the k neighbors with minimum distance will take part in classification/regression. KNN represents an easy and simple machine learning model with few hyperparameters to tune. Some disadvantages are that the number of neighbors should be selected previously and, if the sample size is large, the computational cost during runtime could be high[81]. The distance function could be Euclidean distance,Manhattan distance, Hamming Distance, Minkowski distance. K-nearest neighbors has been used many times, for example by C.Barcelò and J. Monge-Alvarez [8] using Hu Moments as feature. In order to tune the algorithm, they used three types of Metric trees to permit binary classification and speed up the process. Moreover, they used three distance functions: Euclidean, Manhattan and Euclidean2 (squared) resulting in same accuracy for the two Euclidean methods (96.1%). They also observed that normalization had a positive impact on the accuracy of the classifier. They obtained an accuracy that goes from 90% at -15dB noise to signal ratio until 98% for 0 dB SNR. Lucio et al. [10] extracted 79 MFCC and Fast Fourier Transform (FFT) coefficients and used k-Nearest Neighbor (KNN) for classification. From a dataset acquired from 50 individuals, their algorithm achieved sensitivity of 87% in classifying 411 cough sounds with specificity of 84%.

Decision tree is a tree based algorithm used to solve regression and classification problems. The tree is inverted so it starts from a homogeneous root node and goes to highly heterogeneous leaf nodes that are the output. Regression trees are used for dependent variable with continuous values and classification trees are used for dependent variable with discrete values. Decision tree is derived from the independent variables, with each node having a condition over a feature. The nodes generator decides which node to navigate next based on the condition. Once the leaf node is reached, an output is predicted. The right sequence of conditions makes the tree efficient [82]. Entropy/Information gain are used as the criteria to select the conditions in nodes. A recursive, greedy based algorithm is used to derive the tree structure. A random forest consists

of multiple random decision trees. Two types of randomnesses are built into the trees. First, each tree is built on a random sample from the original data. Second, at each tree node, a subset of features are randomly selected to generate the best split. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes the model's prediction. The fundamental concept behind random forest is the wisdom of crowds. So that, a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between models is the key [83]. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. The bagging repeatedly selects a random sample from the original training set, to create new training sets and uses those to fit the trees. This means that it is possible for an already chosen observation to be chosen again [84]. Trees are often used for cough detection sometimes inside other algorithms, [8] sometimes to directly classify. Larson, Lee et al. [11] after event extraction, trained a random forest classifier [12] on the feature set of the training fold. All features are obtained as eigenvalues from a PCA application and they are all used to train the classifier. The maximum number of trees was set to 500, since usually over fitting is not a problem. RF classifier has empirically been shown to work as well as support vector machines and neural networks, but it is less sensitive to parameter variation [12]. After a 5 cross-validation they obtained sensitivity equal to 92%. Di Perna et al. [4] obtained the best performance with XGBoost, an ensemble method based on decision trees. The classifier achieved an accuracy of 91,6%. XGBoost is Gradient Boosting and weak learner, meaning that it uses simple trees but applied many times with a gradient function.

2.5.2 Deep Learning Algorithms

Deep learning algorithms differ from classical methods since they are based on artificial neural networks (ANN) and use multiple layers to progressively extract higher-level features from the raw input. In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. They use the neural network to find associations between a set of inputs and outputs. The basic structure of a neural network is shown in fig.2.7. ANNs are composed of input, hidden, and output

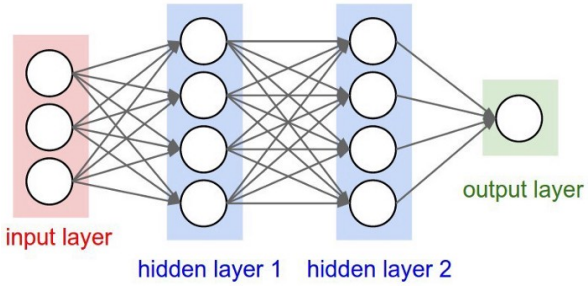

Figure 2.7: *Neural Network basic structure .*

layers — all of which are composed of "nodes". Input layers take in a numerical representation of data (e.g. images), output layers output predictions, while hidden layers are correlated with most of the computation [85]. More in detail, neural networks can be used with different configurations and implementations.

Convolutional Neural Networks (CNN) were inspired by biological processes as the connectivity pattern between neurons resembles the organization of the animal visual cortex [86]. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication. The convolution depends on the activation map of the layer that looks for a particular characteristic of the data (e.g. countours of an image). The activation function is commonly a ReLU layer, which returns 0 if it receives a negative input, while if it receives a positive value, it returns that same value back. It's followed by a max pooling operation, which consists in selecting the maximum element from the region of the feature map covered by the filter. Hence, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map, reducing the feature map dimension. Then, the final convolution gives out the outputs [87].

A deconvolutional neural network (DNN) is a neural network that performs an inverse convolution model. The work of a deconvolutional neural network could be described as constructing layers from an image in an upward direction or as "reverse engineering" the input parameters of a convolutional neural network model.

Long Short Term Memory networks (LSTMs) are a type of Recurrent Neural Networks (RNNs), capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) [88]. LSTMs are explicitly designed to avoid the long-term dependency problem, that instead affects simple RNNs. The input vector is splitted

into time sequences and then looped through the sequences to train the network. Since it is the same set of neurons that are trained in every time instance, it is necessary a "state information" across time.

A probabilistic neural network (PNN) [14] is a feedforward neural network, which is widely used in classification and pattern recognition problems. In the PNN algorithm, the parent probability distribution function (PDF) of each class is approximated. Then, using PDF of each class, the class probability of a new input data is estimated and Bayes' rule is then employed to allocate the class with the highest posterior PDF to new input data. It was introduced by D.F. Specht in 1966 [89]. In a PNN, the operations are organized into a multilayered feedforward network with four layers: Input layer, Pattern layer, Summation layer and Output layer. A Markov stochastic process in system states is a random process in which the probability of transition to a system state depends only on the state of the immediately preceding system (Markov property) and not on how it came to that state [90]. A Hidden Markov Model (HMM) is a Markov chain in which states are not directly observable. More precisely: the chain has a number of states that evolve according to a Markov chain, each one generating an event with a certain probability distribution that depends only on the state. The event is observable but the state is not. Hidden Markov models are particularly known for their applications in recognizing the temporal pattern of spoken speech, handwriting, texture recognition, and bioinformatics (eg HMMer). These methods are generally used in machine learning and for automatic cough segmentation as well. Martinek et al. [91] extracted several time, frequency and entropy features and used a decision tree to discriminate between voluntary cough sounds and speech. They used data from 20 subjects, collecting 46 coughs from each subject, and reported median sensitivity and specificity values of 100% and 95% respectively. However, their method is subject–dependent since the subjects are required to cough at the beginning of each recording in order to obtain individual cough signal patterns. Barry et al. [69] used linear predictive coding coefficients (LPCC) with a probabilistic neural network (PNN) classifier to create an automatic cough counting tool called Hull Automatic Cough Counter (HACC). This successfully discriminated between cough and non–cough events from 33 subjects with a sensitivity of 80% and specificity of 96%. Tracey et al. [6] developed an algorithm for

cough detection to monitor patient recovery from tuberculosis. They extracted MFCCs from the audio signals of 10 test subjects. These were used to detect coughs with a combination of artificial neural network (ANN) and support vector machine (SVM) classifiers achieving an overall sensitivity of 81%. Pramono et al. [1] proposed a convolutional neural network (CNN) architecture and exploited an ensemble implementation to further boost the performance in a device agnostic preserving manner. Barata et al. [3] used Ensemble CNN performing with accuracy of 88,7%. Among the most prominent CNN architectures, there is VGG (Visual Geometry Group) net [92], which was able to significantly improve image recognition accuracies by pushing the depth to 19 layers and increasing the amount of convolutional layers. Miranda et al. [2] evaluated their features with DNN, CNN and LSTM classifiers, using the area under the receiver operating characteristic curve (AUC) as a performance measure. Their classifiers architectures were inspired by previous work on small-footprint keyword spotting [93]. The DNN implementation consisted of three hidden layers with 128 ReLu units per layer. The CNN had one convolutional layer and two 128-unit hidden layers. Two 832-unit layers were adopted for the LSTM implementation. The output is for all models a two-class softmax layer. Their experiments indicate that using a single long audio segment during feature computation provides better performance than the subdivision into smaller overlapped segments. They obtained a maximum of 95,5% accuracy with CNN. They used MFCC obtaining accuracy 93,5% using LSTM network. Additionally, better performance was obtained using frames longer than 25 ms window. Matos et al. [15] proposed the use of hidden Markov models to automatically detect cough sounds, using data collected from continuous ambulatory recordings. HMMs is a statistical method used for characterizing the spectral properties of a signal and has been mainly used in speech recognition systems because of its ability to represent variability and time varying nature of signals [94]. The average detection rate was 82% with a false alarm rate of seven events/h, when considering only events above an energy threshold relative to each recording's average energy. These results suggest that HMMs could be used to the detect cough sounds from ambulatory patients [15]. HMMs represents well the time varying characteristics of cough sounds. The approach, based on HMMs, is a key spotting one, so the idea is to detect occurrences of keywords in a continuous

speech recording [95]. Amrulloh et al. [13] used ANN classification to develop a cough detector using over 1400 cough sounds from 14 subjects from a non–contact recording system achieving sensitivity of 93% and 98% specificity.

2.5.3 Algorithms used on Smartphones

This section simply shows some previous studies that detected cough sounds using the microphones integrated in smartphones, underlying how this can be done and also which algorithms perform better in these scenario. Pramono et al.[1] chose to use a logistic regression model to perform the classification of the isolated cough events. The features extracted from the training set, as explained in previous section, 2.4, are the 13 MFCCs and other spectral attributes. Some of the extracted sound events posses a length that are not typical of a cough sound. Only sound events with length typical of a cough sound are selected to be used in the automatic classification while others are discarded by setting a threshold for duration. This expedient permits a reduction in the calculation effort, an important characteristic due to the generally limited smartphone computation power. [1] In Barata et al. study is well underlined how using a different smartphone can affect the results even though, in general, the scalability between devices is possible. They also observed that the quality of the devices' recordings is reflected in the results, meaning that the accuracy values on low quality devices recordings are lower in comparison to the accuracy values of other higher quality devices [3]. The Khomsay et al. paper presents a cough detection method using Principal Component Analysis (PCA) and Deep Learning Networks (DLN). PCA is used to perform feature extraction before sending the data to train a model by DLN. This step helps data reduction and speeds up the calculation, which is very important in smartphones. They obtained an accuracy of 94.4% [96]. Other approaches have focused on computationally efficient solutions providing cough detection algorithms for the smartphone [8] [11].

2.6 Privacy Issues in voice recordings

In every audio recording application, and so for cough detection as well, an important issue is protecting the privacy of the users. The main criticalities are the possible recognition of the speaker from voice characteristics (pitch, formants), the reconstruction of a private conversation or of personal life activities during the day and the recognition of a person's gender, accent, nationality and emotional state[16]. Many past works in audio privacy tried to hide some cues about the speakers and conversations around them with the purpose of making not possible for a machine learning algorithm to reconstruct valuable information from the feature sets. Therefore, Wyatt et al. have devised audio features that can successfully hide speech intelligibility, while simultaneously providing cues for prosody and recognition of conversations [97]. Most of the related audio privacy works attempt to preserve certain quantities while providing poor features for modern speech recognizers [98]. Chen et al. [99], on the other hand, used linear prediction to replace vowels in speech, while keeping environmental noises as cars and running water intelligible to subjects. Larson et al. [11] attempted to make the speech unintelligible, while preserving the possibility to reconstruct cough sounds. They used a different methodology that consists in applying PCA on the audio spectrum obtaining that, with 25 components, cough can be well classified and the 84% of the spoken words can not be recognised with a reconstruction algorithm. Best results in order to protect privacy of speech were obtained by using less than 15 components, reaching more than 95% of the words as intelligible. In this work, the approach followed to protect the user's privacy is similar to Wyatt et al.[97] one. In the future application, the features could be used in real time for the cough detection and then being deleted. Moreover, the single segment is too small to contain more than one word, thus ensuring that it is not possible to build a word and moreover a sentence using the segment that the algorithm is processing. This guarantees the privacy for the user about his personal conversation and every day life. In this case, the cough event could be considered as a keyword spotting system, that detects if a specified keyword has occurred in a recording, thus obtaining a system that can detect only the presence of the keyword, which is the cough event in this work, without having access to the

speech recording. This solution enables privacy preserving in applications for extracting statistics about the occurrence of specific keywords in a collection of recordings, while not learning anything else about the speech data [16].

2.7 Segment and frame length

In order to choose segment and frame length, Miranda et al. [2] made a complete study using different frame/segment combinations. They performed a grid search in order to evaluate the chosen features (STFT, MFB, MFCC and L-MFCC) extracted with different segment and frame duration for each classifier, DNN, CNN and LSTM. They used as segment lengths values of 160, 320, 480, 640, 800 and 960 ms, while values of 24, 32, 64 and 128 ms for frame lengths. Within each segment, frames were extracted with a 50% overlap. Segments were extracted from the audio event with a 25% overlap until a maximum length of 1 s was reached, which corresponds to the maximum cough duration [100]. All features were computed using the logarithm of the squared magnitude of the spectral coefficients, with a 40-dimensional mel scaling applied for MFB, MFCC and L-MFCC. For both types of MFCCs, the first 13 cepstral coefficients were used as features. Although there is no single best configuration of segment and frame duration, larger values generally exhibit better performance across all considered features and classifiers. Some configurations with short segments also performed well when using a LSTM classifier, producing the best outcome for L-MFCC. The success of long frame lengths is an important observation, as it differs from established practice in automatic speech recognition systems and also from most previous cough detectors, which almost always use 25 ms frames. This is consistent with the fact that deep architectures have the ability to learn from low level features [2]. 640 ms segments and 64 ms frames perform well across all classifiers and features considered, especially for LSTM with a 0.938 AUC.

2.8 Aim and structure of the work

This work fits in the general field of automatic sounds and speech analysis. In particular, it consists in the complex task of identifying respiratory sounds and extracting meaningful information from them. The aim of this work is to develop a classification model to assess the status of the respiratory tract through the count of cough events. The count of the number of cough events is, in fact, a repeatable and easy to measure metric that can be easily assessed once the classification algorithm works. Moreover, cough itself is an important symptom of many common diseases and, if frequent, can strongly affect a person's life quality. For these reasons, the focus of this work was to evaluate and compare different classifiers with the aim of detecting cough events inside audios containing speech and environmental sounds. The algorithms compared in this work were both classical machine learning methods (Random Forest with Bagging, Support Vector Machine, K-Nearest Neighbours) and deep machine learning methods (Neural Network with Long Short-Term Memory). These methods were mostly used in the literature. The LSMT exploits the new direction of study that uses deep learning with recurrent networks to perform classification. This work investigate the effect of using two different datasets to train the classifier. Specifically, it studied the impact of adding to a dataset with only cough sounds and speech, different noises and sound similar to the cough sounds that the classifier is trying to detect. Moreover, the importance of features selection was explored as well, along with the significance of every single feature in giving information able to divide the sample belonging to the "cough" class from the ones belonging to "no cough " class. Then, the final goal was to study the classifiers with their behavior in a complete panorama of possible sounds and to find the classifier with the best performances. The outcome of this work should be considered as a starting point for the future development of a smartphone application that automatically counts cough events during the day or during phone calls, respecting the user's privacy and without interfering with daily life.

The thesis is structured as follows. The first chapter is the introduction, which gives and overview of the context in which the work is inserted. It explains the actual situation of aging of the population and of the issues caused by the Covid-19 pandemic to

the healthcare system. It illustrates the possible solutions, including assisting the patients remotely, using information and communications technology (ICT) , especially smartphones, to make possible to monitor the users at anytime and at a low cost. Then, the importance of speech in detecting diseases and assessing the quality of life is introduced. Finally, the importance of cough and its role in general quality of life and health of the respiratory tract is explained.

In the second chapter the state of art is overviewed. It starts from the great impact that speech analysis had on the prediction and assessment of various diseases, emotions and problematic conditions. Then the attention was focused onto cough physiology, with particular interest on its phases and temporal and spectral characteristics. At this point, the methods used to count cough events, from mechanical based to auditory based, are listed. In the category of auditory based, the features extracted in other works aiming to detect cough events from audio recordings are presented. The same was made after illustrating the various classification methods that can be used to discern cough from other sounds. Then, some studies that employed smartphones recordings and not from professional microphones or ambulatory recordings are shown. Finally, privacy protection techniques and frame-segment settings are described.

The third chapter illustrates the methods used in the work. Firstly, the datasets used and the audio labelling are explained. Then, data preprocessing, including downsampling and standardization are described. After, in the statistical analysis the features normality and significance between classes are evaluated. At this point, all the features extracted are explained in detail, as well as the classifiers analysed. In the last part, cross validation and the performance metrics are illustrated.

In the fourth chapter, the results are analyzed. In particular comparing the performances between the algorithms in order to assess the best ones. Then, also the significance between the two dataset is tested, along with the significance between the classifiers performances when the features are selected or not. All the tables with the performance metrics and the p-values significance are shown. In the last part, the ROC curves are shown as well.

The last chapter concerns the conclusions. It explains again the aim of the work at the light of the most important results achieve. Finally the limits of the work and the

possible developments are explained.

Chapter 3

Materials and methods

3.1 Audio Dataset

3.1.1 Freesound, ESC500 and Speech Commands Dataset

In order to have a sufficient amount of audio data for the study, many online datasets were searched and scanned. Among the possible datasets, the Freesound and the Speech Commands Dataset were chosen to be used in this work. It is important to underline that both exploit the AudioSet Ontology defined by Google to organize the labels. The ontology includes many different areas: Music, Animals, Sound of things, Natural sounds, Channel environment and background, Source-ambiguous sounds and finally, Human sounds. Inside this last folder respiratory sounds, such as cough, sneeze, sniff and breathing, are included as well.

Freesound Dataset Kaggle 2018 Freesound Dataset Kaggle 2018 is an audio dataset containing 11,073 audio files annotated with 41 labels according to the AudioSet Ontology, as mentioned previously. All audio samples in the dataset were gathered from Freesound and were uncompressed PCM 16bit, 44.1 kHz, mono audio files, of different length. **ESC50** is a labeled collection of 2000 environmental audio recordings suitable for bench marking methods of environmental sound classification. The dataset consists of 5-second-long recordings organized into 50 semantical classes (coughing and breathing included). **Speech Commands Dataset** is an audio dataset of spoken

words designed to help train and evaluate keyword spotting systems. Its primary goal is to provide a way to build and test small models that detect when a single word belonging to a set of target words is spoken, with as few false positives as possible, discriminating from background noise or unrelated speech. This dataset is composed by 65034 audios of 1 second duration of 32 different words, with a frequency of 16KHz and mono format.

3.1.2 AudioDataset Creation

In the Matlab R2020b (MathWorks, Inc. USA) workspace the audioDatastore object, that helps in the management and elaboration of audio files collection thanks to embedded functions and libraries, was added. In order to use this element, the audios from the Speech Command and Freesound datasets were reworked in the following way: in a new folder named 'nocough', which corresponds to the name for the label, were uploaded 313 random words from the first dataset. In another folder named 'cough', 311 audios that were manually classified from Freesounds audioset using the Matlab AudioLabeler App as explained in section 3.1.3, were added. In this way we obtained 2 folders that Matlab reads as 2 different labels corresponding to the ones we want for our classification of cough events against the other sounds. This first built dataset has been named "Speech" because it included the speech words from Speech Command as well as the cough sounds. These last speech sounds have been chosen randomly, and consists of words like "bed", "down", "five", "no", "stop", "wow", for a total of 32 different words. This workframe has been used to create another dataset. This second one has been called "All" because it contains also environmental, music sounds or daily life noises in addition to cough and speech. In particular is composed by 313 cough sounds, 157 words from Speech Command and finally the new samples from Freesound that are 156 sounds. In total the dataset has 622 audios as well.

3.1.3 Manual Labelling

Because the audios collected from Freesound present more than one cough event, and sometimes also other sounds, it is necessary to extract and label every single event

before using it. To perform this operation the Matlab tool called AudioLabeler was used. This application permits to create every type of labels, from boolean to string, and to select in the audio the Regions of Interest (ROI) where the sound that correspond to the label is present, as visible in the yellow areas in fig. 3.1. All the audios were exported in a mat file. When the mat file it's uploaded, a function can see the ROI start and end time for every label and use it to divide the audio areas where cough is present from the areas without and from the silenced parts or where other noises are present.

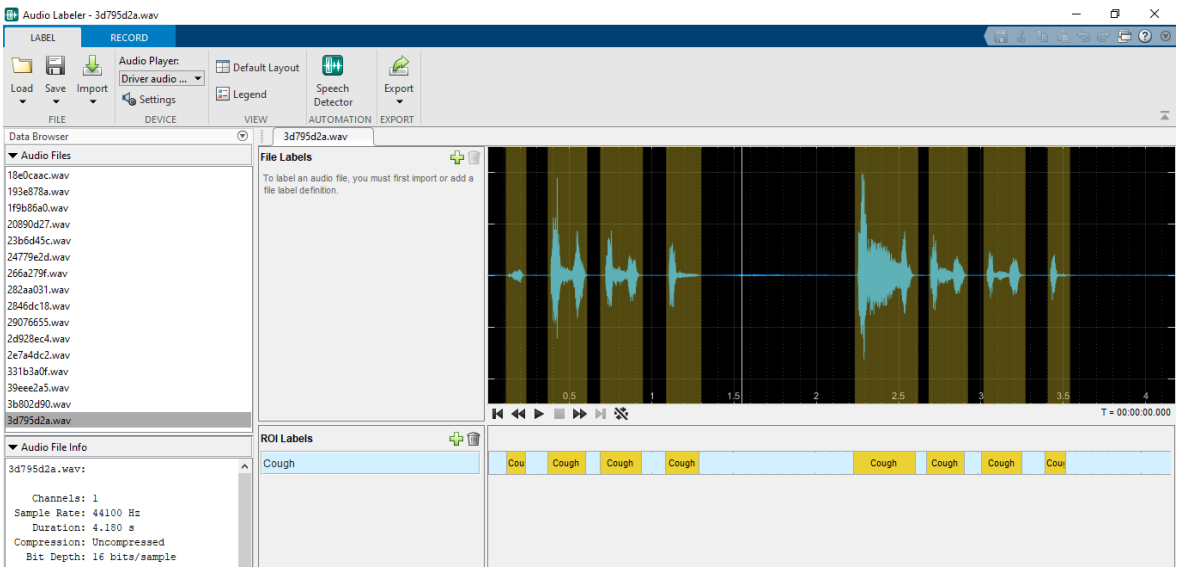

Figure 3.1: *AudioLabeler App*

3.2 Data analysis

An important part of the pre-processing of the signals consists in downsampling of the audios taken from Freesound, that were recorded at 41.1 KHz, to 16 KHz, as the Speech Command Audiodataset files. This has been done to provide coherence in the dataset but also to speed up the calculations. This downsampling is also justified by literature that shows how the sounds we are interested to classify contain frequencies lower than 8 KHz. Therefore, by choosing 16 KHz the Shannon theorem is fulfilled and important information are not lost. Standardization was performed because the audio signals amplitude is different for each recording and, in general, for each subject (due to the

fact that not all the subjects spoke with the same voice intensity). Using this method, the mean of the signals is set to 0 and the standard deviation to 1. Standardization was performed according to the following formulation:

$$y_i = (x_i - \mu_x)/\sigma_x \tag{3.1}$$

where x_i is a sample from the original signal, μ_x and σ_x are respectively the signal's mean and standard deviation, y_i is the corresponding sample of the resulting standardized signal.

3.2.1 Features Extraction

In order to have a compact and easy to use instrument to perform features extraction, a streamline audio feature extractor by Matlab was used. Using this tool makes it easy to insert the window length, the overlap and the features to extract. The frames dimension was chosen according to the values used in literature, in particular, that used by Miranda et al. [2]. The final choice was to select a 64ms frame length with 50% overlap between adjacent frames. With this setting, privacy can be guaranteed thanks to the use of very short frames and small segments of 640 ms in which the features are extracted and the classification is made. At the end, they could be deleted immediately after. The Matlab Audio Toolbox has been used to perform the feature extraction and to analyze the audios. In the following sections, the specific features extracted by the audioFeatureExtractor are explained in detail.

Pitch

Pitch is the relative highness or lowness of a tone as, perceived by the ear, which depends on the vibration frequency of the vocal cords. Pitch tracking refers to the task of estimating the contour of the fundamental frequency F0, i.e. the harmonic corresponding to the lowest frequency present in the voice signal. Such a system is of particular interest in several applications of speech processing, such as speech coding, analysis, synthesis or recognition [101].

Harmonic Ratio

Harmonic ratio is a measure which allows to distinguish odd harmonic energy (such as clarinet) from even important harmonic energy sounds (such as trumpet). The formulation is expressed as:

$$HR = \frac{\sum_{h=1:even}^{H} \alpha^2(h)}{\sum_{h=2:odd}^{H} \alpha^2(h)} \tag{3.2}$$

Spectral Flatness

Spectral flatness, or tonality coefficient, is a measure used to characterize an audio spectrum. It quantifies how much a sound is tone-like with respect to being noise-like. [102] It's called tonality coefficient with the meaning that the number of peaks in a power spectrum opposed to the white noise flat spectrum. A high spectral flatness (approaching 1.0 for white noise) indicates that the spectrum has a similar amount of power in all spectral bands — this would sound similar to white noise, and the graph of the spectrum would appear relatively flat and smooth. A low spectral flatness (approaching 0.0 for a pure tone) indicates that the spectral power is concentrated in a relatively small number of bands — this would typically sound like a mixture of sine waves, and the spectrum would appear "spiky". The spectral flatness is calculated by dividing the geometric mean of the power spectrum by the arithmetic mean of the power spectrum, i.e.:

$$\text{Flatness} = \frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\frac{\sum_{n=0}^{N-1} x(n)}{N}} = \frac{\exp\left(\frac{1}{N}\sum_{n=0}^{N-1} \ln x(n)\right)}{\frac{1}{N}\sum_{n=0}^{N-1} x(n)} \tag{3.3}$$

where x(n) represents the magnitude of bin number n. Note that a single (or more) empty bin yields a flatness of 0, so this measure is most useful when bins are generally not empty [103].

Spectral Kurtosis

Kurtosis measures the flatness of a distribution around its mean value. It's obtained from the 4th order moment:

$$Kurt[X] = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{\mathrm{E}\left[(X-\mu)^4\right]}{\left(\mathrm{E}\left[(X-\mu)^2\right]\right)^2} = \frac{\mu_4}{\sigma^4}, \tag{3.4}$$

X is the sample value in analysis, μ is the mean value and σ is the variance [103]. Different values of the kurtosis correspond to different distributions' shapes:

- K=3 for a normal distribution;
- K<3 for a flatter one;
- K>3 for a more "spiky" one;

Spectral Entropy

Spectral Entropy is defined as the Shannon entropy of the power spectral density (PSD) of the data:

$$H(x, sf) = -0fs/2P(f)log2[P(f)] \tag{3.5}$$

Where P is the normalised PSD, and fs is the sampling frequency [104].

Spectral Flux

Spectral flux is a measure of how quickly the power spectrum of a signal is changing, calculated by comparing the power spectrum for one frame against the power spectrum of the previous frame[105]. More precisely, it is usually calculated as the 2-norm (also known as the Euclidean distance) between the two normalised spectra. Calculated this way, the spectral flux is not dependent upon overall power (since the spectra are normalised), nor on phase considerations (since only the magnitudes are compared). The spectral flux can be used to determine the timbre of an audio signal, or in onset detection [106].

Spectral Crest

Spectral crest is related to flatness and is computed as the ratio of the maximum value within the band to the arithmetic mean of the energy spectrum value [103].

$$SCM\left(nband\right) = \frac{max(a\left(k \in nband\right))}{\frac{1}{K}\sum_{k \in nband}\alpha(k)} \tag{3.6}$$

a(k) is the amplitude in the frequency band k, "nband" represents the number of the frequency bands in which the freqeuncy spectrum has been divided and K is the number of the particular band under analysis.

Spectral Roll Off

The spectral roll-off point is the frequency so that 95 % of the signal energy is contained below that frequency [103]. The formula defining the spectral roll-off is:

$$\sum_{0}^{f} c\alpha^2\left(f\right) = 0.95\sum_{0}^{s} r/2\alpha^2\left(f\right) \tag{3.7}$$

Spectral Slope

The spectral slope shows the amount of decreasing of the spectral amplitude and its computed by linear regression as: $a(f) = slope * f + const$ [103]. The slope in an example signal is shown in fig. 3.2.

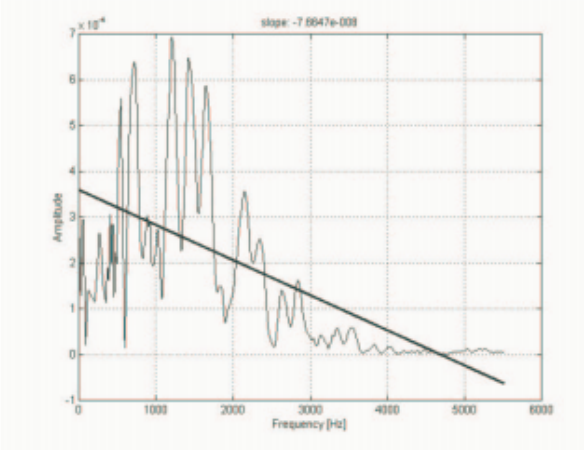

Figure 3.2: *Spectral slope in an example signal.*

Spectral Decrease

This spectral is also a representation of the decreasing of the spectral amplitude, but this formulation comes from perceptual studies and it's more correlated to human perception [103]. The formula is:

$$decrease = \frac{1}{\sum_{k=2}^{K} \alpha} \sum_{k=2}^{K} \frac{a(K) - a(1)}{k-1} \tag{3.8}$$

where a(K) is the amplitude of the K frequency band under analysis.

Spectral Centroid

The spectral centroid is a measure used in digital signal processing to characterise a spectrum. It indicates where the center of mass of the spectrum is located. Perceptually, it has a robust connection with the impression of brightness of a sound [107]. It is calculated as the weighted mean of the frequencies present in the signal, determined using a Fourier transform:

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \tag{3.9}$$

where x(n) represents the weighted frequency value, or the frequencies magnitude, of bin number n, and f(n) represents the center frequency of that bin [103].

Mel Spectrogram

In order to obtain the Mel spectrogram and the Mel Frequency Cepstral Coefficients (MFCC) there are some steps that have to be done. Firstly, it is necessary to divide the audio in smaller windows called short-time frames. This is done since frequency varies during time and because, after a Fourier transform is applied, the time information would be lost without the frames. The second step is windowing used to counteract the assumption made by the Fast Fourier Transform that the data is infinite and to reduce spectral leakage. Usually an hamming window is applied, in fig. 3.3. At this point, a periodogram is created, which is an estimate of the spectral density of a signal, by making a Discrete Fourier Transform or NN-point FFT on each frame to calculate

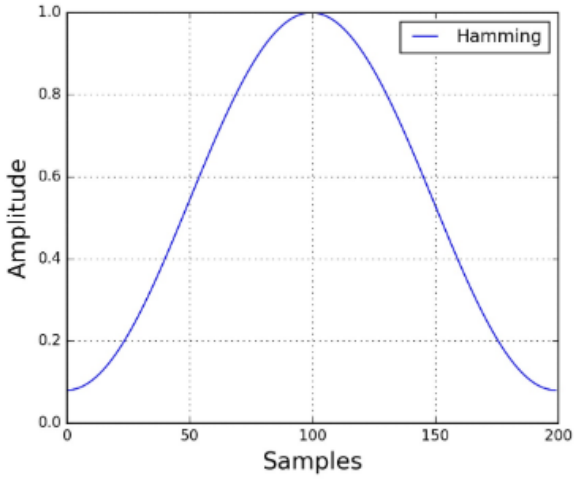

Figure 3.3: *Hamming Window*

the frequency spectrum. This operation is also called Short-Time Fourier-Transform (STFT), where NN is typically 256 or 512. Then the power spectrum is computed. Finally, a Mel spaced Filter Banks, which is a set of 20-40 triangular filters, is applied. Each vector is mainly composed by zeros, but has non-zero elements for a certain section of the spectrum. In order to calculate filterbank energies, each filterbank is multiplied with the power spectrum and then the coefficients are added up. Once this is performed, 20-40 numbers that give an indication of how much energy was in each filterbank are available [108]. A representation of the filter banks is shown in fig. 3.4. To get the filterbanks, lower and upper frequency are chosen and converted to Mels as

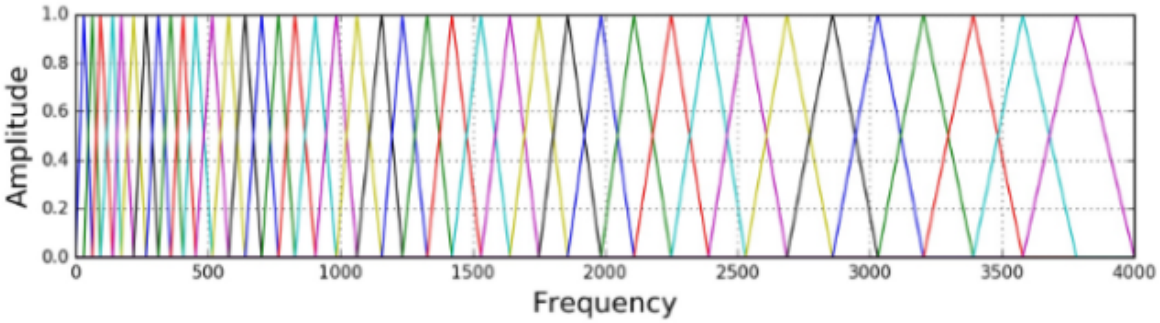

Figure 3.4: *Filter Bank on a Mel Scale*

follows:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{3.10}$$

Good values are 300 Hz for the lower and 8000 Hz for the upper frequency. After applying the Filter Banks, the Mel spectrogram is obtained [109]. In fig. 3.5, a mel spectrogram example is shown.

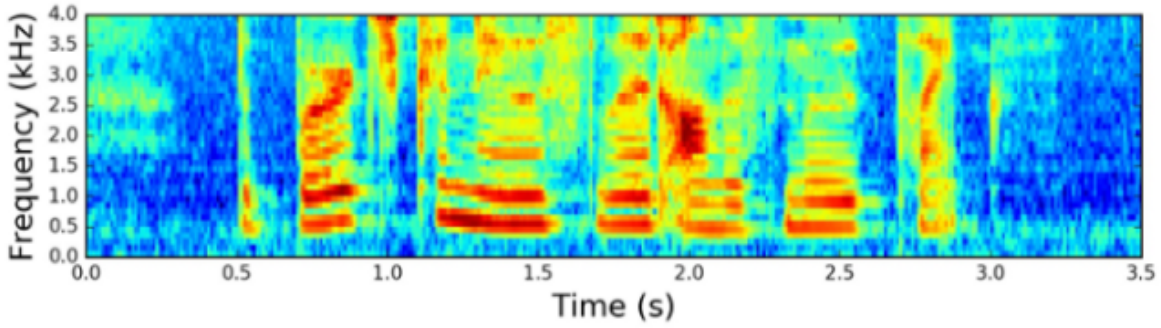

Figure 3.5: *Mel Spectrogram of the signal*

Mel Frequency Cepstral Coefficients

The issues with the Mel spectrogram is that the Filter bank coefficients are highly correlated. Therefore, it is convenient to decorrelate them. For this reason, the logs of the powers at each of the Mel frequencies are taken and a DCT (Discrete cosine transform) is applied. In this way, we obtain the cepstral coefficients, which are the amplitudes of the resulting spectrum [110].

3.2.2 Statistical Analysis

Normality tests are used to determine if a data set can be modeled by a normal distribution and to compute how likely it is for a random variable underlying the data set to be normally distributed (within some tolerance). To assess if the datasets used in this work could be approximated by a normal distribution, the Lilliefors test was used. The Lilliefors test is used to test the null hypothesis that data come from a normally distributed population, when the null hypothesis does not specify the expected value and variance of the distribution [111]. The Lilliefors test statistic formula is:

$$D^* = max \left| \widehat{F}(x) - G(x) \right| \tag{3.11}$$

where $\widehat{F}(x)$ is the empirical cumulative distribution function (CDF) of the sample data and G(x) is the CDF of the hypothesized distribution with estimated parameters equal to the sample parameters.

In order to evaluate whether two normal distributions are significantly different, the t-test can be used. The t-test is a statistical test of parametric type which is used to verify whether the mean value of a distribution differs significantly from a certain reference value [112]. In this test, the variance is unknown and, therefore, it is estimated and used in the following statistic test:

$$Y_0 = \frac{\overline{X} - \mu_0}{s/\sqrt{n}} \tag{3.12}$$

In the case of not normal distribution, in order to assess statistical significance between groups, it is necessary to use a non-parametric method to verify the equality of the medians. The method used was the Wilcoxon rank sum test, which is a non parametric test for equality of population mean ranks of two dependent samples X and Y [113].

3.2.3 Features Selection

After features extraction was performed, a statistical study of the predictors was made to assess if there were significant differences and independence among them. Then, keeping in mind the features significance results, features were selected in order to decrease the dimensionality of the problem. A type of feature selection is the filter type feature selection algorithm, which measures the features' importance based on the characteristics of the features, such as feature variance and feature relevance to the response. Filter type feature selection is uncorrelated to the training algorithm, because it is as part of the data preprocessing step carried out before training a model. Features can be selected in many different ways. One possibility is to select features that possess a stronger correlation to the classification variable. This method is called maximum-relevance selection. Many heuristic algorithms can be used, such as the sequential forward, backward, or floating selections. On the other hand, features can

be selected to be mutually far away from each other while still having "high" correlation to the classification variable. This scheme, termed as Minimum Redundancy Maximum Relevance (MRMR) selection has been found to be more powerful than the maximum relevance selection [114]. For this reason the MRMR was the type of feature selection chosen. The MRMR algorithm [115] finds an optimal set of features that is mutually and maximally dissimilar and can represent the response variable effectively. The algorithm minimizes the redundancy of a feature set and maximizes the relevance of a feature set to the response variable. The algorithm quantifies the redundancy and the relevance using the mutual information of variables—pairwise, mutual information of features and mutual information of a feature and the response [116]. The mutual information between two variables measures how much uncertainty of one variable can be reduced by knowing the other variable. The goal of the MRMR algorithm is to find an optimal set S of features that maximizes VS, the relevance of S with respect to a response variable y, and minimizes WS, the redundancy of S, where VS and WS are defined with mutual information I:

$$V_s = \frac{1}{|S|} \sum_{x \in S} I\left(x, y\right) \tag{3.13}$$

$$W_s = \frac{1}{|S|^2} \sum_{x,z \in S} I\left(x, z\right) \tag{3.14}$$

Where $|S|$ is the number of features in S. Finding an optimal set S requires considering all $2^{|\Omega|}$ combinations, where Ω is the entire feature set. Instead, the MRMR algorithm ranks features through the forward addition scheme, which requires $o(|\Omega| \cdot |S|)$ computations, by using the mutual information quotient (MIQ) value. Then, the MRMR has been applied to our features for both the two datasets. The 15 more important for the first dataset and the 15 more important for the second were selected and used to train the classifiers.

3.3 Classification algorithms

Classification is a type of supervised learning, that uses a prediction label of the training data. It specifies the class to which data elements belong to. In this ork, Matlab and its Deep Learning Toolbox have been used to implement, train and test all the algorithms. In detail, a binary classification was performed to divide the audios belonging to the positive class, "cough", and the audios belonging to the negative one, "no cough". Each classifier was trained with four different datasets:

- one containing all the 56 features extracted and containing both cough, speech sounds and noises
- one containing all the 56 features extracted and containing only cough and speech sounds
- one containing only the 15 features selected by MRMR, and containing both cough, speech sounds and noises
- one containing only the 15 features selected by MRMR and containing only cough and speech sounds

The features selection by MRMR, was explained in subsection 3.2.3. The classifiers that were chosen to be used were:

- Random Forests
- K Nearest Neighbours
- Support Vector Machine
- Long Term Short Memory Neural Network

The first three methods are classical machine learning techniques, while the last one is a deep learning technique which exploits a multi-layers neural network. In the next Subsections the basic principles behind each one of the classifiers and the parameters' choices adopted are explained.

3.3.1 Random Forests

Decision tree is a tree based algorithm used to solve regression and classification problems. They have the form of an inverted tree, starting from a single root and subsequently dividing the data in subgroups called leaves. At every node, a split is made choosing the predictors and their value according to an Entropy/Information gain criteria. Random forest is an ensemble learning method that constructs multiple decision trees during the training and chooses as class output the most present in the individual trees, using, therefore, a majority voting criteria. A simplified representation of random forest is visible in figure 3.6. RFs are a way of averaging multiple deep trees which avoids the problem of the high variance that single deep trees typically have [82].

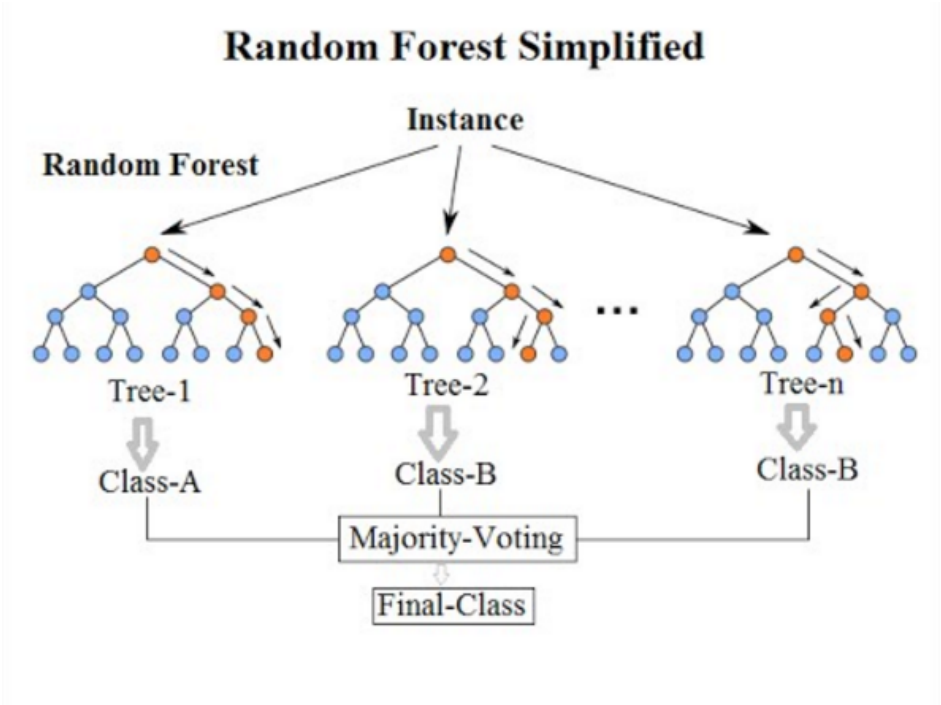

Figure 3.6: *Random Forest Diagram [117].*

The cost of using random forests an increase in the bias, which is an error from erroneous assumptions in the learning algorithm, and some loss of interpretability. However, this approach normally greatly boosts the final model performance. The training algorithm for RFs applies the bootstrap aggregating technique, or bagging technique, to tree learners. The bagging repeatedly selects a random sample from the original

training set, to create new training sets and uses those to fit the trees, as shown in figure 3.7 [84]. This procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. So, while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, if they are uncorrelated. The two hyperparameters tuned for this classifier were

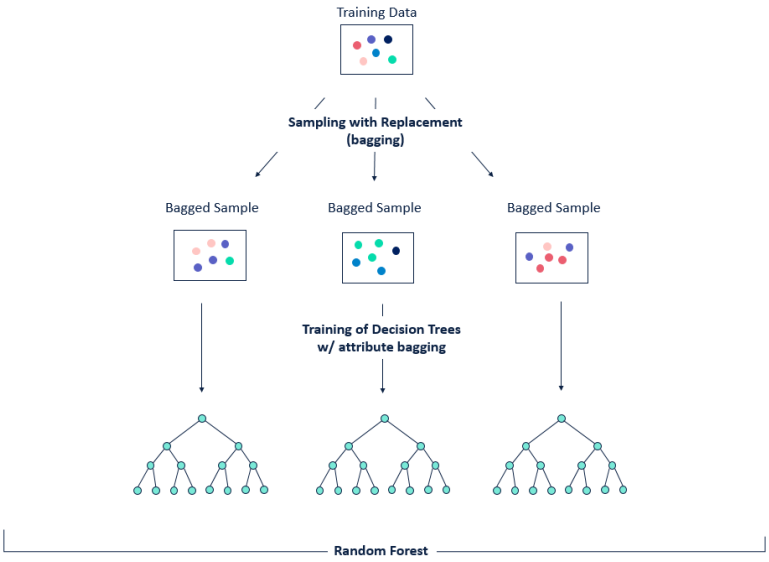

Figure 3.7: *Bootstrap Aggregating [118].*

the number of bagged trees from 5 to 50 and the minimum number of observations per tree leaf with values 5, 10, 20, 50, 100. Obtaining that better performance was when the bagged trees were 50 and a value of 5 for the minimum observations for leaf was selected.

3.3.2 K-Nearest Neighbors

K-nearest neighbors is a non-parametric method used for classification and regression. It does not construct an internal model, but saves the training data instances. The KNN explores the k nearest neighbours for each point, assumes the test datapoint to be similar to the training ones and derives the output. For a classification problem, a majority voting of the k nearest point is made. The training examples are vectors in a multidimensional feature space, each one having a class label. The algorithm does not have a proper training phase, since it only stores the feature vectors and class labels of the training samples. In the classification phase, k is a constant, and an unlabeled

vector is classified by assigning the label which is the most frequent among the k training samples nearest to the test point. A trivial example of how KNN works is shown in figure 3.8. KNN represents an easy and simple machine learning model with few hyperparameters to tune. Some disadvantages concerning the use of KNN are that the number of neighbors has to be selected previously and, if the sample size is large, the computational cost during runtime could be high [81]. The distance function used could be Euclidean distance, Manhattan distance, Hamming Distance, Minkowski distance. For this work, the learner's number of neighbors k was tuned in the range from 1 to 100 and the Euclidean distance was used.

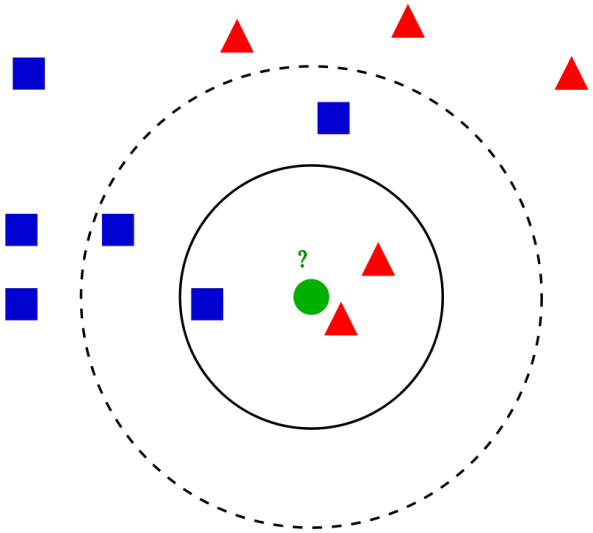

Figure 3.8: *KNN example [119]. The green circle represents the new data. If k=3, the neighbours are the ones inside the plain line and the label for the new data will be "red triangle". If k=5, the label assignment will be done considering the points inside the broken line area, resulting in "blue rectangle" classification.*

3.3.3 Support Vector Machines

The SVMs training alghoritm creates a model that assigns new samples to a class, acting like a binary linear classifier. The model is a representation of the samples in a space and it tries to divide the samples of opposite classes with a gap as wide as possible. Then, if a new sample falls in the positive class, it is classified as positive and vice versa. In more detail, the SVMs build a hyperplane or a set of hyperplanes in N-dimensional space (N, the number of features) that distinctly classifies the data

points. The hyperplane that best divides the data is the one that has the largest

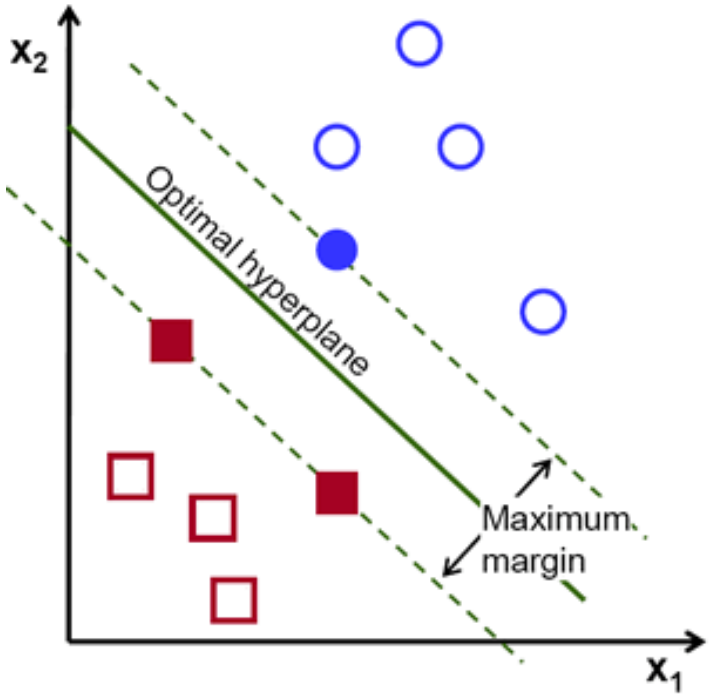

Figure 3.9: *SVMs. Support vectors defining the hyperplane margins*

distance between the nearest training-data point of each class, the margins are visible in figure 3.9. These points are the support vectors that define the separation margins and the objective is to maximize the margins distance, because it corresponds to the lowest generalization error of the classifier. It often happens that the sets to classify are not linearly separable in the finite space. For this reason, it could be necessary to map the original finite-dimensional space into a much higher-dimensional one, where the separation could be doable. The mappings used by SVMs schemes ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function k(x; y) selected according to the specific problem. In case of not linearly separable data, a penalty parameter (C) is introduced that determines the trade-off between increasing the margin size and ensuring that the new sample lies on the correct side of the margin. A regularization parameter is added to balance the margin maximization with the error loss. After adding the regularization parameter, the cost functions becomes:

$$min_w \lambda \|w\| d^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle) \tag{3.15}$$

In this formula, the first factor represents the opposite of margin distance, that has to be minimized, and the second factor represents the regularization part, where a penalty cost is given for every Support vector inside the margins or that was misclassified, so that the SVM has soft margins [79]. Example of kernel functions are linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. In this work, the SVMs used the RBF Kernel which has localized and finite response along the entire x-axis. The RBF Kernel is a general-purpose kernel, used when there is no prior knowledge about the data. The equation describing it is:

$$k\left(x_i, x_j\right) = exp\left(-\gamma \left\|x_i - x_j\right\|^2\right) \tag{3.16}$$

The SVMs classifiers were all tuned with Bayesian optimization and a 5-fold cross validation on Matlab. The bayesian optimization works classifying the locations of points from a Gaussian mixture model. This optimization automatically sets the penalty and the kernel-coefficient hyperparameters.

3.3.4 Neural Network LSTM

A neural network (or artificial neural network) is an adaptive system that learns by using interconnected nodes or neurons in a layered structure that resembles a human brain. A neural network can learn from data, so it can be trained to recognize patterns, classify data, and predict future events. A neural network breaks down the input into abstracted layers. It consists of an input layer, one or more hidden layers, and an output layer. In each layer there are several nodes, or neurons, with each layer using the output of the previous layer as its input, so neurons interconnect the different layers, shown in figure 3.10. It can be trained using many examples to recognize patterns in speech or images, for example, just as the human brain does. Its behavior is defined by the way its individual elements are connected and by the strength, or weights, of those connections. These weights are automatically adjusted during training according to a specified learning rule until the artificial neural network performs the desired task correctly. Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks. The term “deep”

usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150. Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction [120].

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN)

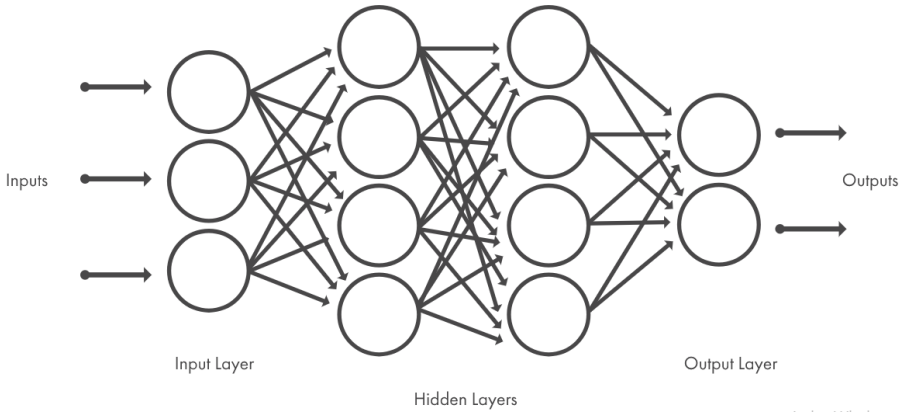

Figure 3.10: *Artificial Neural Networks [120]. Neural networks, organized in layers consisting of a set of interconnected nodes. It's possible to see the input layer, the hidden layers and the output layer.*

architecture used in deep learning [88]. Differently from standard feedforward neural networks, LSTM has feedback connections. Hence, it can process more than single data points, like images, but also entire sequences of data, like speech or video. LSTM networks are well-suited for classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. This is possible thanks to the LSTM cell unit that has the ability to encapsulate forgetting part of its previously stored memory while adding part of the new information. When training an RNN using back-propagation, the gradients which are back-propagated can "vanish", meaning that they can tend to zero, or "explode", meaning that they tend to infinity. The reason behind this behaviour is the computations involved finite-precision numbers. RNNs using LSTM units solve the vanishing gradient problem but still suffer from the exploding gradient problem [121]. A common architecture is composed of a cell, the memory part of the LSTM unit, and three "regulators", usually called gates, of the

flow of information inside the LSTM unit: an input gate, an output gate and a forget gate. The structure is represented in figure 3.11.

The cell is responsible for keeping track of the dependencies between the elements in

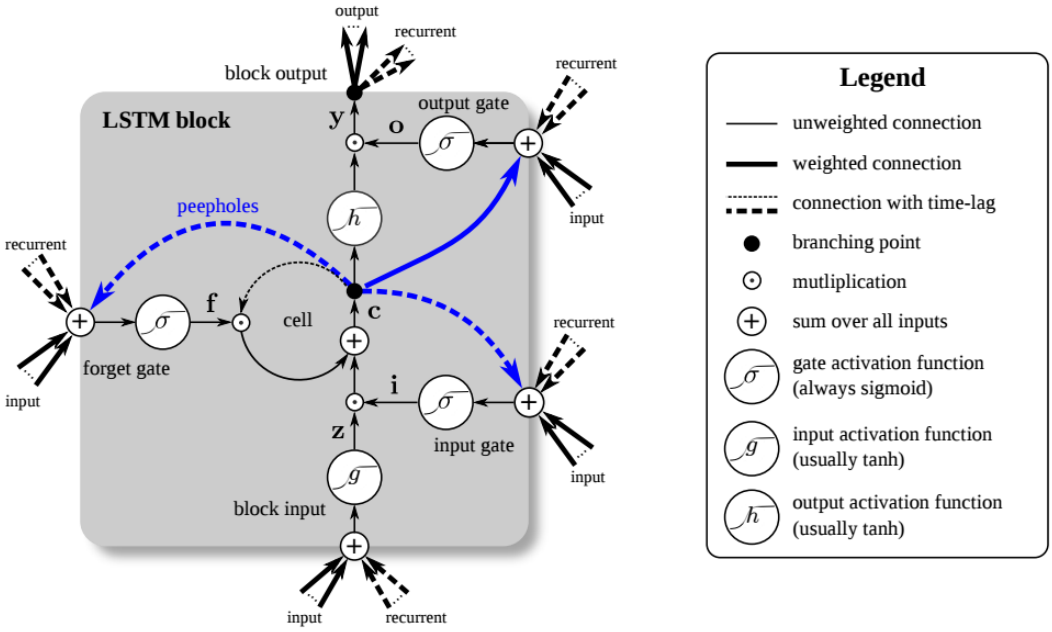

Figure 3.11: *LSTM cell unit. Image by Klaus Greff [122]. The LSTM unit has four input weights (from the data to the input and three gates) and four recurrent weights (from the output to the input and the three gates). Peepholes are extra connections between the memory cell and the gates, but they do not increase the performance by much and are often omitted for simplicity.*

the input sequence. The input gate controls the measure to which a new value flows into the cell, the forget gate controls which values remain in the cell and the output gate controls the values the cell uses to compute the output activation of the LSTM unit. The activation function of the LSTM gates is often the logistic sigmoid function. There are connections into and out of the LSTM gates, a few of which are recurrent. The weights of these connections, which need to be learned during training, determine how the gates operate. The neural network was built in Matlab, with a sequence input layer with a number of neurons equal to the dataset features, 56 and 15. A second layer works as a bilateral LSTM with 100 neurons. This cell unit works in both forward and backward direction, processing all the sentence and trying to adjust the coefficients in order to detect the training keyword. The third layer is a fully connected layer that classifies in the two labels. This layer combines all of the features and local information learned by the previous layers across the image or speech to identify the

patterns. For the classification problem, the last fully connected layer combines the features to classify the images [123]. This is the reason why the output size argument of the last fully connected layer of the network is equal to the number of classes of the data set. In this work, this dimension is therefore set to two, respectively for the "cough" and "no cough" classes. The fourth layer is a softmax layer that takes as input a vector of real numbers, that could be negative, or greater than one. After applying softmax, each component will be in the interval 0 - 1 and the components will add up to 1, so that they can be interpreted as probabilities [124]. Finally, the last layer is the output that actually classifies the labels "cough" and no cough". The training process uses an Adam method. The Adam optimization algorithm is an extension of the stochastic gradient descent vastly used in deep learning applications in computer vision and natural language processing. It is used to iteratively update the network weights during the training. It works by adapting the parameter learning rates based on the average first moment, the mean, and also the average of the second moments of the gradients, the uncentered variance [125]. The training is repeated for 10 epochs, where one epoch consists into one forward pass and one backward pass of all the training examples, and 4 iterations for each epoch, where the number of iterations is the number of passes, each pass using batch size number of examples. The batch size was set to 128.

3.3.5 Cross Validation

At last, all the classifiers were trained with a 10 K fold cross validation, implemented in Matlab, on both the dataset created, as explained in subsection 3.1.2. Cross-validation is a technique for assessing the capability of a statistic model to generalize on an new dataset. Its goal is to test the model's ability to perform prediction using data that was not used for training it, in order to avoid problems like overfitting or selection bias [126].

In a prediction problem, a model is trained on a dataset of known data, the training dataset, and is tested on a dataset of unknown data, called validation set or testing set [127]. One round of cross-validation involves partitioning a sample of data into two complementary subsets, respectively the training set and the validation set. Then, the

performance analysis on the training set is performed and the performance analysis on the validation set is used to validate the model. In the performance analysis the training error and the validation error, respectively, are calculated. At this point, multiple rounds of cross-validation are performed using different partitions, and the validation results are combined over the rounds to give a better estimate of the model's predictive performance, aiming to see if the variance is changed. There are different ways to implement this model. One technique is the Leave-p-out cross-validation (LpO CV), which consists in choosing a number p of observations to use for the validation set and using the remaining observations for the training. The particular case with p=1 is the Leave-one-out cross-validation (LOOCV). In the holdout method, data points are randomly assigned to two sets, the training set and the test set. The size of each of the sets is arbitrary, although typically the test set is smaller than the training set.The model is built on the first set and evaluated on the second. The holdout method involves a single run. So, it is the simplest way to perform cross validation [128], but the indicator of predictive accuracy is unstable and not robust, since it is not being smoothed out by multiple iterations. To avoid the holdout method weakness the k-fold cross-validation could be used. In K fold method the original sample is randomly partitioned into k equal sized subsamples. A single subsample is taken as the validation data for testing the model, and the remaining k - 1 subsamples are used to train the model. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as validation data. The k results can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. 5 or 10 fold cross validation are commonly used [129], but in general k remains an unfixed parameter. When k = n (the number of observations), k fold cross validation is equivalent to leave-one-out cross validation. In stratified k fold cross-validation, the partitions are selected so that the mean response value is approximately equal in all the partitions. In the case of binary classification, this means that each partition contains roughly the same proportions of the two types of class labels. All four classifiers were applied to the datasets with a 10-fold cross validation, keeping around 560 audio in the training set and 62 in the test

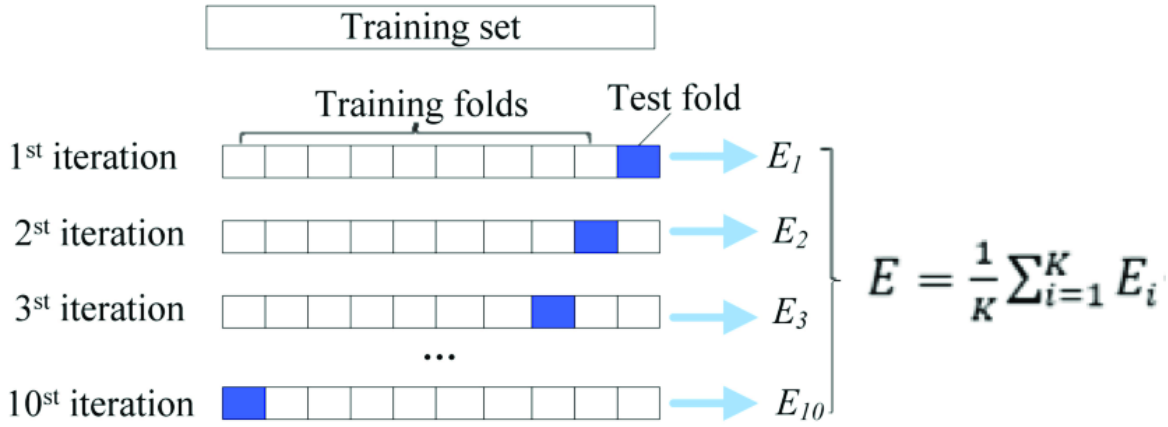

Figure 3.12: *10-fold cross validation [130].*

set. For the classical methods the classification has been made for each frame and not on the entire audio.

3.4 Algorithms performance metrics

In order to evaluate the algorithms performances, a statistical evaluation of the results is required. Typical metrics in two classes classification algorithms are accuracy, precision, recall, F1 score and ROC curve. Then, to assess if the feature selection actually improved the method performance, a Wilcoxon Rank Sum Test was used. In statistics and machine learning, a confusion matrix, or error matrix, [131] is a table with specific form used to represent the predicted data respect the real data. It is usually used for supervised learning, specially in classification. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa) [132]. In figure 3.13 a confusion matrix for binary classification is shown. The table reports the number of false positives, false negatives, true positives, and true negatives.

- False positives (FP) are the values which are predicted as positive, but whose true class is negative.
- False negatives (FN) are the values which are predicted as negative, but whose true class is positive.
- True positives (TP) are the correct predictions for the positive class.

Confusion Matrix		Predicted Class	
		1	0
True Class	1	True Positives (TP)	False Negatives (FN)
	0	False Positives (FP)	True Negatives (TN)

Figure 3.13: *Confusion Matrix for binary classification*

- True negatives (TN) are the correct predictions for the negative class.

With the confusion matrix it is possible to have a more complete description of the performance than just with the proportion of correct classifications, called accuracy. Accuracy counts the correct predictions comparing them with the real labelled values. It is the closeness of the measurements to a specific value. It can be calculated as true positive plus true negative divided by the total number of values:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.17}$$

If the dataset is unbalanced, accuracy can give a misleading information. For this reason, other metrics, easily computed from the confusion matrix, can be used.
Precision, or positive predictive value, is the closeness of the measurements to each other, and it is obtained by dividing the true positives with true positives plus false positives:

$$Precision = \frac{TP}{TP + FP} \tag{3.18}$$

Sensitivity, or recall, measures the proportion of positives that are correctly identified, so it is called true positive rate. It is calculated as true positives divided by true positives plus false negatives, which are the totality of the true positives values (P) [133]:

$$Recall = \frac{TP}{TP + FN} \tag{3.19}$$

Specificity is the true negative rate because shows how many times the negative class has been correctly identified. It is calculated as true negatives divided by true negatives plus false positives:

$$Specificity = \frac{TN}{TN + FP} \tag{3.20}$$

Finally, the F1 score is the harmonic mean of the precision and recall [134]. It is defined as:

$$F_1 = 2 \cdot \frac{precision \cdot sensitivity}{precision + sensitivity} = \frac{2TP}{2TP + FP + FN} \tag{3.21}$$

Sensitivity and specificity can be used in a graphical representations called ROC curve. ROC stands for Receiver Operating Characteristic. It is represented by putting on the X axis the sensitivity and on the Y axis 1-specificity, represented respectively by the True Positive Rate (TPR) and the False Positive Rate (FPR). A ROC curve example is shown in figure 3.14.

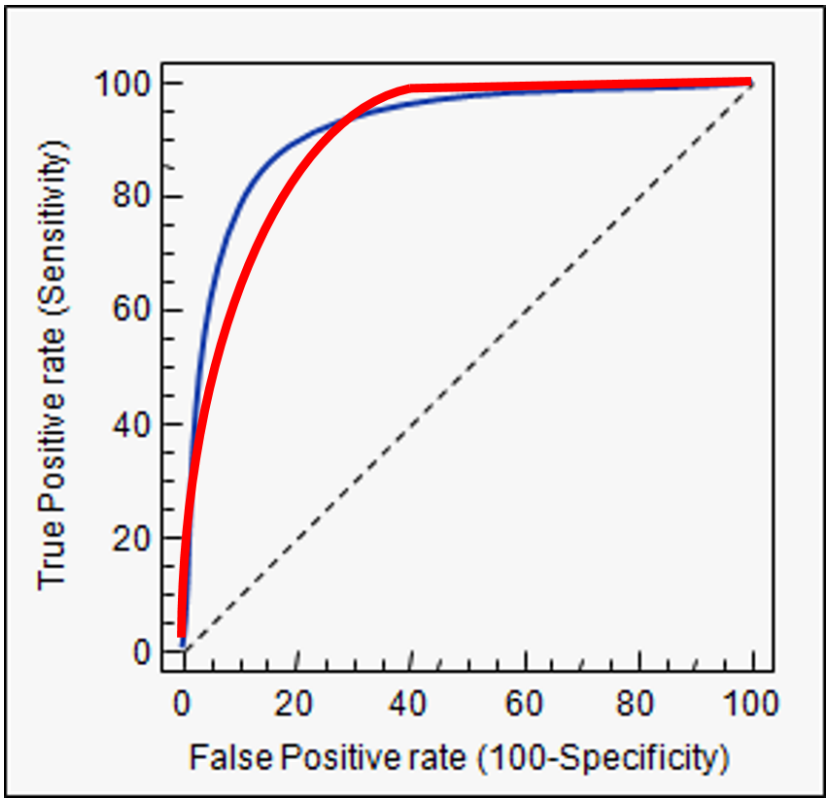

Figure 3.14: *ROC curve example [135].*

The ROC curves analysis of the classifier capacity of evaluating between the positive and the negative class is assessed using the Area Under the Curve AUC. This value

ranges between 0 and 1 and is equivalent to the probability that the classification performance applied to a random sample from the negative class is higher than the performance obtained randomly from the positive class. The diagonal line that goes from (0,0) to (1,1) represents the random classifier (line of no-discrimination) and it has AUC equals to 0.5 [136]. As shown in figure 3.14. These metrics were used to compare the classifiers performances. In particular, accuracy, precision and recall were calculated and compared. The ROC curves were plotted and used to compare the four algorithms visually, for both the datasets and when using all the features extracted or just the 15 selected by MRMR.

Finally, the Kruskal-Wallis test was used to evaluate whether the classifiers performances were statistically different, before comparing them one versus one. This test verifies if the groups come from the same population [137]. It performs an analysis of variance replacing the data with their rank [138].

The other comparisons to assess differences in the performance metrics were made by means of the Wilcoxon rank sum test, which has been already explained in subsection 3.2.2. The Wilcoxon test was used to compare the performances across the two datasets, so if the classifiers trained with the "Complete" dataset performed better than when the training was done with the "Speech & Cough" dataset. Then, the Wilcoxon test was used to compare if there was a significant difference in the performance across the number of features used. In other words, it was evaluated if the classifiers predicted better when all the features extracted were utilized or when only the 15 selected by MRMR were used in the training.

Chapter 4

Results and discussion

4.1 Extracted voice features

For all the features in both datasets and for both classes (cough versus no cough) median and interquartile range were computed. A normality test, specifically the Lilliefors test, was applied on the two datasets', "Complete" and "Speech & Cough", features, giving as results that, in both the populations the majority of the features are not normally distributed. The only normal distribution is the feature "MFCC8" of the class "cough". Therefore, for the sake of simplicity, and since in the class "no cough" there are no normal distributions at all, it was considered not normal as well. For this reason a T test could not be applied to distinguish whether the features were significantly different between the positive class, "cough", and the negative class, "no-cough". To assess this difference, a Wilcoxon rank sum test was used, since it is suitable to be applied to not normal distributions. The results obtained showed that, in the "Complete" dataset, all the features, except three, were found significantly different for the two populations. These three features are "MFCC13", "spectralKurtosis" and "spectralRolloffPoint". The p-values obtained were respectively 0.955, 0.608 and 0.160. These three features, along with other features reported by way of example, are represented in figure 4.1, where the boxes represent the median value and the red line represents the interquartile range. The light blue colored boxes represent the features that are not significantly different between classes. The same predictors extracted from

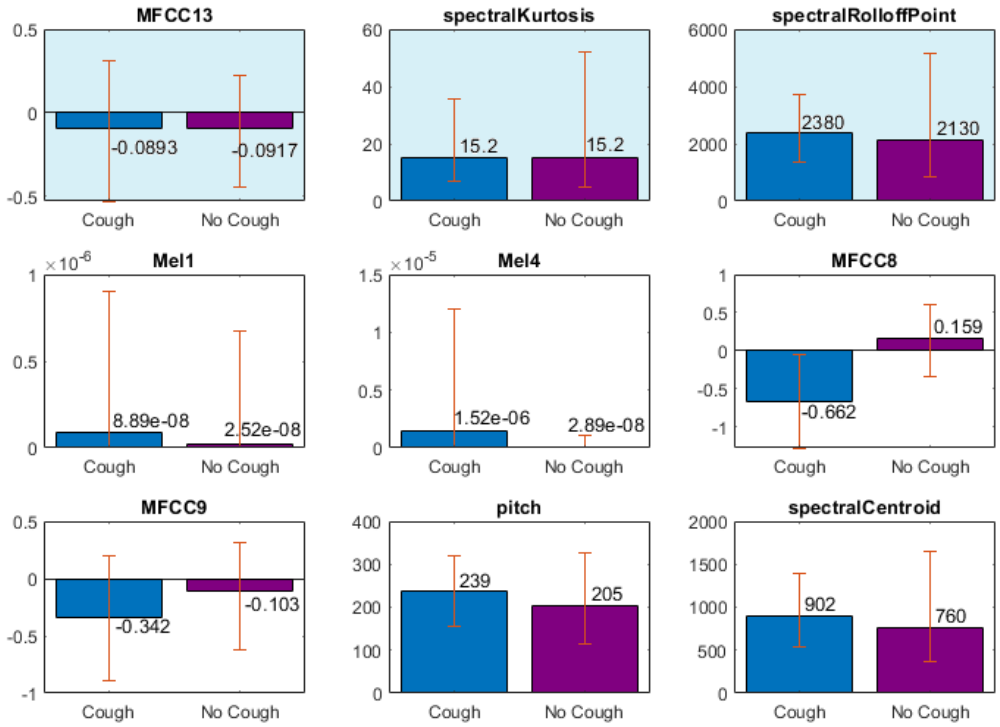

Figure 4.1: *Median and interquartile range for the extracted features. "Complete" Dataset. The light blue colored boxes represent the features that are not significantly different between classes.*

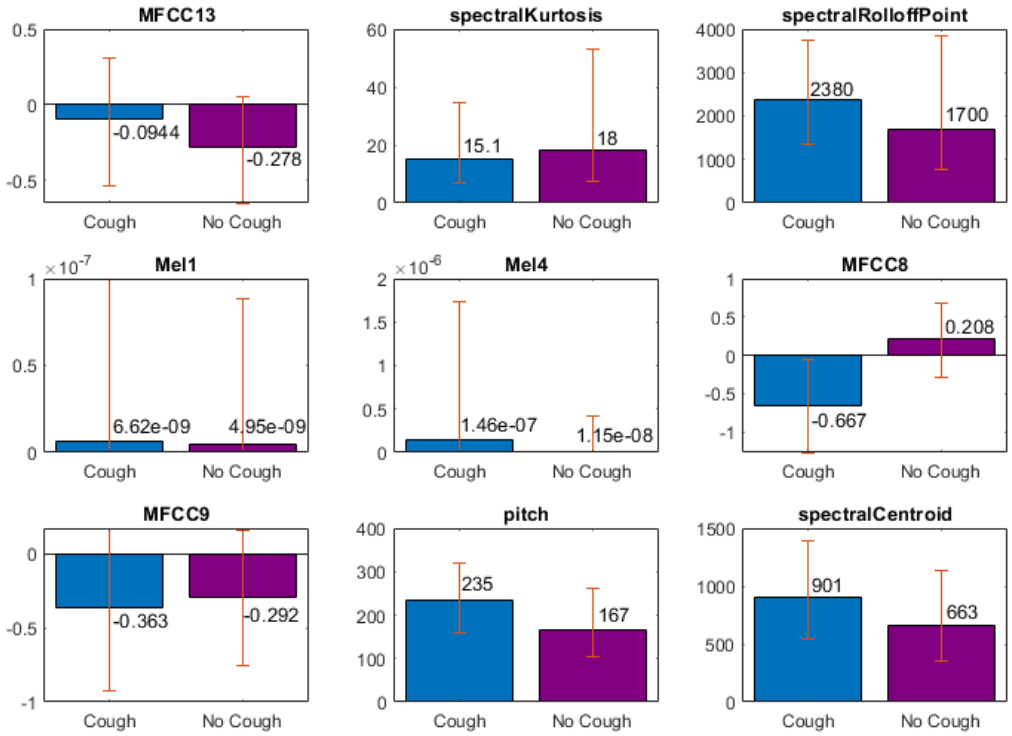

Figure 4.2: *Median and interquartile range for the extracted features. "Speech & Cough" Dataset. All the features are significantly different.*

the "Speech & Cough" dataset are shown in figure 4.2. In this case, the features resulted all to be significantly different. Other than these nine features, taken as example, the other features selected by the MRMR are represented into appendix A.

4.2 Selection of voice features

After assessing that the features showed significant difference between "cough" and "no cough" population, an automatic feature selector was applied. The algorithm selecting the features follows a Minimum Redundancy Maximum Relevance (MRMR) method, as explained in subsection 3.2.3. The MRMR results for the "Complete" dataset are shown in figure 4.3, which displays the best 15 features. The predictors shown in the figure are ordered according to the predictor importance score, and the three best predictors for this dataset are "Mel4", "MFCC9" and "spectralKurtosis". In figure

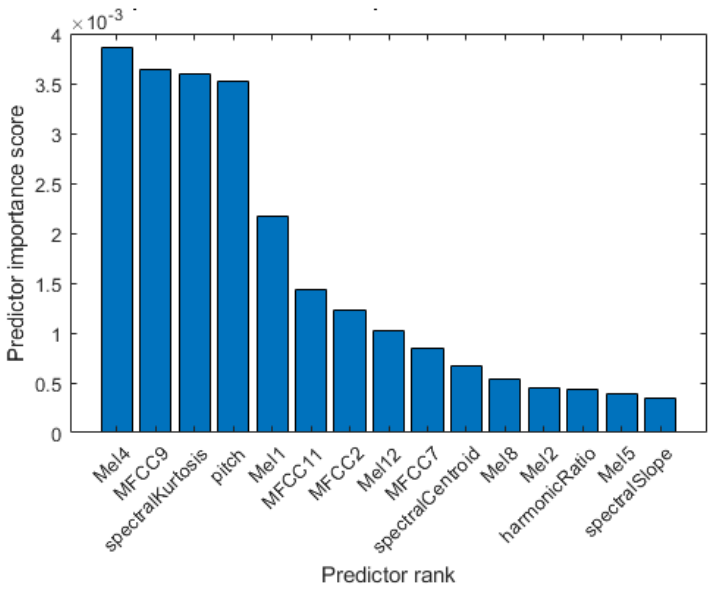

Figure 4.3: *MRMR from "Complete" Dataset. Best 15 predictors obtained applying MRMR.*

4.4, the MRMR values for the "Speech & Cough" dataset are shown. In this case, the predictor importance scores are bigger with respect to the "Complete" dataset of two orders of magnitude. The best features, according to the selector, are "Mel32", "MFCC12" and "MFCC8".

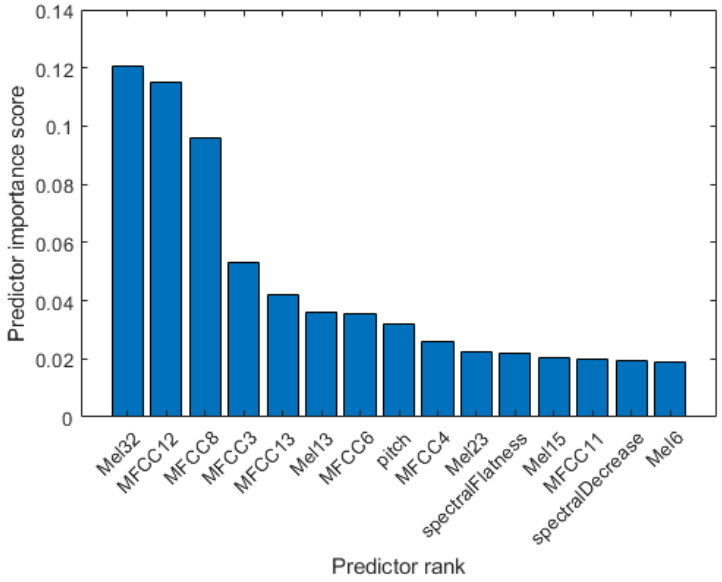

Figure 4.4: *MRMR from "Speech & Cough" Dataset. Best 15 predictors obtained applying MRMR.*

4.3 Classification Results

The classifiers performances results are summarized in table 4.1 and table 4.2. The performance metrics used were accuracy, precision, recall. The statistical significance of the performance results were assessed respectively between the two datasets, between the two cases in which all the features were used and the case in which the fifteen selected features were used, and across the different classifiers compared one by one. All the performance metrics were computed as the median, with the corresponding

Table 4.1: *Comparison between the different classification algorithms when all 56 features are used respect to when only 15 selected features are used, in terms of accuracy, precision and recall. Classifiers were trained with the "Complete" dataset.*

	Accuracy			**Precision**			**Recall**		
	56 Features	15 Features	**p-value**	56 Features	15 Features	**p-value**	56 Features	15 Features	**p-value**
RF	88.2 ±1.7%	86.1 ±4.8%	0.089	76.3 ±9%	75.1 ±4.8%	0.307	82.5 ±8.5%	78.7 ±9.3%	0.241
KNN	85.8 ±3%	76.5 ±2.3%	**0.001***	82.6 ±6.2%	66.3 ±2.3%	**0.001***	75.5 ±10.5%	60.9 ±10.5%	**0.002***
SVM	87.6 ±1.3%	87.6 ±1.2%	0.678	78.5 ±14.3%	78.7 ±11.6%	0.734	80.5 ±14.3%	78.5 ±4.7%	0.545
LSTM	94.4 ±2.8%	91.1 ±6.5%	0.124	96.8 ±2.4	93.5 ±6.4%	0.163	92.4 ±3.3%	90.6 ±8.5%	0.239

$p < 0.05$ for Wilcoxon Test

interquartile range, of the single values obtained during the 10-fold cross validation. In all the results tables, that follows, the combination that were significantly different are highlighted in bold and with an asterisk, and their p-value, which is lower than 0.05, is shown. Starting from the "Complete" dataset, the classifiers general performance is

Table 4.2: *Comparison between the different classification algorithms when all 56 features are used respect to when only 15 selected features are used, in terms of accuracy, precision and recall. Classifiers were trained with the "Speech & Cough" dataset*

	Accuracy			**Precision**			**Recall**		
	56 Features	15 Features	**p-value**	56 Features	15 Features	**p-value**	56 Features	15 Features	**p-value**
RF	90.2 ±2.2%	89.8 ±3.9%	0.571	73.1 ±13.8	74.6 ±15.2	0.970	86.6 ±5.1%	85.3 ±6.7%	0.571
KNN	79.8 ±3.3	75.6 ±2.5%	**0.011***	69.5 ±18.8%	61.3 ±16.2	0.121	63.9 ±9%	57 ±5.8%	0.212
SVM	85.2 ±2.9%	84.8 ±2.8%	0.678	76.5 ±4.6%	75.9 ±5.1%	0.734	65.6 ±17.7	65.4 ±18.5%	0.545
LSTM	95.9 ±2.8%	94.4 ±5.4%	**0.001***	95.2 ±8.2%	93.2 ±8.9%	0.469	98.2 ±8.1%	93.3 ±5.1	0.354

$p < 0.05$ for Wilcoxon Test

good, ranging from a median of the accuracy of 94.4 ± 2.8 % for the Long Short Term Memory Network to a 85.8 ± 3% for the KNN, when all the features are used. While 15 features are selected, the best performance is obtained by LSMT with 91.1 ± 6.5% and the worst by KNN with 76.5 ± 2.3% Due to the fact that the algorithms were trained and tested by frames, the originally balanced dataset became unbalanced towards the negative "nocough" class. Hence, precision and recall are more robust indicator of the performance. The KNN performance in the case when only 15 features were used is significantly different from the one with all features, as shown in the table 4.1 in the p-value column. p-values were obtained using the Wilcoxon rank sum test.

For what regards the "Speech & Cough" dataset, the classifiers' performance ranges from the larger 95.9 ± 2.8% accuracy for the LSTM to the lower 79.8 ± 3.3% for the KNN, when all 56 features are used. If the classifiers utilize only the 15 selected features the performance goes from an accuracy of 94.4 ± 5.4% to one of 75.6 ± 2.5%. Then, only the accuracy metric of KNN and LSTM shows a statistical difference when using all the features or just the selected ones, table 4.2. Generally, the selection of the features did not bring any performance improvement, if not a decreasing capacity of prediction in some cases.

The four tables, table 4.3, table 4.4, table 4.5 and table 4.6 compare the four algorithms: RF, KNN, SVM and LSTM. The comparison is made for each dataset, "Complete" and "Speech", and for the two cases using all the 56 features and the 15 selected features. From table 4.3, table 4.4, table 4.5 and table 4.6, it's possible to see if there is significant statistical difference between the four classifiers. The p values from the Kruskal Wallis test made on all the algorithms together were lower then 0.001 for all cases. So, for the sake of simplicity, has been omitted from the tables. As it is possible to observe, for all

Table 4.3: *Comparison between the different classification algorithms in terms of accuracy, precision and recall. Classifiers were trained on the "Complete" dataset, considering all 56 features*

Metrics	Classifiers				P values					
	RF	**KNN**	**SVM**	**LSTM**	**RF-KNN**	**RF-SVM**	**RF-LSTM**	**KNN-SVM**	**KNN-LSTM**	**SVM-LSTM**
Accuracy	88.2 ±1.7%	86.1 ±3%	87.6 ±1.3%	94.4 ±2.8%	0.151	0.915	0.068	0.465	**0.001***	**0.010***
Precision	76.3 ±9%	82.6 ±6.2	78.5 ±14.3%	96.8 ±2.4%	0.752	0.992	**0.001***	0.894	**0.014***	**0.001***
Recall	82.5 ±8.5%	75.5 ±10.5%	80.5 ±14.3%	92.4 ±3.3%	0.152	0.984	**0.035***	0.302	**0.001***	**0.012***

$p < 0.05$ for Wilcoxon Test

Table 4.4: *Comparison between the different classification algorithms in terms of accuracy, precision and recall. Classifiers were trained on the "Complete" dataset, considering 15 selected features*

Metrics	Classifiers				P values					
	RF	**KNN**	**SVM**	**LSTM**	**RF-KNN**	**RF-SVM**	**RF-LSTM**	**KNN-SVM**	**KNN-LSTM**	**SVM-LSTM**
Accuracy	86.1 ±4.8%	76.5 ±2.3%	87.8 ±1.2%	91.1 ±6.5%	**0,035***	0.886	0.059	**0.003***	**0.001***	0.283
Precision	75.1 ±2.3%	66.3 ±2.3%	78.7 ±11.6%	93.5 ±6.4%	0.489	0.436	**0.001***	**0.018***	**0.001***	0.134
Recall	78.7 ±9.3%	60.9 ±10.5%	78.5 ±4.7%	90.6 ±8.5%	**0.032***	0.996	**0.039**	**0.016***	**0.001***	0.072

$p < 0.05$ for Wilcoxon Test

the datasets, the classifiers present performance differences. So, a further analysis has been made to evaluate whether every single method classifies differently from another, by comparing them one versus one. Considering the case where all the features are used, from the table 4.3 , dataset "Complete", is possible to see that LSTM performs better than the three other algorithms. In table 4.5, dataset "Speech & Cough" the p-values indicate that LSTM is clearly better than SVM and KNN, while only the precision is better than the RF's one. Then, RF has a better recall than KNN. In the case of the classifiers when only the 15 selected features are used, table 4.4 and table 4.6, show that KNN performs poorly with respect to all the other methods. LSTM performs better than KNN, of course, and also than RF. The LSTM performance metrics outperformed other studies results, such as Liu et al. [9] and Tracey et al. [6] Lucio et al. [10] which achieved, with different methods, respectively the 91%, 81% and 87% of sensitivity in detecting cough events. And also Pramono et al [1], Larson et al. [11] and Breiman et al. [12] that all obtained a sensitivity equal to 92%.

In order to assess whether the classifiers perform better on a simpler dataset with just cough and speech, like "Speech & Cough", with respect to a more complete one with cough, speech and ambient noises, like "Complete", a Kruskal Wallis test was applied. The test compares every algorithm for the "all features" case and for the "selected features" one. The comparison was made for each algorithm where the training was made with the first dataset and when has been made with the second. In table 4.7 the

Table 4.5: *Comparison between the different classification algorithms in terms of accuracy, precision and recall. Classifiers were trained on the "Speech & Cough" dataset, considering all 56 features*

Metrics	Classifiers				P values					
	RF	**KNN**	**SVM**	**LSTM**	**RF-KNN**	**RF- SVM**	**RF-LSTM**	**KNN-SVM**	**KNN-LSTM**	**SVM-LSTM**
Accuracy	90.2 ±2.2%	79.8 ±3.3%	85.2 ±2.9%	95.9 ±2.8%	0.075	0.660	0.206	0.587	**0.001***	**0.001***
Precision	73.1 ±13.8%	69.5 ±18.8%	76.5% ±4.6%	95.2 ±8.2%	0.795	0.992	**0.001***	0.922	**0.014***	**0.002***
Recall	86.6 ±5.1%	63.9 ±9%	85.6 ±17.7%	98.2 ±8.11%	**0.019***	0.526	0.302	0.408	**0.001***	**0.010'***

$p < 0.05$ for Wilcoxon Test

Table 4.6: *Comparison between the different classification algorithms in terms of accuracy, precision and recall. Classifiers were trained on the "Speech & Cough" dataset, considering all 15 features*

Metrics	Classifiers				P values					
	RF	**KNN**	**SVM**	**LSTM**	**RF-KNN**	**RF- SVM**	**RF-LSTM**	**KNN-SVM**	**KNN-LSTM**	**SVM-LSTM**
Accuracy	89.9 ±3.9%	75.6 ±2.5%	84.8 ±2.8%	94.4 ±5.4%	**0.002***	0.894	0.550	**0.020***	**0.001***	0.178
Precision	74.6 ±15.2%	61.3 ±16.2%	75.9 ±5.1%	93.2 ±8.9%	0.199	0.790	**0.008***	**0.019***	**0,001***	0.108
Recall	85.3 ±6.7	57 ±5.8%	65.4 ±18.5%	93.3 ±5.1%	**0.002***	0.618	0.408	0.083	**0.001***	**0.029***

$p < 0.05$ for Wilcoxon Test

p-values point out that in KNN case with all features, the dataset has an impact on the performance. In fact the precision went from 81.5% with "Complete' dataset against a 72.2% with "Speech & Cough". The SVM has only the accuracy different when 15 features are used, as shown in table 4.8 so it has better accuracy when "Complete" dataset is used. The LSTM has a significantly higher precision and recall when the "Complete" dataset is used with respect to "Speech & Cough" one, when all features are utilized.

Table 4.7: *Comparison between the different datasets "Complete" and "Speech & Cough" in terms of accuracy, precision and recall. The classifiers were trained with all the 56 features*

	Accuracy			**Precision**			**Recall**		
	Complete	Speech & Cough	p value	Complete	Speech & Cough	p value	Complete	Speech & Cough	p value
RF	88.2 ±1.7%	90.2 ±2.2%	0.162	76.3 ±9%	73 ±13.8%	0.734	82.5 ±8.5%	86.6 ±5.1%	0.162
KNN	86.1 ±3%	79.8 ±3.3%	**0.001***	82.6 ±6.2%	69.5 ±18.8%	0.089	75.5 ±10.5%	63.9 ±9%	**0.021***
SVM	87.6 ±1.3%	85.2 ±2.9%	0.054	78.5 ±14.3%	76.5 ±4.6%	0.623	80.5 ±14.3%	65.6 ±17.7%	0.140
LSTM	94.4 ±2.8%	95.9 ±2.8%	0.179	96.8 ±2.4%	95.2 ±8.2%	**0.001***	92.4 ±3.3%	98.2 ±8.1%	**0.001***

$p < 0.05$ for Wilcoxon Test

Table 4.8: *Comparison between the different datasets "Complete" and "Speech & Cough" in terms of accuracy, precision and recall. The classifiers were trained with 15 features selected*

	Accuracy			**Precision**			**Recall**		
	Complete	Speech & Cough	p value	Complete	Speech & Cough	p value	Complete	Speech & Cough	p value
RF	86.1 ±4.8%	89.8 ±3.9%	0.121	75.1 ±4.8%	74.6 ±15.2%	0.385	78.7 ±9.3%	85.3 ±6.7%	0.064
KNN	76.5 ±2.3%	75.6 ±2.5%	0.162	66.3 ±2.3%	61.3 ±16.2%	0.678	60.9 ±10.5%	57 ±5.8%	0.427
SVM	87.8 ±1.2%	84.8 ±2.8%	**0.021***	78.7 ±11.6%	75.9 ±5.1%	0.427	78.5 ±4.7%	65.4 ±18.5%	0.186
LSTM	91.1 ±6.5%	94.4 ±5.4%	0.239	93.5 ±6.4%	93.2 ±8.9%	0.761	90.6 ±8.5%	93.3 ±5.1%	0.150

$p < 0.05$ for Wilcoxon Test

4.3.1 ROC Curves

A simple way to visualize the different classifiers performances is by using the ROC Curve. The ROC curve represents the "True positive rate" against the "False positive rate". The area under each curve, AUC, is computed and can be used as a performance

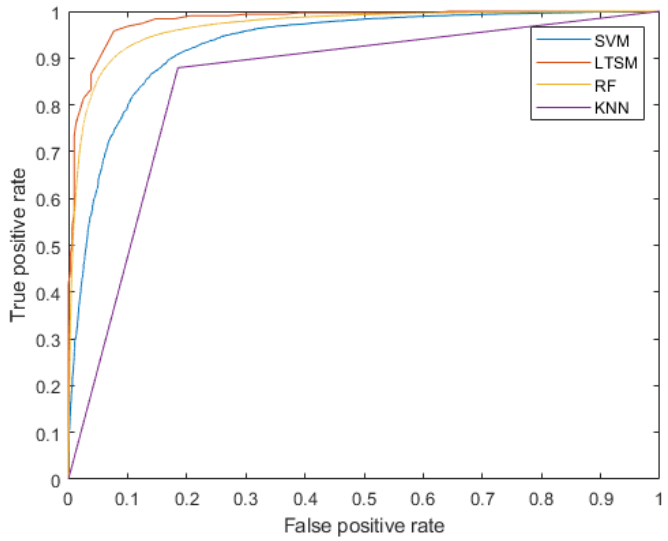

Figure 4.5: *ROC when classification algorithms are trained with "Complete" dataset and all 56 Features*

metric as well. The ROC Curves shown in figure 4.5 represent the results from "Complete" dataset with all features used for the four different classification algorithms. As it is possible to observe, the LTSM and RF have a greater area under the curve with respect to SVM and KNN, showing a better general performance.
The ROC Curves shown in figure 4.6 represent the results from "Complete" dataset with the selected 15 features used. The LTSM, RF and SVM have a greater area under the curve with respect to KNN, showing a better general performance. The ROC Curves in figure 4.7 represent the results from "Speech & Cough" dataset when all the features are used. The classifiers show different curves, the best one is the LSMT's one, while RF ans SVM perform similarly and KNN presents the curve with the smaller area, meaning the worst performance. The ROC Curves shown in figure 4.8 represent the results from "Speech & Cough" dataset with the selected 15 features used. The LTSM, RF and SVM have a greater area under the curve respect to KNN, showing a better general performance.

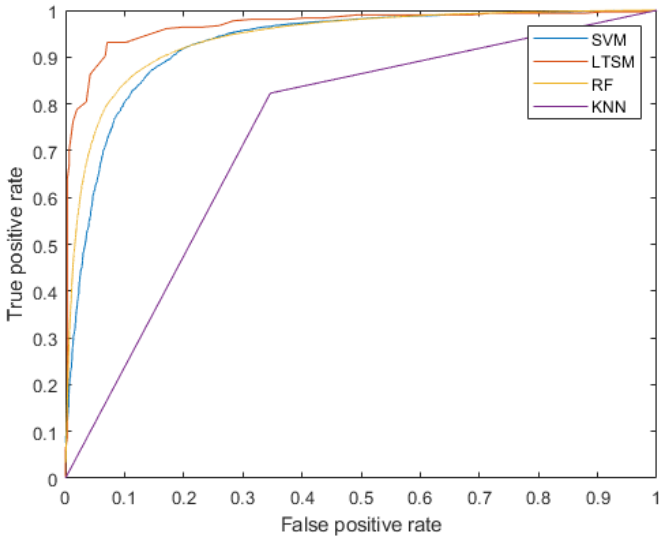

Figure 4.6: *ROC when classification algorithms are trained with "Complete" dataset and selected 15 Features*

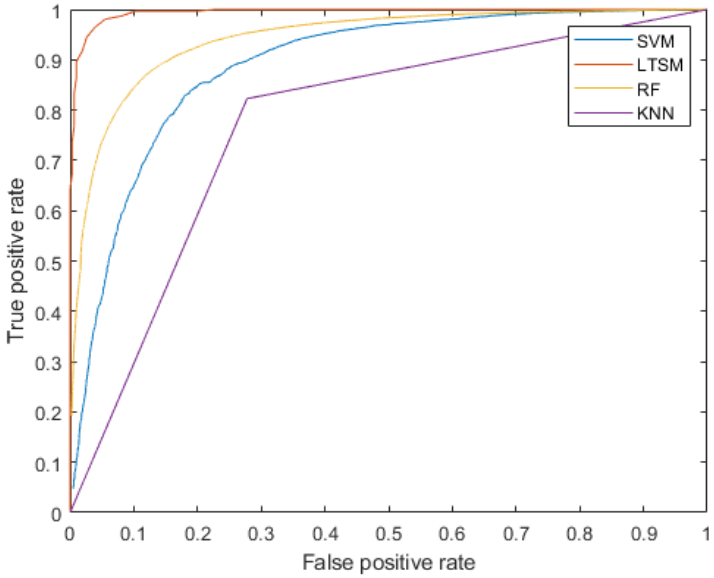

Figure 4.7: *ROC when classification algorithms are trained with "Speech & Cough" dataset and all 56 Features*

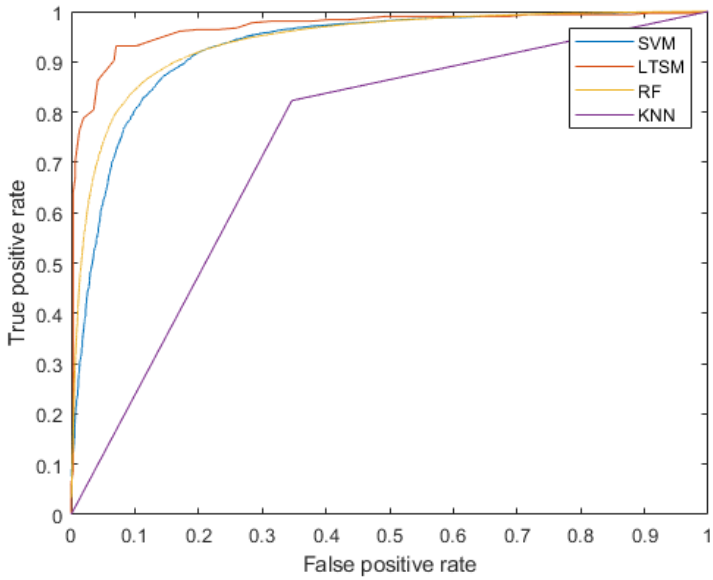

Figure 4.8: *ROC when classification algorithms are trained with "Speech & Cough" dataset and selected 15 Features*

Chapter 5

Conclusions

5.1 Conclusions

This thesis work uses machine learning techniques typically used in speech recognition to identify respiratory sounds and extract information from them. The aim of this work was to develop a classification model to assess patients' respiratory health and quality of life through the count of cough events. This thesis work is articulated across different fields of investigation. Initially, a deep study of the literature was performed in order to identify which of the many parameters that could be extracted from the audio were the more promising and which frame length was the more appropriate to detect cough events. After those elements were defined, features were extracted from the audio divided in overlapping frames. Then, through statistical analysis, the most significant features were chosen to perform classification on the collected data. Different classifiers, trained to detect cough events inside audios containing speech and environmental sounds, were compared and evaluated. The algorithms compared in this work were both classical machine learning methods, such as Random Forest with Bagging, Support Vector Machine, K-Nearest Neighbours and deep machine learning methods, Long Short-Term Memory. They were trained with two different datasets, one containing speech sounds and cough, the other containing also environmental noises, to investigate possible differences in performance when environmental noises were added. The classifier performances were obtained after a 10-fold cross validation. The obtained

performance metrics were analyzed and the overall classifiers analysis showed that there was difference between them. The best classifier resulted to be LSTM. Its performance metrics outperformed other studies results, such as Liu et al. [9] and Tracey et al. [6] Lucio et al. [10] which achieved, with different methods, respectively 91%, 81% and 87% sensitivity in detecting cough events. LSTM instead performed with 92.4 ± 3.3% sensitivity and 96.8 ± 2.4% precision, if trained with "Complete" dataset and all the features. Then, the metrics analysis showed that all the methods were robust to noise. This could be inferred from the fact that the algorithms' performances did not show significant difference when the two different datasets were used. Only for the KNN classifier, when using the "Complete" dataset, an increase in the classification ability was observed. LSTM, specifically, obtained p-values for both precision and sensitivity lower than 0.001, showing that this method is robust to noise. The work explored the statistical significance of the features selection, so if selecting 15 features instead of using all the 56 brought benefits to the classification. The result was that, in the algorithms used, it decreased the dimensionality of the problem but without improving or decreasing the performance. KNN was the the algorithm whose performance decreased with the features selection, showing that some important information for the classifier was lost. The other classifiers did not show difference when the features used were only 15. In particular, LSTM did not show a significant difference when using the features subset, obtaining p-values for precision and sensitivity equal to 0.163 and 0.239 respectively. In general, features selection resulted not to be a mandatory step, especially for the LSTM neural network. Anyway, because it did not decrease the performance, it could be applied to reduce the dimensionality of the problem. In conclusion, LSTM was shown to be significantly better than all the other methods, robust to noise and still performing after features selection. For these reasons, it is the indicated choice for future developments.

5.2 Limits and Future Developments

One of the limits of this work are the lack of different neural networks architectures and the comparison with the VGG (Visual Geometry Group) method, developed at

the University of Oxford. VGG nets are Convolutional neural networks developed to perform localization and classification tasks [139]. They can be used as a method to detect the cough events, along with the LSTM. Otherwise, they can be used as a comparison with the LSTM, due to the fact that VGG nets are already implemented and optimized for sounds recognition, cough included. Moreover, some LSTM parameters, such as hidden neurons number, could be tuned furthermore, and the most performing segment length for cough detection, which is the part of the audio that has to be considered as the unit to classify, should be assessed. A possible development of this work could be to evaluate and optimize the segment length, which is the part of the audio used to identify the single cough event and that the algorithm tries to classify. Another possibility could be assessing the minimum number of features the classifier could use while maintaining good performance. Finally, a possible application of this work could be using the best algorithm configuration to detect on-fly the cough events and to count them during the day. Within this framework, the development of a mobile app, which could be used on a day-by-day basis by a large amount of users, but specially designed for elderly, in a transparent and non-intrusive way, would be useful. Further improvements could consist in the creation of an User Interface and, in a more advanced future, the insertion of an emotion analyzer from speech into the model, that could be implemented with the same predictors, classifier and in a parallel algorithm.

Appendix A

Full Features Statistical Analysis

The following figures represent all the features median and interquartile range for each class. The predictors represented are the one chosen by MRMR, without the most important that instead are represented in section 4.1. In figure A.1 and figure A.2 the features from dataset "Complete" are represented. In figure A.3 and figure A.4 the features from dataset "Speech & Cough" are shown.

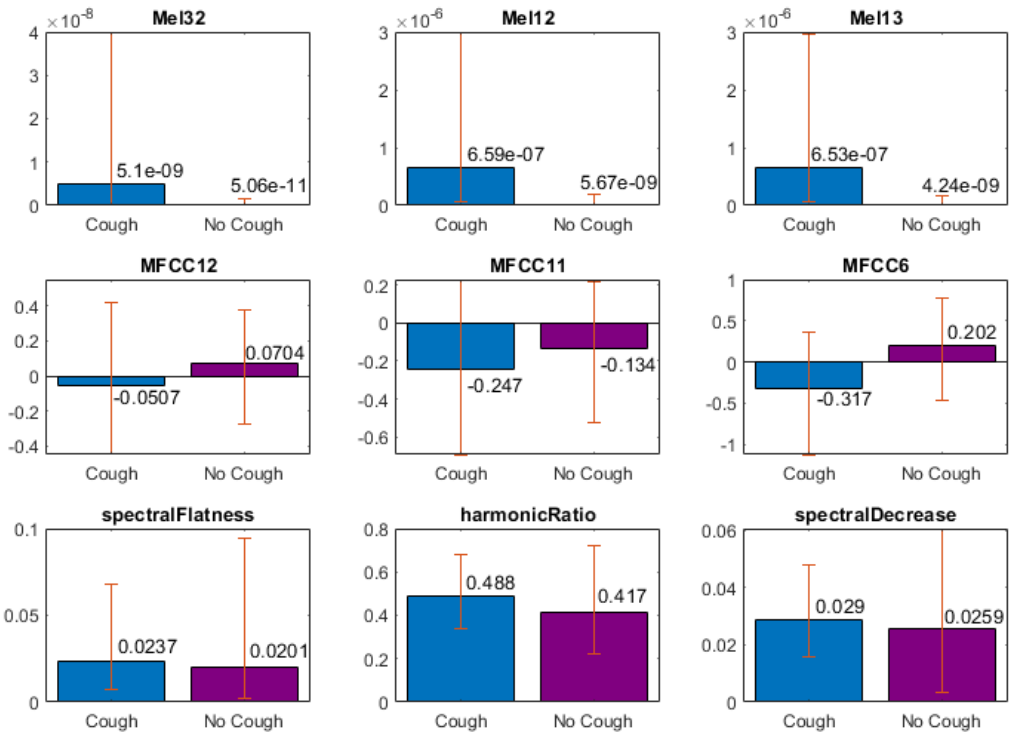

Figure A.1: *Median & IQR: "Complete", part 2.*

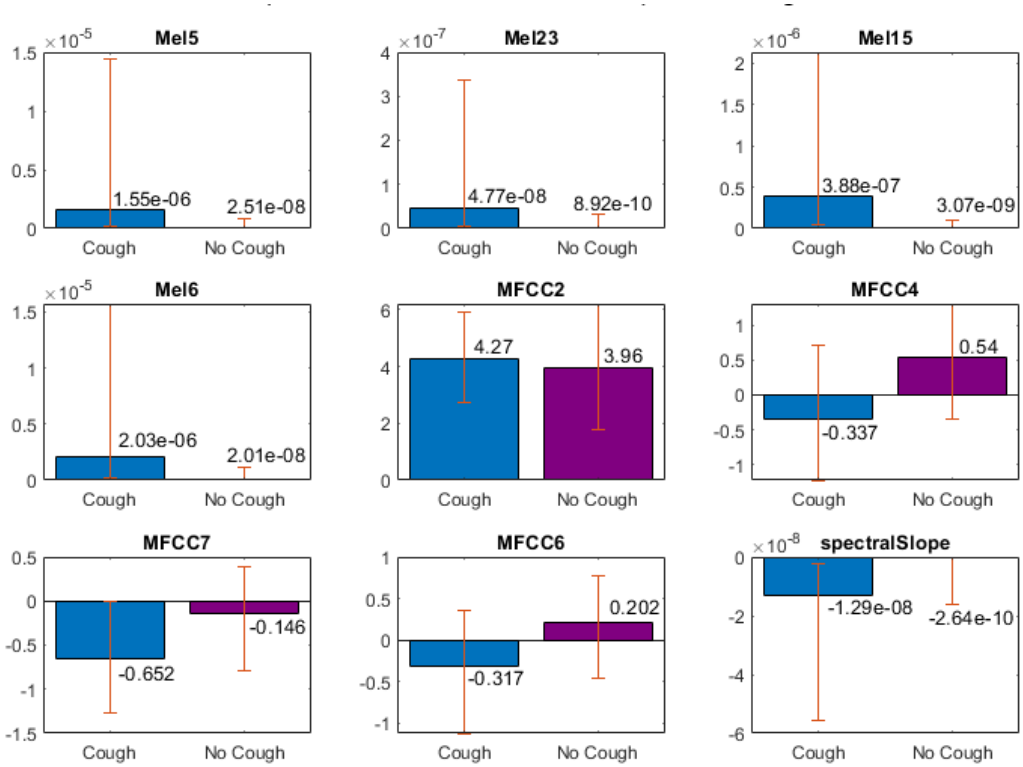

Figure A.2: *Median & IQR: "Complete", part 3.*

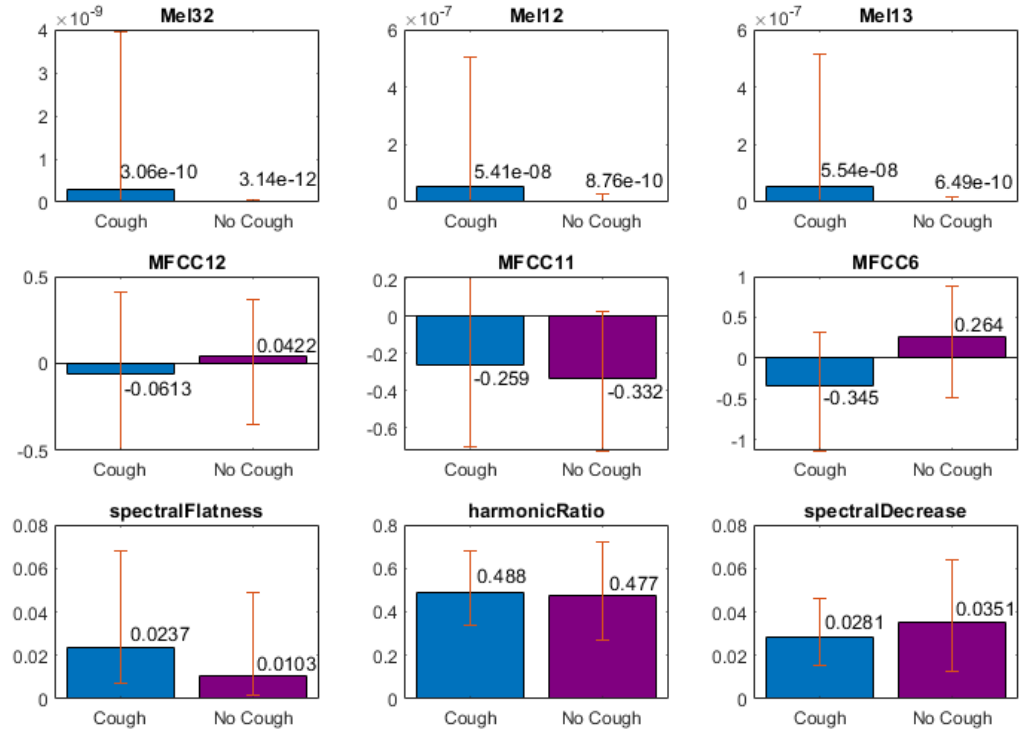

Figure A.3: *Median & IQR: "Speech & Cough", part 2.*

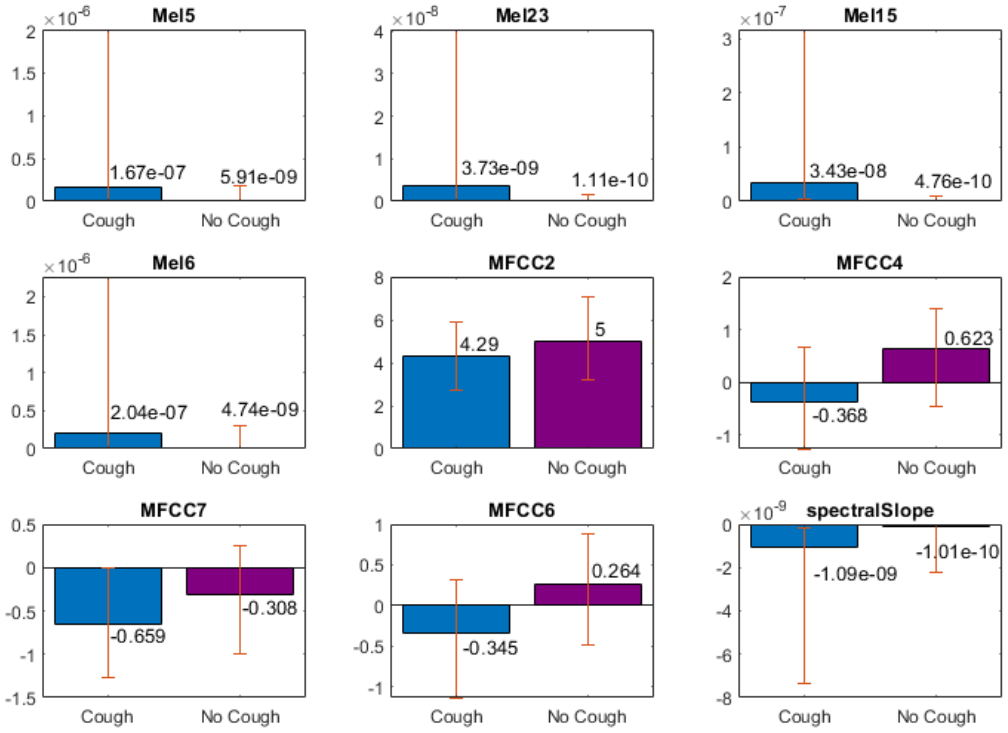

Figure A.4: *Median & IQR: "Speech & Cough", part 3.*

List of Figures

List of Tables

Bibliography

[1] R. Pramono, A. Imtiaz, and E. Rodriguez-Villegas, "A cough-based algorithm for automatic diagnosis of pertussis," *PloS one*, vol. 11, p. e0162128, 09 2016.

[2] I. Miranda, A. Diacon, and T. Niesler, "A comparative study of features for acoustic cough detection using deep architectures *," vol. 2019, pp. 2601–2605, 07 2019.

[3] F. Barata, K. Kipfer, M. Weber, P. Tinschert, E. Fleisch, and T. Kowatsch, "Towards device-agnostic mobile cough detection with convolutional neural networks," in *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1–11, 2019.

[4] L. Di Perna, G. Spina, S. Thackray-Nocera, M. G. Crooks, A. H. Morice, P. Soda, and A. C. den Brinker, "An automated and unobtrusive system for cough detection," 2017.

[5] V. Swarnkar, U. R. Abeyratne, Y. Amrulloh, C. Hukins, R. Triasih, and A. Setyati, "Neural network based algorithm for automatic identification of cough sounds," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1764–1767, 2013.

[6] B. H. Tracey, G. Comina, S. Larson, M. Bravard, J. W. López, and R. 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*, pp. 6017–6020, 2011.

[7] R. Pramono, A. Imtiaz, and E. Rodriguez-Villegas, "Automatic identification of cough events from acoustic signals," vol. 2019, pp. 217–220, 07 2019.

[8] C. Hoyos-Barceló, J. Monge-Álvarez, Z. Pervez, L. M. San-José-Revuelta, and P. Casaseca-de-la Higuera, "Efficient computation of image moments for robust cough detection using smartphones," pp. 176–185, 2018.

[9] J. Liu, M. You, G. Li, Z. Wang, X. Xu, Z. Qiu, and et al., "Cough signal recognition with gammatone cepstral coefficients," pp. 160–164, 2013.

[10] C. Lúcio, C. Teixeira, J. Henriques, P. de Carvalho, and R. P. Paiva, "Voluntary cough detection by internal sound analysis," 10 2014.

[11] E. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. Patel, "Accurate and privacy preserving cough sensing using a low-cost microphone," 2011.

[12] L. Breiman, "Random forests," p. 5–32, 2001.

[13] Y. Amrulloha, U. Abeyratnea, V. Swarnkar, R. Triasih, and A. Setyatib, "Automatic cough segmentation from non-contact sound recordings in pediatric wards," *Biomedical Signal Processing and Control*, vol. 21, pp. 126–136, 08 2015.

[14] B. Mohebali, A. Tahmassebi, A. Meyer-Baese, and A. H. Gandomi, "Chapter 14 - probabilistic neural networks: a brief overview of theory, implementation, and application," in *Handbook of Probabilistic Models* (P. Samui, D. Tien Bui, S. Chakraborty, and R. C. Deo, eds.), pp. 347 – 367, Butterworth-Heinemann, 2020.

[15] S. Matos, S. S. Birring, I. D. Pavord, and H. Evans, "Detection of cough signals in continuous audio recordings using hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1078–1083, 2006.

[16] P. Bageshree, D. Chinmayi, H. Harshada, and Z. Mrunal, "Stress detection from speech signal using mfcc, svm and machine learning techniques," *International Journal Of Latest Trends In Engineering and Technology*, vol. 17, pp. 41–44, 2020.

[17] T. John, "Europe's population is aging rapidly. here's how to turn that into an opportunity," 2019. `https://edition.cnn.com/2019/06/29/europe/europe-aging-population-int/index.html`.

[18] B. G. Celler, T. Hesketh, W. Earnshaw, and E. Ilsar, "An instrumentation system for the remote monitoring of changes in functional health status of the elderly at home,"

in *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, pp. 908–909 vol.2, 1994.

[19] S. Fox and M. Duggan, "Tracking for health," *Pew Research Center's Internet ∧ American Life Project*, 2013.

[20] W. H. Organization, "Global observatory for ehealth series - volume 3 mhealth: New horizons for health through mobile technologies," 2011.

[21] W. H. Organization, "Global diffusion of ehealth: Making universal health coverage achievable. report of the third global survey on ehealth," 2016.

[22] M. T. Nkosi, F. Mekuria, and S. H. Gejibo, "Challenges in mobile bio-sensor based mhealth development," in *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, pp. 21–27, 2011.

[23] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. MüLler, and S. Narayanan, "Paralinguistics in speech and language—state-of-the-art and the challenge," *Computer Speech &Language*, vol. 27, no. 1, pp. 4–39, 2013.

[24] A. Fedorova, O. Glembek, T. Kinnunen, and P. Matějka, "Exploring ann back-ends for i-vector based speaker age estimation," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[25] T. Ebihara, P. Gui, C. Ooyama, K. Kozaki, and S. Ebihara, "Cough reflex sensitivity and urge-to-cough deterioration in dementia with lewy bodies," *ERJ Open Research*, vol. 6, no. 1, 2020.

[26] K. Sekizawa, "Lack of cough reflex in aspiration pneumonia," *The Lancet*, vol. 335, pp. 1228–1229, 1990.

[27] J. Cacioppo, L. Tassinary, and G. Berntson, *Handbook of psychophysiology.* 01 2007.

[28] P. Rajasekaran, G. Doddington, and J. Picone, "Recognition of speech under stress and in noise," in *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, pp. 733–736, 1986.

[29] H. Kurniawan, A. V. Maslov, and M. Pechenizkiy, "Stress detection from speech and galvanic skin response signals," in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pp. 209–214, 2013.

[30] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "Rastaplp speech analysis," 1991.

[31] O. Simantiraki, G. Giannakakis, A. Pampouchidou, and M. Tsiknakis, "Stress detection from speech using spectral slope measurements," 11 2016.

[32] T. Nwe, H. Li, and M. Dong, "Analysis and detection of speech under sleep deprivation," in *INTERSPEECH*, 2006.

[33] H. Greeley, J. Berg, E. Friets, J. Wilson, G. Greenough, J. Picone, J. Whitmore, and T. Nesthus, "Fatigue estimation using voice analysis," *Behavior research methods*, vol. 39, pp. 610–9, 09 2007.

[34] T. Roehrs and T. Roth, "Multiple sleep latency test," *Journal of clinical neurophysiology : official publication of the American Electroencephalographic Society*, vol. 9, pp. 63–7, 02 1992.

[35] H. Greeley, E. Friets, J. Wilson, S. Raghavan, J. Picone, and J. Berg, "Detecting fatigue from voice using speech recognition," *International Symposium on Signal Processing and Information Technology*, vol. 0, pp. 567–571, 08 2006.

[36] B. Gunsel, C. Sezgin, and J. Krajewski, "Sleepiness detection from speech by perceptual features," 05 2013.

[37] T. J. Holwerda, D. J. Deeg, A. T. Beekman, T. G. van Tilburg, M. L. Stek, C. Jonker, and R. A. Schoevers, "Feelings of loneliness, but not social isolation, predict dementia onset: results from the amsterdam study of the elderly (amstel)," *Journal of Neurology, Neurosurgery ∧ Psychiatry*, vol. 85, no. 2, pp. 135–142, 2014.

[38] R. S. Wilson, K. R. Krueger, S. E. Arnold, J. A. Schneider, J. F. Kelly, L. L. Barnes, Y. Tang, and D. A. Bennett, "Loneliness and risk of alzheimer disease," *Archives of general psychiatry*, vol. 64, no. 2, pp. 234–240, 2007.

[39] M. Luchetti, J. H. Lee, D. Aschwanden, A. Sesker, J. E. Strickhouser, A. Terracciano, and A. R. Sutin, "The trajectory of loneliness in response to covid-19.," *American Psychologist*, 2020.

[40] A. M. Paredes, E. E. Lee, L. Chik, S. Gupta, B. W. Palmer, L. A. Palinkas, H.-C. Kim, and D. V. Jeste, "Qualitative study of loneliness in a senior housing community: the

importance of wisdom and other coping strategies," *Aging ∧ Mental Health*, pp. 1–8, 2020.

[41] V. D. Badal, S. A. Graham, C. A. Depp, and al., "Prediction of loneliness in older adults using natural language processing: Exploring sex differences in speech," *The American Journal of Geriatric Psychiatry*, 2020.

[42] D. Russell, "Ucla loneliness scale (version 3): Reliability, validity, and factor structure," *Journal of personality assessment*, vol. 66, pp. 20–40, 03 1996.

[43] "Watson natural language understanding - overview." IBM.

[44] R. Plutchik, *Emotions and life: Perspectives from psychology, biology, and evolution.* American Psychological Association, 2003.

[45] P. Gómez-Vilda, J. Mekyska, J. M. Ferrández, D. Palacios-Alonso, A. Gómez-Rodellar, Rodellar-Biarge, and et al., "Parkinson disease detection from speech articulation neuromechanics," *Frontiers in neuroinformatics*, vol. 11, 2017.

[46] C. Carmona-Duarte, R. Plamondon, P. Gomez, M. Ferrer, J. Alonso, and A. Londral, *Application of the Lognormal Model to the Vocal Tract Movement to Detect Neurological Diseases in Voice*, pp. 25–35. 06 2016.

[47] R. Petersen, G. Smith, S. Waring, R. Ivnik, E. Tangalos, and E. Kokmen, "Mild cognitive impairment: Clinical characterization and outcome," *Archives of neurology*, vol. 56, pp. 303–8, 04 1999.

[48] C. Matteo, *Automatic speech analysis for early detection of functional cognitive impairment in elderly population.* PhD thesis, 2019.

[49] B. Roark, M. Mitchell, J. Hosom, K. Hollingshead, and J. Kaye, "Spoken language derived measures for detecting mild cognitive impairment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2081–2090, 2011.

[50] N. Barroso, K. Lopez-de Ipiña, H. Eguiraun Martinez, J. Solé-Casals, M. Ecay, A. Ezeiza, P. Martinez-Lage, and U. Lizardui, "Alzheimer disease diagnosis based on automatic spontaneous speech analysis," 06 2013.

[51] M. F. Julian Appell, Andrew Kertesz, "A study of language functioning in alzheimer patients," vol. Volume 17, Issue 1, pp. 73–91, 1982.

[52] S. Yamanda, S. Ebihara, T. Ebihara, M. Yamasaki, T. Asamura, M. Asada, K. Une, and H. Arai, "Impaired urge-to-cough in elderly patients with aspiration pneumonia.," *Cough (London, England)*, vol. 4, p. 11, 12 2008.

[53] m. Cleveland Clinic medical professional, "Respiratory system." https://my.clevelandclinic.org/health/articles/21205-respiratory-system, 2020.

[54] U. Illustrator, "Respiratory system." https://visualsonline.cancer.gov/details.cfm?imageid=1773, 2001. National Cancer Institute.

[55] "Covid-19 and vascular disease," *EBioMedicine*, vol. 58, 2020.

[56] N. P. Tortora, Gerard J; Anagnostakos, *Principles of anatomy and physiology.* New York: New York : Harper Row, 1987.

[57] G. Anderson and K. D. Harrison, "Study of language." http://www.swarthmore.edu/SocSci/langhotspots/glossary.html, 2007.

[58] "Chapter 11 - language," in *Fundamentals of Cognitive Neuroscience* (B. J. Baars and N. M. Gage, eds.), pp. 313 – 328, San Diego: Academic Press, 2013.

[59] "Cough." https://www.findatopdoc.com/Medical-Library/Symptoms/Cough. Findatopdoc.com.

[60] P. Piirila and A. R. Sovijarvi, "Objective assessment of cough," 1995.

[61] A. A. Raj and S. S. Birring, "Clinical assessment of chronic cough severity," *Pulmonary Pharmacology & Therapeutics*, vol. 20, no. 4, pp. 334 – 337, 2007.

[62] J. Smith and A. Woodcock, "New developments in the objective assessment of cough," *SpringerLink*, vol. 186, pp. 48–54, 2008.

[63] J. Hsu, R. Stone, R. Logan-Sinclair, M. Worsdell, C. Busst, and K. Chung, "Coughing frequency in patients with persistent cough: assessment using a 24 hour ambulatory recorder," *European Respiratory Journal*, vol. 7, no. 7, pp. 1246–1253, 1994.

[64] S. S. Kraman, G. R. Wodicka, G. A. Pressler, and H. Pasterkamp, "Comparison of lung sound transducers using a bioacoustic transducer testing system," *Journal of Applied Physiology*, vol. 101, no. 2, pp. 469–476, 2006.

[65] H. A. Bickerman and S. E. Itkin, "The effect of a new bronchodilator aerosol on the air flow dynamics of the maximal voluntary cough of patients with bronchial asthma and pulmonary emphysema," *Journal of chronic diseases*, vol. 8, no. 5, pp. 629–636, 1958.

[66] V. Gross, C. Reinke, F. Dette, R. Koch, D. Vasilescu, T. Penzel, and U. Koehler, "Mobile nocturnal long-term monitoring of wheezing and cough," *Biomedical Engineering/Biomedizinische Technik*, vol. 52, no. 1, pp. 73–76, 2007.

[67] K. McGuinness, A. Kelsall, J. Lowe, A. Woodcock, and J. Smith, "Automated cough detection: a novel approach," *Am J Respir Crit Care Med*, vol. 175, p. A381, 2007.

[68] M. A. Coyle, D. B. Keenan, L. S. Henderson, M. L. Watkins, B. K. Haumann, D. W. Mayleben, and M. G. Wilson, "Evaluation of an ambulatory system for the quantification of cough frequency in patients with chronic obstructive pulmonary disease," *Cough*, vol. 1, no. 1, p. 3, 2005.

[69] S. Barry, A. Dane, A. Morice, and A. Walmsley, "The automatic recognition and counting of cough," *Cough (London, England)*, vol. 2, p. 8, 02 2006.

[70] E. Sejdic, I. Djurovic, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," *Digital Signal Processing*, vol. 19, pp. 153–183, 01 2009.

[71] Ming-Kuei Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[72] H. A. Bickerman and S. E. Itkin, "The effect of a new bronchodilator aerosol on the air flow dynamics of the maximal voluntary cough of patients with bronchial asthma and pulmonary emphysema," *Journal of chronic diseases*, no. no. 5, pp. 629–636, vol. 8, 1958.

[73] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *nature, vol. 521*, no. 7553, p. 436, 2015.

[74] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine, vol. 29*, no. 6, p. 82–97, 2012.

[75] I. S. A. Krizhevsky and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, p. 1097–1105, 2012.

[76] L. Deng, X. Yu, and et al., "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing, vol.7*, no. 3–4, pp. 197–387, 2014.

[77] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[78] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, p. 273–297, 1995.

[79] R. Gandhi, "Support vector machine — introduction to machine learning algorithms."

[80] A. Pant, "Introduction to logistic regression."

[81] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[82] T. Yiu, "Understanding random forest."

[83] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, (USA), p. 278, IEEE Computer Society, 1995.

[84] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[85] Y.-Y. Chen, Y.-H. Lin, C.-C. Kung, M.-H. Chung, and I.-H. Yen, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," *Sensors*, vol. 19, p. 2047, 05 2019.

[86] K. Fukushima, "Neocognitron," *Scholarpedia*, vol. 2, no. 1, p. 1717, 2007. revision #91558.

[87] C. C. Chatterjee, "Basics of the classic cnn."

[88] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[89] D. F. Specht, "Generation of polynomial discriminant functions for pattern recognition," *IEEE Transactions on Electronic Computers*, vol. EC-16, no. 3, pp. 308–319, 1967.

[90] "Markov chain. definition of markov chain in us english by oxford dictionaries." https://www.lexico.com/en/definition/markov_chain. Oxford Dictionaries, English.

[91] J. Martinek, M. Tatar, and M. Javorka, "Distinction between voluntary cough sound and speech in volunteers by spectral and complexity analysis," *Journal of physiology and pharmacology : an official journal of the Polish Physiological Society*, vol. 59 Suppl 6, pp. 433–40, 01 2009.

[92] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[93] C. Chen, G.and Parada and G. Heigold, "Small-footprint keyword spotting using deep neural networks," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4087–4091, 2014.

[94] L. Rabiner, "Fundamentals of speech recognition," *Fundamentals of speech recognition*, 1993.

[95] R. C. Rose and D. B. Paul, "A hidden markov model based keyword recognition system," in *International Conference on Acoustics, Speech, and Signal Processing*, pp. 129–132 vol.1, 1990.

[96] S. Khomsay, R. Vanijjirattikhan, and J. Suwatthikul, "Cough detection using pca and deep learning," pp. 101–106, 10 2019.

[97] D. Wyatt, T. Choudhury, and J. Bilmes, "Conversation detection and speaker segmentation in privacy-sensitive situated speech data," vol. 1, pp. 586–589, 01 2007.

[98] M. Pathak, *Privacy Preserving Techniques for Speech*. PhD thesis, 2010.

[99] F. Chen, J. Adcock, and S. Krishnagiri, "Audio privacy: Reducing speech intelligibility while preserving environmental sounds," pp. 733–736, 01 2008.

[100] J. S. J. Korpas and M. Vrabec, "Analysis of the cough sound: an overview," *Pulmonary Pharmacology*, no. 5-6, pp. 261–268, 1996.

[101] A. A. T. Drugman, "Joint robust voicing detection and pitch estimation based on residual harmonics," *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[102] S. Dubnov, "Generalization of spectral flatness measure for non-gaussian linear processes," p. 698–701, 2004.

[103] "A large set of audio features for sound description - technical report," *IRCAM*, 2003.

[104] T. Inouye and et al., "Quantification of eeg irregularity by use of the entropy of the power spectrum," *Electroencephalography and clinical neurophysiology*, vol. 79, pp. 204–210, 1991.

[105] D. Giannoulis, M. Massberg, and J. D. Reiss, "Automating dynamic range compression," *Journal of the Audio Engineering Society*, vol. 61, 2013.

[106] S. Dixon, "Onset detection revisited," *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006.

[107] J. W. J. M., Gordon, "Perceptual effects of spectral modifications on musical timbres," *Journal of the Acoustical Society of America*, vol. 63, p. 1493–1500, 1978.

[108] "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4*, pp. 357–366, 1980.

[109] A. A. X. Huang and H. Hon, "Spoken language processing: A guide to theory, algorithm, and system development," *Prentice Hall*, 2001.

[110] A. R. Mohamed, "Deep neural network acoustic models for asr," 2014.

[111] H. W.Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[112] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.

[113] H. Toutenburg, "Hollander, m., d. a. wolfe: Nonparametric statistical methods. john wiley ∧ sons, new york-sydney-tokyo-mexico city 1973. 503 s.," *Biometrische Zeitschrift*, vol. 17, no. 8, pp. 526–526, 1975.

[114] B. Auffarth, M. López, and J. Cerquides, "Comparison of redundancy and relevance measures for feature selection in tissue classification of ct images," in *ICDM*, 2010.

[115] C. Ding and H. Peng, "Peng, h.: Minimum redundancy feature selection from microarray gene expression data. journal of bioinformatics and computational biology 3(2), 185-205," *Journal of bioinformatics and computational biology*, vol. 3, pp. 185–205, 05 2005.

[116] G. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1315–1321, 1999.

[117] V. Jagannath, "Diagram of a random decision forest."

[118] S. Firmin, "Seeing the forest for the trees: An introduction to random forest."

[119] A. A. AnAj, "Example of k-nearest neighbour classificationnb."

[120] "What is a neural network? 3 things you need to know."

[121] "Why can rnns with lstm units also suffer from "exploding gradients"?," 2018.

[122] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[123] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[124] C. M. Bishop. Springer, 2006.

[125] J. Brownlee, "Gentle introduction to the adam optimization algorithm for deep learning," 2017.

[126] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *J. Mach. Learn. Res.*, vol. 11, p. 2079–2107, Aug. 2010.

[127] "Newbie question: Confused about train, validation and test data!," 2015.

[128] J. Schneider, "Cross validation," 1997.

[129] G. J. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing microarray gene expression data*. Wiley, 2004.

[130] Y. Li, C. Wang, and K. Han, "Rfamyloid: A web server for predicting amyloid proteins," *International journal of molecular sciences*, vol. 19, 07 2018.

[131] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77 – 89, 1997.

[132] D. Powers, "Evaluation: From precision, recall and f-factor to roc, informedness, markedness ∧ correlation," *Mach. Learn. Technol.*, vol. 2, 01 2008.

[133] D. G. Altman and J. M. Bland, "Statistics notes: Diagnostic tests 1: sensitivity and specificity," *BMJ*, vol. 308, no. 6943, p. 1552, 1994.

[134] Y. Sasaki, "The truth of the f-measure," *Teach Tutor Mater*, 01 2007.

[135] user2149631 (https://stats.stackexchange.com/users/82986/user2149631), "Will roc curve for a model always be symmetric if we have enough training data?." Cross Validated. URL:https://stats.stackexchange.com/q/264477 (version: 2018-01-31).

[136] M. Zweig and G. Campbell, "Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine.," *Clinical chemistry*, vol. 39 4, pp. 561–77, 1993.

[137] S. Siegel and J. N. John Castellan, *Nonparametric Statistics for the Behavioral Sciences (second edition)*. New York: McGraw-Hill, 1988.

[138] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[139] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.