Biomedical Signal Processing and Control 71 (2022) 103175

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Automated detection of COVID-19 cough

Alberto Tena [a], Francesc Clarià [b], Francesc Solsona [b,*]

[a] CIMNE, Building C1, North Campus, UPC. Gran Capità, 08034 Barcelona, Spain
[b] Dept. of Computer Science & INSPIRES, University of Lleida. Jaume II 69, E-25001 Lleida, Spain

ARTICLE INFO

Keywords:
Keyword: COVID-19
Automated cough detection
Diagnosis
Signal processing
Time–frequency

ABSTRACT

Easy detection of COVID-19 is a challenge. Quick biological tests do not give enough accuracy. Success in the fight against new outbreaks depends not only on the efficiency of the tests used, but also on the cost, time elapsed and the number of tests that can be done massively. Our proposal provides a solution to this challenge. The main objective is to design a freely available, quick and efficient methodology for the automatic detection of COVID-19 in raw audio files.

Our proposal is based on automated extraction of time–frequency cough features and selection of the more significant ones to be used to diagnose COVID-19 using a supervised machine-learning algorithm.

Random Forest has performed better than the other models analysed in this study. An accuracy close to 90% was obtained.

This study demonstrates the feasibility of the automatic diagnose of COVID-19 from coughs, and its applicability to detecting new outbreaks.

1. Introduction

COVID19 (COronaVIrus Disease of 2019), caused by the Severe Acute Respiratory Syndrome (SARS-CoV2) virus, was announced as a global pandemic on February 11, 2020 by the World Health Organisation (WHO). By mid-February, 2021, one year after the beginning of the COVID-19 pandemic, over 108 million confirmed cases of COVID-19 had been reported worldwide, with almost 2,400,000 deaths [1].

During this time, it has been demonstrated that COVID-19 outbreaks are very hard to contain with current testing approaches unless region-wide confinement measures are sustained. This is partly because of the limitations of current viral and serological tests and the lack of complementary pre-screening methods [2].

According to the WHO-China Joint Mission report (COVID-19) [3], typical signs and symptoms of COVID-19 are fever (87.9%), dry cough (67.7%), fatigue (38.1%), sputum production (33.4%), shortness of breath (18.6%), sore throat (13.9%), headache (13.6%), myalgia or arthralgia (14.8%), chills (11.4%), nausea or vomiting (5.0%), nasal congestion (4.8%), diarrhoea (3.7%), hemoptysis (0.9%), and conjunctival congestion (0.8%).

Several researchers have proposed methods for identifying cough sounds from audio recordings [4,5]. Automatic cough classification is an active research area in which several researchers have proposed methods for identifying a wide range of respiratory diseases and types of coughs (namely dry and wet coughs) through cough analysis and machine-learning algorithms [6,7].

Various studies have begun to work on the design of machine-learning tools to detect COVID-19 [8–16] as complementary pre-screening method. These are based on the analysis of the sound of voices, and the sounds we make when we breath or cough and which change when our respiratory system is affected. These changes range from coarse, clearly audible changes, to minute changes (called *micro* signatures) that are inaudible to the untrained listener, but nevertheless present [9]. These works have been performed in own datasets and no idenfication of the main features has been performed. We are also interested in the automatic identification of COVID-19 cough from any raw audio recording. Overall, finding a general method and the main cough features from audio records for diagnosing COVID-19 is a challenge.

The difficulty is to find good machine-learning features. Some works in the literature, as we have mentioned before, advocate some features, but in the particular case of COVID-19, it remains to be seen which properties, brands, signs (that is, features) are those that uniquely identify COVID-19. So, the big challenge is to identify the best features that discriminate the COVID-19 cough. In addition, we want to find the group of features with better performance for each type of experiment,

[*] Corresponding author.
E-mail addresses: atena@cimne.upc.edu (A. Tena), francisco.claria@udl.cat (F. Clarià), francesc.solsona@udl.cat (F. Solsona).

https://doi.org/10.1016/j.bspc.2021.103175
Received 16 July 2021; Received in revised form 26 August 2021; Accepted 7 September 2021
Available online 13 September 2021

as for example, comparing COVID-19 and pertussis coughs.

The goal of this paper is to develop a pre-screening method that could lead to automated identification of COVID-19 through the analysis of cough time–frequency representations (TFR) with similar performance presented in [8–16]. TFRs permit the evolution of the periodicity and frequency components over time to be observed, allowing the analysis of non-stationary signals. Moreover, this representation, which maintains the time dependence of signal features, gives the possibility of introducing more related features than traditional analysis. This way, we go a step further by finding the set of time–frequency features that could allow COVID-19 coughs to be distinguished from other cough patterns and validate it as a more generic proposal by applying our method to various datasets from different sources.

In the present work, prior to performing the TFR analysis, the YAMNet [17] deep neuronal network was used for the automatic identification of cough sounds in raw audio files. Then, a TFR analysis of a Choi-Williams distribution (CWD) was carried out in the cough-samples identified to obtain discriminatory features for an automated diagnosis of COVID-19. 39 features were extracted and the sets which showed better performance at discriminating COVID-19 cough were selected. For that purpose, the main objectives (and contributions) of this research are:

- To design a free, quick and efficient methodology for the automatic detection of COVID-19 in raw audio files based on the time–frequency analysis of the cough.
- To obtain the time–frequency discriminatory features leading to automated identification of COVID-19.
- To find an optimal supervised machine-learning algorithm to diagnose COVID-19 from the cough features found.

2. Methods

The methods presented in this section were implemented and a synthetic dataset based on a random sample of COVID-19 and non-COVID-19 coughs is freely available online [18]. It was built in R using the synthpop package [19]. Also, the code of the machine-learning models used is also provided.

This section presents the corpus, the automatic cough identification process and the basis theory used to obtain the time–frequency features. The classification models were fitted by a set of the most important features, obtained by two different techniques, namely feature selection and feature extraction. The most popular supervised models in cough classification are then presented. Finally, the model's performance metrics are introduced.

2.1. Data Corpus

This section describes the data collection framework used in this work. It consisted of the COVID-19 dataset the University of Lleida collected for this study which was approved by the Research Ethics Committee for Biomedical Research Projects (CEIm) at the University Hospital Arnau de Vilanova of Lleida, and three additional existing publicly available COVID-19 datasets, namely University of Cambridge [20], Coswara [21] and Virufy [22] datasets. Additionally, the Pertussis dataset [6], which includes recordings of patients with pertussis cough, was also used.

Our analysis used four sets. The first set (C) consisted of subjects tested COVID-19 positive; the second set (N) were subjects tested COVID-19 negative; the third set (NC) were non-COVID subjects, but who had non-specified-coughs as a symptom; and the fourth set (PT) were non-COVID subjects but who presented pertussis cough.

Table 1 shows the set of participants selected and Table 2 shows the demographic data for each group.

Table 1
Corpus. UdL: University of Lleida; UC: University of Cambridge.

	UdL	UC	Coswara	Virufy	Pertussis	Total
C	49	142	107	48	0	346
N	3	137	133	73	0	346
NC	0	53	48	0	0	101
PT	0	0	0	0	20	20
Total	52	332	288	121	20	813

Table 2
Demographic dataset properties. NA: Data not-available.

	C	N	NC	PT
Males (%)	68.0	50.5	55.2	NA
Females (%)	32.0	49.5	44.8	NA
Age	48.9 ± 11.9	40.8 ± 9.1	44.6 ± 7.3	NA

2.2. Automatic Cough Identification

Fig. 1 shows an overview of the automatic cough identification process developed which was inspired by [23].

The YAMNet deep neuronal network [17] was used for the automatic identification of the cough samples registered in the raw audio files. YAMNet classifies audio segments into sound classes described by the AudioSet ontology [24] employing MobileNet [25]. The MobileNet structure is built on depthwise separable convolutions which factorises a standard convolution into a depthwise and a pointwise convolution (1 x 1 convolution kernel) [26]. Depthwise convolution applies the filter to each input channel, and 1 x 1 pointwise convolution is used to combine the outputs of the depthwise convolution. The YAMNet body architecture employing MobileNet is defined in Fig. 2.

All layers are depthwise separable convolutions except for the first layer, which is a standard convolution, and the last few layers which are pooling, fully connected layers, and a softmax layer for classification. Each convolution layer used ReLU as the activation function, and batchnorm was used for the standardised distribution of batches. The convolution layer structure is shown in Fig. 3.

To obtain the input layer passed to YAMNet, the original audio waveforms of the raw audio files were pre-processed. They were resampled to 16 kHz and buffered into L overlapping segments. Each segment was 0.98 s and the segments were overlapped by 0.8575 s. They were converted to a magnitude spectrogram with 257 frequency bins using a one-sided short-time Fourier transform (STFT) with a 25-ms periodic Hann window with a 10-ms hop and a 512-point Discrete Fourier Transform (DFT). Then, the magnitude spectrum was passed through a 64-band mel-spaced filter bank and the magnitudes of each band were summed. The audio was represented by a 96-by-64-by-1-by-L array, where 96 is the number of spectrums in the mel spectrogram and 64 is the number of mel bands. Finally, the mel spectrograms were converted to a log scale. The 96-by-64-by-1-by-L array of mel spectrograms was the input layer passed through YAMNet. The output from YAMNet (L-by-512 matrix) corresponds to confidence scores for each of the 521 sound classes over time.

The post-processing consisted of selecting the sound regions labeled as "cough" for analysis. Firstly, to detect the sound event region, the 521 confidence signals were passed through a moving mean filter with a window length of 7 and each signal through a moving median filter with a window length of 3. Although other better filters exists, combining mean and median filters offers good performance at reasonable computational costs [27]. The window length of the mean filter was computed as the $Segment\_duration/Hope\_length - 1$ where *Segment_duration* was the duration of the L segments (0.98 s) and *Hope_length* was the hope length between two consecutive segments (0.1225 s). The length of the median filter was established considering optimal computational costs.

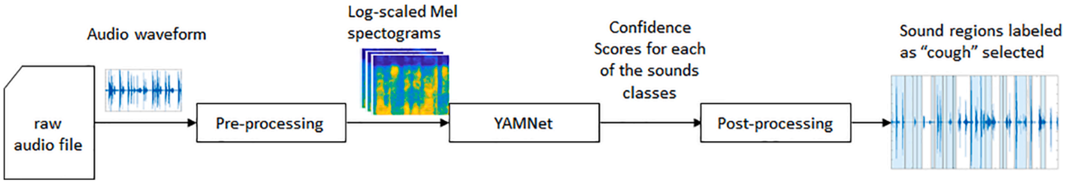

Audio waveform
Log-scaled Mel spectograms
Confidence Scores for each of the sounds classes
Sound regions labeled as "cough" selected
raw audio file
Pre-processing
YAMNet
Post-processing

Fig. 1. Overview of the automatic cough identification process.

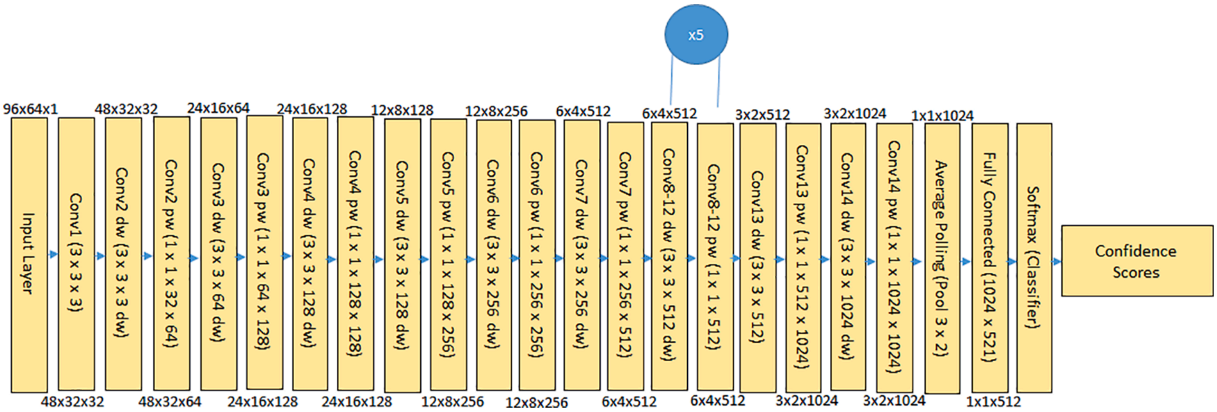

x5
96x64x1
48x32x32
24x16x64
24x16x128
12x8x128
12x8x256
6x4x512
6x4x512
3x2x512
3x2x1024
1x1x1024
Input Layer
Conv1 {3 x 3 x 3}
Conv2 dw {3 x 3 x 3 dw}
Conv2 pw {1 x 1 x 32 x 64}
Conv3 dw {3 x 3 x 64 dw}
Conv3 pw {1 x 1 x 64 x 128}
Conv4 dw {3 x 3 x 128 dw}
Conv4 pw {1 x 1 x 128 x 128}
Conv5 dw {3 x 3 x 128 dw}
Conv5 pw {1 x 1 x 128 x 256}
Conv6 dw {3 x 3 x 256 dw}
Conv6 pw {1 x 1 x 256 x 256}
Conv7 dw {3 x 3 x 256 dw}
Conv7 pw {1 x 1 x 256 x 512}
Conv8-12 dw {3 x 3 x 512 dw}
Conv8-12 pw {1 x 1 x 512}
Conv13 dw {3 x 3 x 512}
Conv13 pw {1 x 1 x 512 x 1024}
Conv14 dw {3 x 3 x 1024 dw}
Conv14 pw {1 x 1 x 1024 x 1024}
Average Polling {Pool 3 x 2}
Fully Connected {1024 x 521}
Softmax {Classifier}
Confidence Scores
48x32x32
48x32x64
24x16x128
24x16x128
12x8x256
12x8x256
6x4x512
6x4x512
3x2x1024
3x2x1024
1x1x512

Fig. 2. YAMNet Body Architecture. Conv: Convolution. dw: Depthwise. pw: Pointwise.

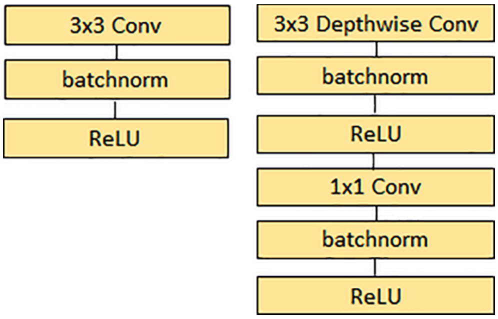


Fig. 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

Then, the confidence signals were converted into binary masks. After running several trials, a threshold of 0.35 was set because it showed the best performance at detecting "cough" samples. Any sound shorter than 0.5 s was discarded for analysis and regions shorter than 0.25 s were merged.

The identified sound regions that overlapped by 50% or more were consolidated into single regions. The region start time selected was the smallest start time and the region end time selected was the largest end time of all sounds in the group.

Then, the sound regions labelled as "cough" by YAMNET were selected for analysis. The boundaries of these cough samples were selected by using the detectSpeech algorithm available in [23], which is based on [28] using a Hann window with 0.03·*Sampling_rate* seconds hop. Finally, the first 600 ms of each cough sample identified were re-sampled at 8,820 Hz and normalised to obtain the Time–frequency representations and features.

Fig. 4 illustrates the process of the automatic identification of cough boundaries in a raw audio file. Fig. 4a shows the sound classification performed by YAMNET. Fig. 4b shows the resulting audio signal after the selection of those audio regions labelled as "cough". Fig. 4c shows the boundaries of the cough samples defined for analysis.

2.3. Time–frequency Representation

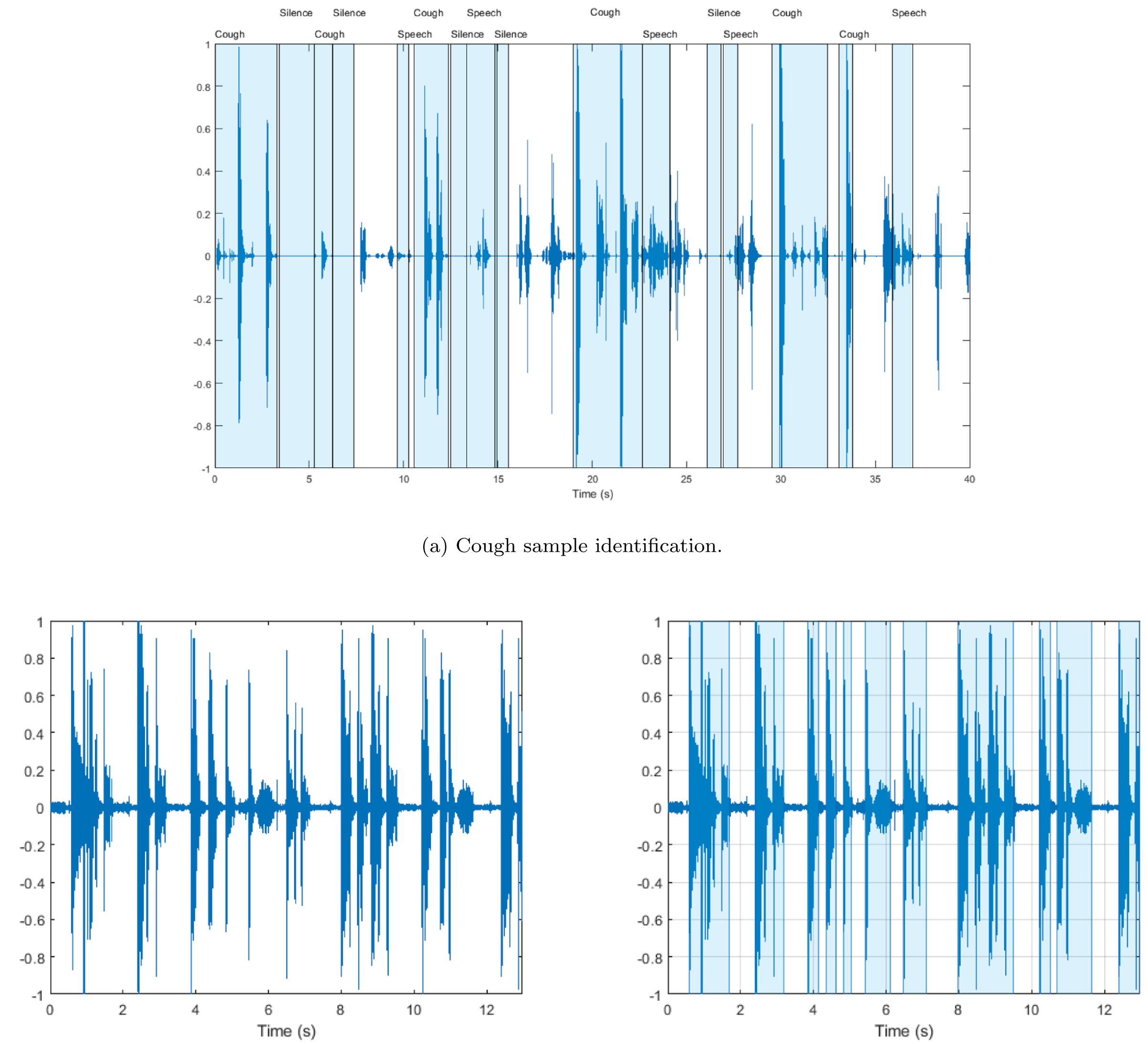
The Wigner distribution (WD) has been used in different fields and applied to the study of time-varying and strongly non-stationary systems. Since the energy is a quadratic representation of the signal, the quadratic structure of the time–frequency representation (TFR) is intuitive and reasonably accepted when the TFR is interpreted as an energy distribution in time and frequency [29]. From all TFRs that represent energy, the WD satisfies many desired mathematical properties. For example, the WD is always real, symmetrical with respect to the time and frequency axes, satisfying the marginal properties and the instantaneous frequency. Furthermore, the group delay may be obtained. Eq. 1 represents the WD of the signal *x(t)*.

$$WD\left(t,f\right)=\int x\left(t+\tau\Big/2\right)x^{*}\left(t-\tau\Big/2\right)e^{-j2\pi f\tau}d\tau, \tag{1}$$

where t and f represent time and frequency respectively, and $x^{*}(t)$ is the conjugate of $x(t)$.

Basically, the WD of a real signal *x(t)* is calculated in a similar way to a convolution. At each particular time, the signal is overlapped by itself and inverted on the time axis, and multiplied by itself. Finally, the Fourier transform of this product is carried out. Note that neither will the WD be necessarily zero when *x(t)* nor would the WD necessarily be zero at frequencies that do not exist in the spectrum. Evidence of this phenomenon has been called interference terms and cross-terms. The interference terms are undesired since they make it difficult to obtain a clear and intuitive spectrum of the signal, as two energy regions perfectly delimited are expected to be obtained.

The possibility of using the WD as a representation of the signal spectral density at each particular time induces the generation of another distribution from the WD to minimise these interference terms while simultaneously maintaining certain properties. To achieve this, we calculated the convolution of the WD of each cough sample was calculated with the Choi–Williams exponential function *h(t,f)* [30] (Eq. 2). By convolving the Wigner distribution with the Choi–Williams exponential, the Choi–Williams distribution (CWD) was obtained (Eq. 3).



(a) Cough sample identification.

(b) Selection of cough samples.

(c) Cough sample boundaries for analysis.

Fig. 4. Automatic identification of cough samples in a raw audio file.

$$h\left(t,f\right) = \sqrt{\frac{4\pi}{\sigma_c}} e^{-4\pi^2\frac{(tf)^2}{\sigma_c}}, \tag{2}$$

where σ_c is a scaling factor.

$$CWD\left(t,f\right) = \iint h\left(t-t',f-f'\right) WD\left(t',f'\right) dt'\,df' \tag{3}$$

CWD preserves the properties of WD [30,31], such as the marginal properties and instantaneous frequency. Moreover, it is able to reduce the WD interference by estimating an adequate σ_c parameter. In this study, the σ_c parameter was established at 0.05 to eliminate the interference produced. So, the CWD is a new function of the time–frequency distribution that allows the interference terms to be minimised.

Then, in order to obtain statistical parameters, the density function $CWD(f,t)$ was normalised to have an area equal to 1. So, it can be associated with a joint probability density function $CWD_N(f,t)$ of the time and frequency variables. Their marginal distributions, which do not contain the interference, still represent, although in a normalised manner, the instantaneous power (Eq. 4) and and spectral density energy (Eq. 5) of the original signal.

$$m_t\left(t\right) = \int_{-\infty}^{\infty} CWD_N\left(f,t\right) df = |x(t)|^2 \tag{4}$$

$$m_f\left(f\right) = \int_{-\infty}^{\infty} CWD_N\left(f,t\right) dt = |X(f)|^2 \tag{5}$$

Therefore, the group delay (Eq. 6) and the mean frequency of the spectrum (Eq. 7) can be defined as:

$$t_g = \iint tCWD_N\left(t,f\right) dt\,df \tag{6}$$

$$f_m = \iint fCWD_N\left(t,f\right) dt\,df \tag{7}$$

The joint time–frequency moments of a non-stationary signal comprise a set of time-varying parameters that characterise the signal spectrum as it evolves over time. They are related to the conditional temporal moments and the joint time–frequency moments. The joint time–frequency moment is an integral function of frequency, given time, and marginal distribution. The conditional temporal moment is an integral function of time, given frequency, and marginal distribution. The calculation of the joint time–frequency moment $t^n f^m$ (Eq. 8) is a double integral through time and frequency [32].

$$\langle t^n f^m \rangle = \iint \left(t - t_g\right)^n \left(f - f_m\right)^m CWD_N\left(t,f\right) dt\, df \tag{8}$$

where n and m are the frequency and time moment orders.

The moments of the marginal density functions, that define the relationship between $m_t(t)$ and $m_f(f)$, $\langle m_t(t)^n m_f(f)^m \rangle$, are given in Eq. 9.

$$\left\langle m_t(t)^n m_f(f)^m \right\rangle = \frac{1}{std\left(m_t(t)^n\right) std\left(m_f(f)^m\right)} \iint \left(m_t\left(t\right) - \overline{m_t(t)}\right)^n \left(m_f\left(f\right) - \overline{m_f\left(f\right)}\right)^m dt\, df \tag{9}$$

CWD minimises the interference. However, negative values still remain. To solve this issue, the CWD was reformulated as the product of its marginal distributions. Therefore, the joint probability density distribution *pD* (Eq. 10) was obtained. This procedure was only possible because the marginal distributions of the CWD were statistically independent. To corroborate this, the moments of the CWD_N from $n = 1$ and $m = 1$ to $n = 15$ and $m = 15$ were computed, and little covariability was observed. This meant that the marginal distributions could be considered statistically independent.

$$pD(f,t) = m_f\left(f\right) \cdot m_t\left(t\right) \tag{10}$$

Fig. 5(a) corresponds to the WD of a cough sample. It shows how the interference terms of the WD make it difficult to obtain a clear and intuitive spectrum of the signal. The new function $pD(f,t)$ (Fig. 5(c)) is equal to WD without interference (as CWD, Fig. 5(b)) and negative values.

2.4. Time–frequency features

This section explains how a total of 39 features were obtained from the time frequency representation of each cough sample. 28 of them corresponded to the instantaneous spectral energy, instantaneous frequency, instantaneous frequency peak and spectral information. These were obtained by dividing the spectrum (0–4,410 Hz) into 7 frequency bands: 1, 0–80 Hz; 2, 80–250 Hz; 3, 250–550 Hz; 4, 550–900 Hz; 5, 900–1,500 Hz; 6, 1,500–3,000 Hz; 7, 3,000–4,410 Hz. The mean frequency of the total spectrum, the joint, instantaneous and spectral Shannon entropies, the Kurtosis, 3 joint time–frequency moments and 3 joint moments of the marginal signals of instantaneous power and spectral density were also computed.

The instantaneous spectral energy, *E(t)* (Eq. 11), was calculated for each cough sample as the $pD(f,t)$ integral in the frequency domain. Next, the instantaneous frequency, $f_{mi}(t)$, of the spectrum was computed [31] as the average frequency of the spectrum with respect to time (Eq. 12).

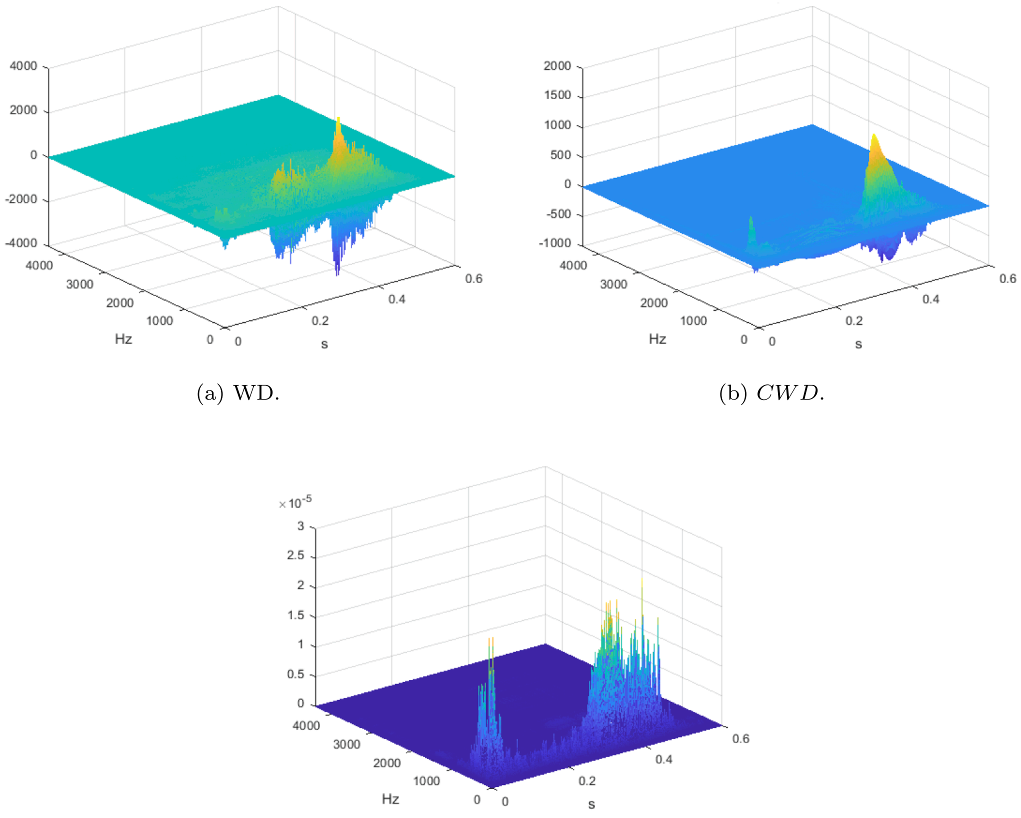
$$E\left(t\right) = \int_{f_1}^{f_2} pD\left(f,t\right) df, \tag{11}$$

where f_1 and f_2 are the lower and upper frequencies of each band.

$$f_{mi}\left(t\right) = \int_{f_1}^{f_2} \frac{1}{E(t)} f pD\left(f,t\right) df \tag{12}$$

The Instantaneous Frequency Peak, $f\_Cres(t)$ (Eq. 13), is defined as the maximum frequency value at every instant.

$$f\_Cres\left(t\right) = \frac{1}{E(t)} argmax_f \left[\prod_{f_1}^{f_2} f \cdot pD\left(f,t\right)\right] \tag{13}$$



(a) WD.

(b) *CWD*.

(c) pD.

Fig. 5. Time–frequency representations of the same COVID-19 subject.

Then, the joint Shannon (*H_tf*), instantaneous (*H_t*) and spectral information (*H_f*) entropies were measured by means of the Shannon entropy method. Shannon entropies were used to quantify the regularity, uncertainty or randomness of these distributions. Entropy can express the information mean that an event provides when it takes place, the uncertainty about the outcome of an event and the dispersion of the probabilities with which the events take place.

Therefore, to obtain the entropy measurements from the $pD(f,t)$ and with the aim of having a range of values able to discriminate levels of spectral amplitude accurately enough, $pD(f,t)$ was quantified with $N = 2^q$ levels and $q = 20$. When the joint probability density function is quantified ($pD_N(f,t)$), the joint Shannon entropy (*H_tf*), in this case in a range of 0 to 20 bits, can be obtained (Eq. 14).

$$H\_tf = -\int\int log_2\left(pD_N\left(t,f\right)\right)\cdot pD_N\left(t,f\right)dfdt \tag{14}$$

According to Eq. 10, the joint probability density distribution quantified ($pD_N(f,t)$) is defined in Eq. 15.

$$pD_N\left(t,f\right) = m_{fN}\cdot m_{tN}, \tag{15}$$

where $m_{tN}(t)$ is the quantified instantaneous marginal obtained from the $m_t(t)$ and $m_{fN}(f)$ is the quantified frequency marginal obtained from the $m_f(f)$. Therefore, the joint entropy can also be expressed as in Eq. 16.

$$H\_tf = H\_t + H\_f, \tag{16}$$

where *H_t* (Eq. 17) and *H_f* (Eq. 18) are the instantaneous and spectral entropy respectively.

$$H\_t = -\int log_2\left(m_{tN}\left(t\right)\right)\cdot m_{tN}\left(t\right)dt \tag{17}$$

$$H\_f = -\int log_2\left(m_{fN}\left(f\right)\right)\cdot m_{fN}\left(f\right)df \tag{18}$$

Then, the spectral information, $IE(f)$ (Eq. 19) is obtained.

$$IE\left(f\right) = -log_2\left(m_{fN}\left(f\right)\right) \tag{19}$$

Then, the Kurtosis (K) can be found (Eq. 20).

$$K = \left\langle m_t(t)^n m_f(f)^m\right\rangle \tag{20}$$

for $n = 4$ and $m = 0$.

Starting from the computed parameters $E(t), f_m, f_{mi}(t), f\_Cres(t), H\_tf, H\_t, H\_f, IE(f), \langle t^n f^m\rangle$ and K, a total of 39 features were obtained. The averages of $E(t), f_{mi}(t), f\_Cres(t)$ and $IE(t)$ were obtained for each of the 7 frequency bands: 1, 0–80 Hz; 2, 80–250 Hz; 3, 250–550 Hz; 4, 550–900 Hz; 5, 900–1,500 Hz; 6, 1,500–3,000 Hz; 7, 3,000–4,410 Hz. The joint time–frequency moments $\langle t^n f^m\rangle$ for $n = 1$ and $m = 1, n = 7$ and $m = 7$ and $n = 15$ and $m = 15$, and the same joint moments of the marginal signals of instantaneous power and spectral density $\langle m_t(t)^n m_f(f)^m\rangle$, were considered for analysis among all the moments computed. Then, the 39 features obtained were coded as follows:

- f_Cres1…f_Cres7: As the average of f_Crest(t) for each 7-bands.
- Enr_Bn1…Enr_Bn7: As the average of E(t) for each 7-bands.
- fm: As the value of the parameter f_m.
- f_Med1…f_Med7: As the average of $f_{mi}(t)$ for each 7-bands.
- IE_Bn1…IE_Bn7: As the average of $IE(f)$ for each 7-bands.
- H_tf: As the value of the parameter H_tf.
- H_f: As the value of the parameter H_f.
- H_t: As the value of the parameter H_t.
- kurt_Mgt: As the value of the parameter K.
- momC11: As the value of the $n = 1$ and $m = 1$ joint time–frequency moment.
- momC77: As the value of the $n = 7$ and $m = 7$ joint time–frequency moment.
- momC15: As the value of the $n = 15$ and $m = 15$ joint time–frequency moment.
- momM11: As the value of the $n = 1$ and $m = 1$ joint instantaneous power and spectral density moment.
- momM77: As the value of the $n = 7$ and $m = 7$ joint instantaneous power and spectral density moment.
- momM15: As the value of the $n = 15$ and $m = 15$ joint instantaneous power and spectral density moment.

2.5. *Feature selection*

The Recursive Feature Elimination (RFE) is a recursive process that ranks features according to some measure of their importance. At each iteration, feature importance is measured and the less relevant one is removed. The recursion is needed because for some measures the relative importance of each feature can change when evaluated over a different subset of features during the stepwise elimination process. RFE was implemented in R by using the caret package to select the set of features (S_i) which obtained the best accuracy for each classification model. Performance evaluation of each set of features was done by using stratified 10-fold cross-validation [33].

2.6. *Feature extraction*

Feature extraction is a process of dimensionality reduction by which an initial set of features is reduced while preserving the information in the original dataset. An Autoencoder was implemented in R using the Keras package to perform this task.

An Autoencoder is a specific type of a neural network, one mainly designed to encode the input data into a compressed and meaningful representation, and then decode it back so that the reconstructed input is similar as possible to the original. The Autoencoder maps the input data x to a hidden representation using the function $z = f(Px + b)$ parameterised by $\{P, b\}$. f is the activation function. The hidden representation is then mapped linearly to the output using $\widehat{x} = Wz + b'$. The parameters are optimised to minimise the mean square error of $\|\widehat{x} - x\|_2^2$ over all training points.

Fig. 6 shows the Autoencoder architecture employed. It consists of three modules: the encoder, the decoder and the bottleneck. The encoder is formed by an input layer of 39 nodes and two hidden layers of 30 and 20 nodes respectively. The bottleneck has 15 nodes and the decoder consists of two hidden layers of 20 and 30 nodes respectively and an output layer of 39 nodes. The activation function selected was the *tanh* function. As the purpose of our Autoencoder was to reduce the feature range of our original dataset, we took the compressed data contained in the bottleneck layer. So, the 39 original features were reduced to 15.

2.7. *Classification models*

Five groups of subjects (C, N, NC, PT and NNC) were defined for analysis. The C group contained COVID-19 subjects. N contained subjects tested COVID-19 negative who had no cough. NC was formed of non-COVID-19 subjects with non-specific–cough as a symptom. PT had non-COVID-19 subjects with pertussis cough. Finally, the NNC group merged all non-COVID-19 subjects (N, NC and PT). Then, four classification experiments were performed. These consisted of C vs. N, C vs. NC, C vs. PT and C vs. NNC.

The most popular supervised models in cough classification were used and were implemented in R. These were Random Forest (RF), Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Logistic Regression (LR) and Naïve Bayes (NB). The classification models were fitted on the one hand to the selected features obtained by means of RFE and, on the other hand, to the features extracted by means of the Autoencoder. Finally, 10-fold cross-validation [33] was

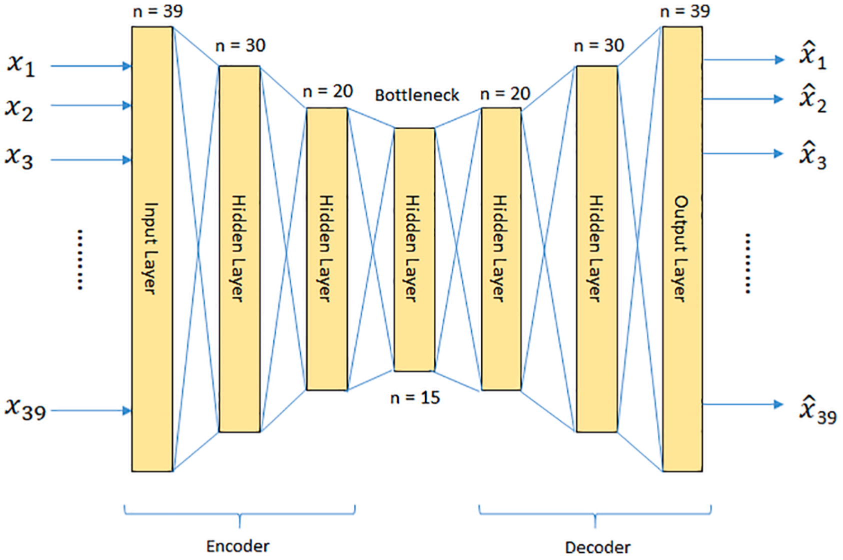


Fig. 6. Autoencoder Architecture.

implemented in R using the caret package to draw suitable conclusions. An upsampling technique with replacement was applied to the training data by making the group distributions equal to deal with the unbalanced dataset that could bias the classification models.

The first classifier employed was the RF. It was implemented using the R randomForest package with a forest of 500 decision tree predictors. The optimal number of features that were randomly distributed to each decision tree, was optimized for each classification problem by using the train function included in the R caret package. Each decision tree performed the classification independently and RF computed each tree predictor classification as one "vote". The majority of the votes computed by all the tree predictors decided the overall RF prediction.

Next, SVM, a powerful kernel-based classification paradigm, was implemented using the R e1071 package. A C-Support Vector Classification [34] was used with a linear kernel that was optimised through the tune function, assigning values 0.0001, 0.0005, 0.001, 0.01, 0.1, 1, 1.25, 1.5, 1.75, 2 and 5 to the C parameter, which controls the trade-off between a low training error and a low testing error. The value of *C* which gave the best performance was chosen.

Then, LDA was implemented using the R MASS package. This estimated the mean and variance from the training set and computed the covariance matrix to capture the co-variance between the groups to make predictions by estimating the probability that the test set belongs to every group.

LR was implemented by using the Gaussian generalised linear model, applying the R Stats package for binomial distributions. A logit link function was used to model the probability of "success". The purpose of the logit link was to take a linear combination of the covariate values and convert these into a probability scale.

Finally, standard NB based on applying Bayes' theorem was implemented using the e1071 package [35].

2.8. Performance metrics

There are four possible results in the classification task: If the sample is positive and it is classified as positive, it is counted as a *true positive* (TP) and when classified as negative, it is considered a *false negative* (FN). If the sample is negative and is classified as negative or positive, it is considered a *true negative* (TN) or *false positive* (FP) respectively. Based on that, the Accuracy, Sensitivity (also known as recall), Specificity, Precision and F-score metrics ([36]) were used to evaluate the performance of the classification models using a classification threshold of 50%. The Area Under the Curve (AUC) was also calculated.

- **Accuracy** (Eq. 21). Ratio between the correctly classified samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

- **Sensitivity** (Eq. 22). Proportion of correctly classified positive samples compared to the total number of positive samples.

$$Sensitivity = \frac{TP}{TP + FN} \tag{22}$$

- **Specificity** (Eq. 23). Proportion of correctly classified negative samples compared to the total number of negative samples.

$$Specificity = \frac{TN}{TN + FP} \tag{23}$$

- **Precision** (Eq. 24). Proportion of positive samples that were correctly classified compared to the total number of positive predicted samples.

$$Precision = \frac{TP}{FP + TP} \tag{24}$$

- **F-score** (Eq. 25). Harmonic mean of the precision and sensitivity.

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{25}$$

- **AUC** (Eq. 26). The Receiver operating characteristics (ROC) curve is a two-dimensional graph in which Sensitivity is plotted on the y-axis and 1 −*Specificity* is plotted on the x-axis. The points of the curve are obtained by sweeping the classification threshold from the most positive classification value to the most negative. The AUC score is a scalar value that measures the area under the ROC curve and is always bounded between 0..1.

$$AUC = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}1_{p_i > p_j}, \tag{26}$$

where i runs over all m samples with true label positive, and j runs over all n samples with true label negative; p_i and p_j denote the probability score assigned by the classifier to sample i and j, respectively.

3. Results

Firstly, a visual appraisal of time–frequency representations of coughs from C, N, NC and PT subjects is presented. Then, the distributions of the features obtained for each of the five groups defined for analysis were explored. Finally, the four experiments defined were implemented and the classification models were evaluated.

3.1. pD Representation

Fig. 7 shows the comparison of the $pD(f,t)$ of coughs from C, N, NC and PT subjects. Fig. 7(a) corresponds to a C subject who tested positive in a PCR. Figs. 7(b), 7(c) and 7(d) correspond to N, NC and PT subjects respectively.

The visual appraisal of Fig. 7(a) shows how the energy of the $pD(f,t)$ is concentrated in the frequency range from 0 to 1 kHz. In Fig. 7(b), low-energy frequency components can be observed at higher frequencies. In Fig. 7(c), low-energy frequency components can be also observed at higher frequencies but only ranged from 0 to 2 kHz. In Fig. 7(d) energy components of the $pD(f,t)$ can be observed in the frequency range from 0 to 3 kHz although the higher amplitudes are present in frequencies ranging from 0 to 1 kHz. It can be observed that there are no interference terms in any figure.

3.2. Data exploration

A total of 39 time–frequency features were obtained in this study: f_Cres1:f_Cres7, Enr_Bn1:Enr_Bn7, f_Med1:f_Med7, IE_Bn1:IE_Bn7, H_tf, H_t, H_f, fm, kurt_Mgt, MomC_11, MomC_77, MomC_1515, MomM_11, MomM_77 and MomC_1515.

There were remarkable differences in the mean and standard deviation between the features, and more specifically, in the following features (Fig. 8): f_Cres1, f_Cres3, Enr_Bn1, Enr_Bn2, Enr_Bn6, f_Med1, f_Med3, f_Med7, IE_Bn2, IE_Bn3, IE_Bn5 and IE_Bn7.

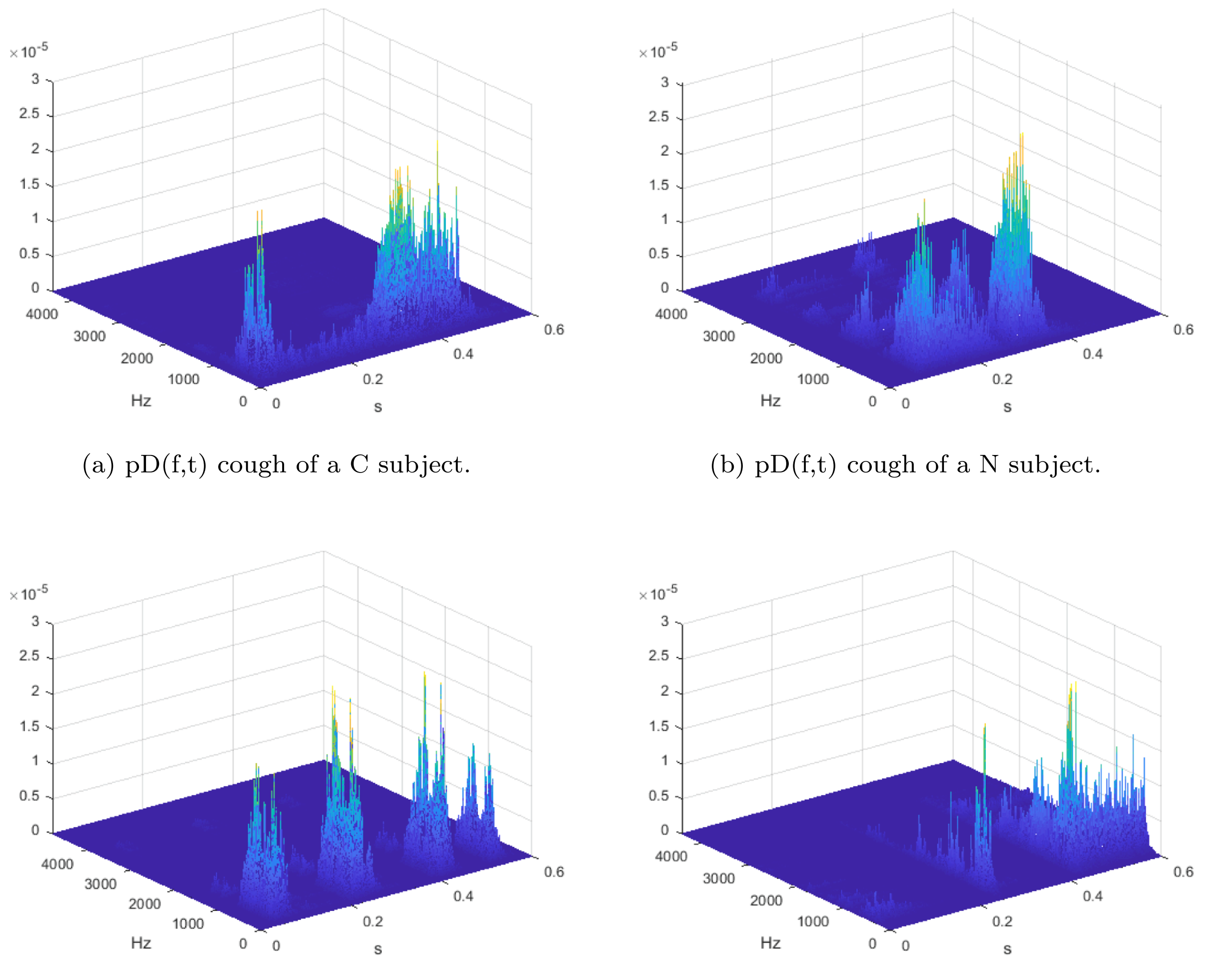
3.3. Feature Selection, Feature Extraction and Classification Models

The set of features which obtained the best accuracy by first applying RFE and then Autoencoder to each classification model were selected for analysis. Then, each classification model was applied to these selected features.

3.3.1. RFE

Table 3 shows the classification performance of the classification models fitted with the features selected by RFE tested for the 4 experiments defined.

In the first experiment, C vs. N, the results indicate that RF obtained



(a) pD(f,t) cough of a C subject.

(b) pD(f,t) cough of a N subject.

(c) pD(f,t) cough of a NC subject

(d) pD(f,t) cough of a PT subject

Fig. 7. $pD(f,t)$ cough representation of C, N, NC and PT subjects.

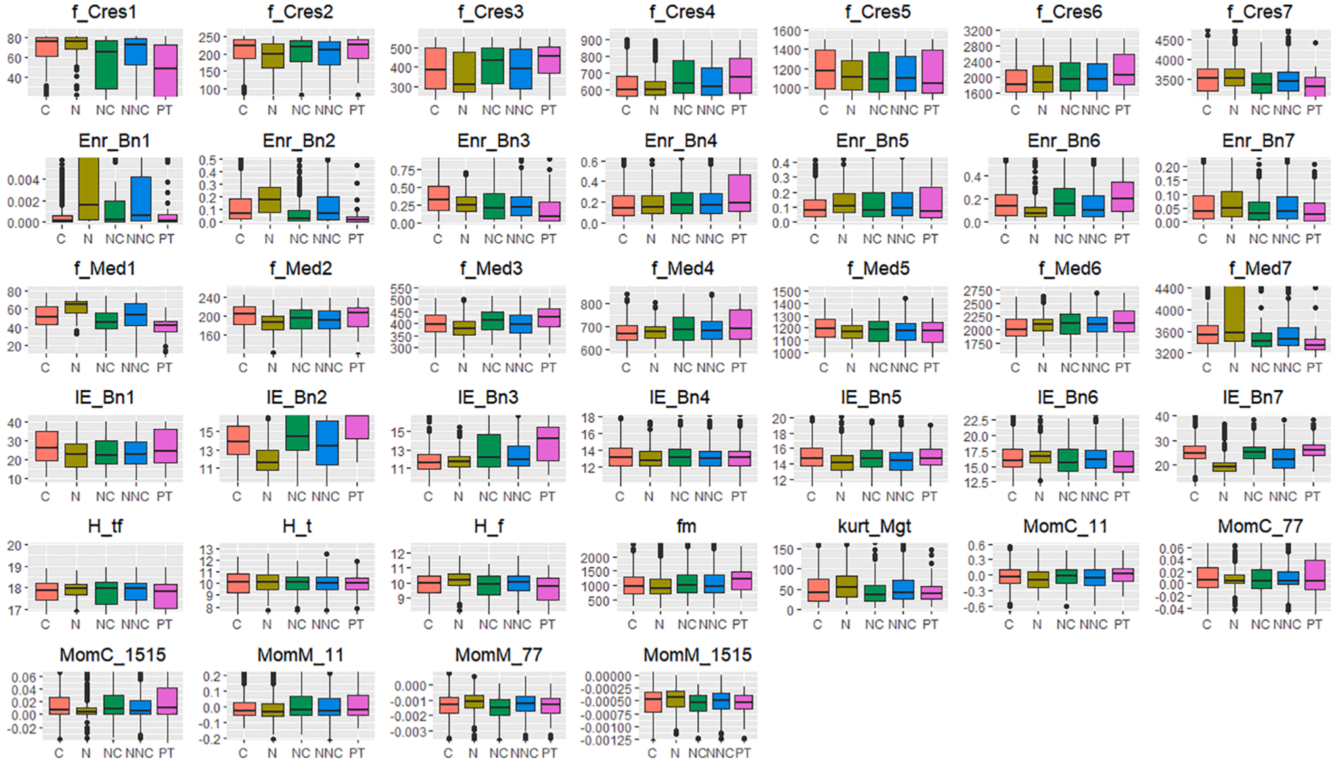


Fig. 8. Box plot of the time–frequency features obtained from C, N, NC, NNC and PT groups. Remarkable differences in the mean and standard deviation can be shown.

Table 3
Classification performance of the models fitted with the features selected previously with RFE.

		C vs. N	C vs. NC	C vs. NNC	C vs. PT
RF	Accuracy	**89.79**	**88.79**	**85.53**	**94.81**
	Sensitivity	**93.81**	95.49	**85.96**	**98.91**
	Specificity	81.54	**76.09**	**85.09**	72.00
	Precision	90.97	**88.42**	**85.14**	95.20
	F-score	**92.10**	**91.79**	**85.58**	**97.00**
	AUC	**96.04**	**92.53**	**89.65**	95.67
SVM	Accuracy	83.23	78.33	74.55	89.49
	Sensitivity	82.57	80.22	76.79	90.65
	Specificity	**84.90**	74.72	72.35	83.00
	Precision	**91.84**	85.76	73.80	96.75
	F-score	86.79	81.59	74.97	93.58
	AUC	92.15	88.35	75.73	**97.29**
LR	Accuracy	80.78	75.85	73.38	89.49
	Sensitivity	79.50	77.16	74.45	90.29
	Specificity	83.41	73.37	72.33	**85.00**
	Precision	90.84	84.78	73.02	**97.13**
	F-score	84.67	80.68	73.49	93.56
	AUC	92.73	87.98	75.86	88.82
NB	Accuracy	80.78	77.86	71.96	86.41
	Sensitivity	81.65	**95.86**	60.79	87.92
	Specificity	78.98	43.73	82.99	78.00
	Precision	88.95	76.39	77.87	95.75
	F-score	84.94	85.01	68.09	91.57
	AUC	87.50	82.07	73.94	92.06
LDA	Accuracy	79.56	76.32	72.32	84.16
	Sensitivity	78.08	78.24	73.91	84.00
	Specificity	82.68	72.68	70.57	**85.00**
	Precision	90.47	84.60	71.70	96.86
	F-score	83.53	81.20	72.52	89.95
	AUC	92.69	88.38	76.04	96.71

the best overall performance with an *Accuracy* = 89.79%, *Sensitivity* = 93.81, *F-score*= 92.10 and an *AUC* = 96.04. SVM obtained the best *Specificity*, 84.90% and the best *Precision*, 91.84%.

IE_Bn7, Enr_Bn1, IE_Bn2, f_Med1, IE_Bn1 were the top five features obtained with RFE which fitted the model that obtained the best overall performance (RF). The model was also fitted with f_Med7, IE_Bn5, Enr_Bn6, Enr_Bn7, fm, f_Med2, Enr_Bn3, f_Cres7, Enr_Bn5, H_f, kurt_Mgt, Enr_Bn2, IE_Bn6, IE_Bn3, f_Med6, MomC_1515, f_Cres1, f_Med5, f_Med4, f_Cres2, H_t, f_Cres6, Enr_Bn4, IE_Bn4, MomC_11, f_Med3, MomC_77, MomM_11, f_Cres3, H_tf, f_Cres5, MomM_77, MomM_1515 and f_Cres4, which were the set of features selected by RFE.

In the second experiment, C vs. NC, the results indicate that RF obtained the best *Accuracy* = 88.79%, *Specificity* = 76.09%, *Precision* = 88.42%, *F-score*= 91.79% and *AUC* = 92.53. NB obtained the best *Sensitivity* = 95.86%.

IE_Bn3, f_Med7, IE_Bn1, Enr_Bn2 and Enr_Bn3 were the top five features which fitted the model that obtained the best overall performance (RF). The remaining features selected by REF were Enr_Bn1, f_Med1, Enr_Bn4, IE_Bn2, f_Cres1, IE_Bn7, f_Med2, f_Med4, Enr_Bn5, MomM_11, IE_Bn6, f_Cres2, H_t, f_Med6, f_Med5, IE_Bn5, f_Med3, Enr_Bn7, kurt_Mgt, IE_Bn4, H_f, Enr_Bn6, fm, MomM_1515, H_tf, f_Cres6, f_Cres7, MomM77, f_Cres3, f_Cres4, MomC_1515, f_Cres5, MomC_77 and MomC_11.

In the third experiment, C vs. NNC, the results indicate that RF obtained the best *Accuracy* = 85.53%, *Sensitivity* = 85.96, *Specificity* = 85.09, *Precision* = 85.14, *F-score*= 85.58 and *AUC* = 89.65.

IE_Bn3, f_Med7, IE_Bn1, Enr_Bn2 and Enr_Bn3 were the top five features which fitted the model that obtained the best overall performance (RF). The remaining features selected by REF were Enr_Bn1, f_Med1, Enr_Bn4, IE_Bn2, f_Cres1, IE_Bn7, f_Med2, f_Med4, Enr_Bn5, MomM_11, IE_Bn6, f_Cres2, H_t, f_Med6, f_Med5, IE_Bn5, f_Med3, Enr_Bn7, kurt_Mgt, IE_Bn4, H_f, Enr_Bn6, fm, MomM_1515, H_tf, f_Cres6, f_Cres7, MomM77, f_Cres3, f_Cres4, MomC_1515, f_Cres5, MomC_77 and MomC_11.

In the fourth experiment, C vs. PT, the results indicate that RF obtained the best *Accuracy* = 94.81%, *Sensitivity*= 98.91 and *F-score* = 97.00. LR and LDA obtained the best *Specificity* = 85.00, LR obtained the best *Precision*= 97.13 and SVM obtained the best *AUC* = 97.29.

IE_Bn3, Enr_Bn4, Enr_Bn3, IE_Bn2, Enr_Bn2 were the top five features which fitted the RF model which obtained the best overall performance. The remaining features selected by RFE were f_Med1, IE_Bn1, f_Med7, f_Med4, f_Cres1, Enr_Bn1, IE_Bn6, f_Cres2, IE_Bn7, f_Med2 and f_Cres6.

3.3.2. Autoencoder

Then, the classification models were fitted with 15 features extracted by means of the Autoencoder. Table 4 shows the classification performance of the classification models tested for the 4 experiments defined.

In the first experiment, C vs. N, the results indicate that RF obtained the best *Accuracy* = 83.67%, *Sensitivity* = 89.58, *F-score* = 88.04 and *AUC* = 93.56. LDA obtained the best *Specificity*, 84.90% and the best *Precision*, 91.13%.

In the second experiment, C vs. NC, the results indicate that RF obtained the best *Accuracy* = 87.73%, *Sensitivity* = 96.94%, *Specificity* = 70.25%, *Precision* = 86.22%, *F-score* = 91.21% and *AUC* = 90.73.

In the third experiment, C vs. NNC, the results show that RF obtained the best *Accuracy* = 79.74%, *Sensitivity* = 79.70, *Specificity* = 79.79, *Precision* = 79.58, *F-score* = 79.52 and *AUC* = 83.57.

In the fourth experiment, C vs. PT, the results indicate that RF obtained the best *Accuracy* = 91.92%, *Sensitivity* = 97.29 and *F-score* = 95.32. LDA obtained the best *Specificity* = 83.00, SVM obtained the best *Precision* = 96.12 and LR obtained the best *AUC* = 95.72.

4. Discussion

This research directly addresses a recent statement released by the WHO [1] which believes in the use of rapid tests essential to control people infected with COVID-19. We demonstrated the feasibility of automatic detection of COVID-19 positives from the time–frequency analysis of coughs.

The visual appraisal of the time–frequency representations confirmed differences in the frequency distribution of the voluntary coughs of the C, N, NC and PT subjects.

The features selected by RFE to fit the models obtained better results on the overall performance of the models than those features extracted by means of the Autoencoder. Furthermore, the rank of the features selected by RFE which fitted the model that obtained the best performance depended highly on the experiment done. This means that when comparing coughs, a good selection of the features must be chosen.

Table 4
Classification performance of the models fitted with the 15 features extracted by means of the Autoencoder.

		C vs. N	C vs. NC	C vs. NNC	C vs. PT
RF	Accuracy	**83.67**	**87.73**	**79.74**	**91.92**
	Sensitivity	**89.58**	**96.94**	**79.70**	**97.29**
	Specificity	71.58	**70.25**	**79.79**	62.00
	Precision	86.62	**86.22**	**79.58**	93.48
	F-score	**88.04**	**91.21**	**79.52**	**95.32**
	AUC	**93.56**	**90.73**	**83.57**	95.01
SVM	Accuracy	79.57	71.85	68.30	81.72
	Sensitivity	77.54	72.85	67.97	81.85
	Specificity	83.79	70.03	68.61	81.00
	Precision	91.01	82.32	68.23	**96.12**
	F-score	83.36	77.18	68.00	88.22
	AUC	91.08	83.43	69.08	95.23
LR	Accuracy	79.21	69.97	66.60	78.98
	Sensitivity	76.99	70.16	65.84	79.17
	Specificity	83.79	69.65	67.36	78.00
	Precision	90.79	81.45	66.51	95.41
	F-score	83.04	75.29	66.11	86.16
	AUC	91.16	83.32	68.61	**95.72**
NB	Accuracy	76.53	73.97	70.98	84.00
	Sensitivity	73.04	80.21	72.66	86.15
	Specificity	83.80	62.18	69.32	72.00
	Precision	90.16	80.20	70.25	94.55
	F-score	80.47	80.03	71.35	90.08
	AUC	89.85	81.84	73.53	95.14
LDA	Accuracy	77.76	71.61	67.23	82.21
	Sensitivity	74.30	72.85	67.25	79.71
	Specificity	**84.90**	69.32	67.19	**83.00**
	Precision	**91.13**	81.92	67.00	95.73
	F-score	81.59	77.01	67.04	88.55
	AUC	91.00	82.97	68.48	95.33

The classification models performed better when comparing C vs. PT than when comparing C vs. N, C vs. NC or C vs. NNC, although a good performance was observed for all the experiments. In C vs. PT, the metrics that performed better were *Accuracy* = 94.81%, *Sensitivity* = 98.91% for RF, *Precision* = 97.13% for LR, *F* − *score* = 97% for RF and *AUC* = 97.29 for SVM. This experiment better detected positive COVID-19 coughs but did not work so well for classifying pertussis coughs (*Specificity* = 85% for LR and LDA). Instead, in the other experiments, the detection of positive and negative cases was more balanced. This was specially so in the C vs. NNC experiment, which obtained the best *Specificity* = 85.09. This experiment reflects a more real case scenario where COVID-19 coughs co-exist with coughs of different patterns. In the four classification experiments done, RF showed the best overall performance.

4.1. Limitations

Although in general, high performance was obtained in RF, its Specificity was not the optimal. Overall, Specificity outcomes were lower. That means that correctly classifying negative samples is an issue. This must be due to classification mistakes in the dataset. Additional efforts must be made to curate the corpus. Furthermore, further analyses comparing COVID-19 cough patterns with cough patterns from other conditions, such as asthma or bronchitis, are needed.

4.2. Comparison With Prior Work

Other existing works, such as Laguarta et al. [8], extracted MFCCs from cough recordings and input them into a pre-trained CNN. Their model achieved an AUC of 97% with a *Sensitivity* = 98.5% and a *Specificity* of 94.2%. Pahar et al. [12] presented a machine-learning based COVID-19 cough classifier able to discriminate COVID-19 positive coughs from both COVID-19 negative and healthy coughs recorded on a smartphone. They obtained an AUC of 98% using the Resnet50 classifier to discriminate between COVID-19 positive and healthy coughs, while an LSTM classifier was best able to discriminate between COVID-19 positive and COVID-19 negative coughs with an AUC of 94%. Brown et al. [13] used coughs and breathing to understand how discernible COVID-19 sounds are from those in asthma or healthy controls. Their results showed that a simple binary machine-learning classifier are able to classify healthy and COVID-19 sounds correctly. Their models achieved an AUC of above 80% across all tasks.

The RF model used in this paper performed similarly to the ones used by other authors (Accuracy and AUC close to, or above 90% depending on the experiment) although automated cough detection introduced some performance penalty. Additionally, our methodology allows coughs in samples of raw audio recordings to be detected automatically by using the YAMNet deep neuronal network [17]. We also found the set of time–frequency features that could lead to distinguishing COVID-19 coughs from other cough patterns. In addition, the high performance obtained in various sampling sources (UdL, UC, Virufy and Coswara) validates our method as a more generic proposal.

Newer machine-learning works have shown lower results. For example, an accuracy of 85.2% with RF and 70.6% with CNN, were obtained in [14,15] respectively. Recently [16], an accuracy of 90% was obtained with a recurrent neural network (RNN) by using the Coswara dataset. However, the accuracy dropped to 80% with Coswara and Virufy simultaneously. This fact demonstrates that obtaining good outcomes when different datasets are used is a challenge. Our proposal behaved much better even when three additional datasets (UdL, UC and Pertussis) were used.

5. Conclusions

This study demonstrates the feasibility of the automatic detection of COVID-19 from coughs. Excellent results were achieved by fitting an RF model with the set of the time–frequency features selected by RFE for distinguishing COVID-19 coughs. This new methodology presented could lead to automatic identification of COVID-19 by using existing simple and portable devices. It could be the core of a pre-screening mobile app for use as an early response to further COVID-19 outbreaks or other pandemics that may arise in the future.

We will gather more quality data, especially different cough patterns from other conditions, and curate the actual corpus to further train, fine-tune, and improving performance of the models.

CRediT authorship contribution statement

Alberto Tena: Conceptualization, Methodology, Formal analysis, Resources, Software, Data curation, Visualization, Writing - original draft, Validation. **Francesc Clarià:** Conceptualization, Formal analysis, Data curation, Resources, Software, Investigation, Visualization, Validation. **Francesc Solsona:** Writing - review & editing, Supervision, Resources, Project administration, Investigantion, Validation, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Intelligent Energy Europe programme and the Ministerio de Economía y Competitividad under contract TIN2017-84553-C2-2-R and the Ministerio de Ciencia e Innovación under contract PID2020-113614RB-C22. We thank Alfredo Jover and Maria Ramirez from the Functional Unit of Nosocomial Infections, and Cristina Acosta from the Internal Medicine, of the Arnau de Vilanova University Hospital in Lleida, who recorded coughs using the website covid.udl.cat. We also thank the University of Cambridge (in particular, Cecilia Mascolo), the Coswara and Virufy projects, who voluntary share their cough datasets. Finally, we also thank everyone who volunteered their cough.

References

[1] WHO. WHO coronavirus disease (COVID-19) dashboard. https://covid19.who.int. Date accessed: August 10, 2021.

[2] Du. Zhanwei, Abhishek Pandey, Yuan Bai, et al., Comparative cost-effectiveness of SARS-CoV-2 testing strategies in the USA: a modelling study, Lancet Public Health 6 (3) (2021) e184–e191.

[3] Wannian (PRC) Aylward, Bruce (WHO); Liang. Report of the WHO-China Joint Mission on Coronavirus Disease 2019 (COVID-19), 2020.

[4] J. Martinek, M. Tatar, M. Javorka, Distinction between voluntary cough sound and Speech in volunteers by spectral and complexity analysis, J. Physiol. Pharmacol. 59 (SUPPL. 6) (2008) 433–440.

[5] Hanieh Chatrzarrin, Amaya Arcelus, Rafik Goubran, Frank Knoefel, Feature extraction for the differentiation of dry and wet cough sounds, in: MeMeA 2011–2011 IEEE International Symposium on Medical Measurements and Applications, IEEE, Proceedings, 2011, pp. 162–166.

[6] Renard Xaviero Adhi Pramono, Syed Anas Imtiaz, and Esther Rodriguez-Villegas. A cough-based algorithm for automatic diagnosis of pertussis. PLoS ONE, 11(9):1–20, 2016.

[7] Yusuf Amrulloh, Udantha Abeyratne, Vinayak Swarnkar, and Rina Triasih. Cough Sound Analysis for Pneumonia and Asthma Classification in Pediatric Population. Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS, 2015-Octob:127–131, 2015.

[8] J. Laguarta, F. Hueto, B. Subirana, COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings, IEEE Open J. Eng. Med. Biol. 1 (2020) 275–281.

[9] Carnegie Mellon University. COVID Voice Detector. https://cvd.lti. cmu.edu/. Date accessed: August 25, 2021.

[10] Vocalis Health. COVID-19 Study. https://vocalishealth.com/. Date accessed: August 25, 2021.

[11] Ali Imran, Iryna Posokhova, Haneya N Qureshi, Usama Masood, Sajid Riaz, Kamran Ali, Charles N John, and Muhammad Nabeel. AI4COVID-19: AI Enabled Preliminary Diagnosis for COVID-19 from Cough Samples via an App. IEEE Access, pages 1–12, 2020.

[12] Madhurananda Pahar, Marisa Klopper, Robin Warren, and Thomas Niesler. COVID-19 Cough Classification using Machine Learning and Global Smartphone Recordings, 2020.

[13] Chloë Brown, Jagmohan Chauhan, Andreas Grammenos, et al. Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 7 2020.

[14] Jayavrinda Vrindavanam, Raghunandan Srinath, Hari Haran Shankar, Gaurav Nagesh, Machine Learning based COVID-19 Cough Classification Models - A Comparative Analysis, in: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 420–426.

[15] Redacción Médica. Coronavirus: síntomas 'falsos' que nada tienen que ver con el Covid-19, 2020.

[16] Ke Feng, Fengyu He, Jessica Steinmann, and Ilteris Demirkiran. Deep-learning Based Approach to Identify Covid-19. In SoutheastCon 2021, pages 1–4, 2021.

[17] YAMNet. https://github.com/tensorflow/models/tree/master/ research/au dioset/yamnet. Date accessed: August 25 2021.

[18] Alberto Tena. COVID-19 Models and Data repository. https://github.com/atenad/ COVID. Date accessed: August 25, 2021.

[19] Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke Creation of Synthetic Data in R. Journal of Statistical Software, 74(11):1–26, 2016.

[20] University of Cambridge. COVID-19 Sounds App. https://www.covid-19-sounds. org/en/. Date accessed: August 25, 2021.

[21] Indian Institute of Science (IISc) Bangalore. Project Coswara. https://coswara.iisc. ac.in/. Date accessed: August 25, 2021.

[22] Amil Khanzada, Chandan Chaurasia, Nikki Perez, and Lisa Chionis. Virufy. https:// virufy.org/. Date accessed: August 25, 2021, 2020.

[23] Matlab. Audio Toolbox. https://github.com/atenad/COVID. Date accessed: August 25, 2021.

[24] J.F. Gemmeke, D.P.W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R.C. Moore, M. Plakal, M. Ritter, Audio Set: An ontology and human-labeled dataset for audio events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 776–780.

[25] Andrew Howard, Zhu Menglong, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR, abs/1704.0, 2017.

[26] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, Joel S Emer, Efficient Processing of Deep Neural Networks: A Tutorial and Survey, Proc. IEEE 105 (12) (2017) 2295–2329.

[27] X.X. Sun, W. Qu, Comparison between Mean Filter and Median Filter Algorithm in Image Denoising Field, Appl. Mech. Mater. 644–650 (2014) 4112–4116.

[28] Theodoros Giannakopoulos, A Method for Silence Removal and Segmentation of Speech Signals, Implemented in MATLAB, University of Athens, Athens, 2009.

[29] F. Hlawatsch, G.F. Boudreaux-Bartels, Linear and quadratic time-frequency signal representations, IEEE Signal Process. Mag. 9 (2) (1992) 21–67.

[30] Leon Cohen, Time Frequency Analysis: Theory and Applications, Prentice-Hall, 1995.

[31] Francesc Claria, Montserrat Vallverdú, Rafał Baranowski, Lidia Chojnowska, Pere Caminal, Heart rate variability analysis based on time-frequency representation and entropies in hypertrophic cardiomyopathy patients, Physiol. Measure. 29 (3) (2008) 401–416.

[32] Patrick Loughlin. What are the time-frequency moments of a signal? Proceedings of SPIE - The International Society for Optical Engineering, 4474, 2001.

[33] Payam Refaeilzadeh, Lei Tang, Huan Liu, Cross-Validation, in: Encyclopedia of Database Systems, Springer, US, Boston, MA, 2009, pp. 532–538.

[34] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A Training Algorithm for Optimal Margin Classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery.

[35] David Meyer and others. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien, 2019.

[36] Alaa Tharwat. Classification assessment methods. Applied Computing and Informatics, 2018.