Delft University of Technology

BAP

EE3L11

Stress Detection System Using ECG and Respiratory Signals.

Authors:
Isar Meijer (4722930)
Bob Morssink (4722620)

June 19, 2020

Abstract

There are many different studies that try to use physiological features to determine stress. But there exists a lot of uncertainty about which physiological signals and features are the best classifiers and a lot of discrepancies in classification accuracies exist. This study proposes a novel method for the detection of stress. This method contains a two-layered approach to the stress detection problem. Since most features are influenced by speaking, this study suggests that before stress detection takes place a speaking detection algorithm is used. During this study both ECG and respiration data are used to classify stress. The linear influence of the respiration is removed from the ECG data with orthogonal subspace projection to improve the ECG features. An average classification accuracy of 80% is achieved on the test dataset, and a classification accuracy of 77% is achieved on a second dataset which was obtained with a different experimental setup. This shows the real-world applicability and robustness of the designed algorithm. This study, also shows that the influence of speaking on the features is crucial. In literature, a lot of classifiers for stress incorrectly ignore the influence that speaking has on the classification. Combined with a faulty data acquisition method, this possibly results in classifiers trained on detecting speaking instead of stress. With this newly proposed method, the stress detection algorithm is more robust against the influence of speaking.

Preface

This report was written in the context of the Bachelor Graduation Project to obtain the Electrical Engineering Bachelor at Delft University of Technology. We worked with a group of 6 and the main goal of the project was to design a wearable system that can be used to detect stress in subjects throughout the day. Three subgroups of 2 people were formed:

The Pre-Processing subgroup. This subgroup handles filtering and cleanup of the ECG and respiratory signals.

The System Design subgroup. This subgroup designs the complete system, from selecting the sensors that could be used to implement this systems, to the GUI used to display a subjects stress levels.

We are the Stress Detection subgroup. Our task was to detect stress from the ECG and respiratory signals using machine learning.

Almost every morning the 6 of us had a skype meeting at 9am to discuss the progress that each subgroup had made, and what still had to be done. Together as a group we also had to write a business plan and write an ethics essay, and many skype meetings were planned to do all this. We would like to thank the members of the other subgroups for their work, and all the fun skype meetings that we had. Even though the Covid-19 virus changed the situation a lot, we still managed to have a good project.

The cooperation within our subgroup worked very well. We often had long skype meetings while the two of us worked on different parts of the project, sometimes talking some decisions over before continuing again. We had the luck that most of our project was implementing software. So by using github we were able to seemlessly share the progressions we made throughout the project.

We would like to thank dr. Carolina Varon Perez for her continuous help and support throughout the project. We also want to express our sincere gratitude to both dr. Ioan Lager and dr. Carolina Varon Perez for giving us the opportunity to continue the project amid the Covid-19 situation. We would also like to thank prof.dr. Leo de Vreede and dr. Francesco Fioranelli for taking the time to be on the jury for our final assessment.

- Isar Meijer
- Bob Morssink

Contents

Chapter 1

Introduction

1.1 Problem Definition

Stress is an issue that plays a big role in the lives of a lot of people. A lot of research has been done on the exact effects of stress. But there is not a reliable non-invasive way to continuously detect stress levels throughout the day [1]. So there is a definite need for a system that monitors stress, to get better insights into this phenomenon. Two non-invasive methods to detect stress makes use of electrocardiogram (ECG) data and respiratory data. A lot of research has gone into the subject of how to measure stress with an ECG and a respiratory signal [2]. However, to detect stress throughout the day, an ECG and respiratory sensor needs to be worn all the time. A possible implementation is a wearable device that is non-obstructive for the user. For the wearable device to be non-obstructive it needs to be light, which limits the power available on the wearable device. The available power is supplied to an ECG and respiratory sensor, a processor, which performers the processing and stress detection, and communication hardware. In [3] a convolutional deep neural network is used, these types of machine learning algorithms provide stress detection with a high accuracy. But training these types of networks takes a very long time and requires a lot of data. Therefore deep neural networks will not be considered but instead simple machine learning algorithms will be used.

1.2 State of the art analysis

The stress response of the human body is controlled by the autonomous nervous system (ANS), which consists of two branches, the sympathetic nervous system (SNS) and the parasympathetic nervous system (PNS). Under stressful circumstances the SNS is activated and hormones for the stress response are secreted. When the stress has been alleviated, the PNS is activated to decrease the response [4]. A high sympathetic activity thus indicates that a subject is stressed, and a high parasympathetic activity indicates that a subject is not stressed.

A lot of focus in research papers is put on the analysis of the Heart Rate Variability (HRV) and its relation to stress, for example in [1, 5, 6]. The HRV can be used to quantify the sympathetic and the parasympathetic activity, as they both modulate the HRV [6]. The power spectrum of the HRV is used for this task. Two frequency bands are defined; the low frequency band (LF) is defined between 0.04 Hz and 0.15 Hz and is believed to quantify the activity of both the SNS and PNS. The high frequency band (HF) which is defined between 0.15 and 0.4 Hz describes the PNS activity and is mainly driven by the modulations of the respiratory activity on the HRV, called the respiratory sinus arrhythmia (RSA). A high RSA thus indicates low stress levels. The definition of the HF band only works when the respiratory rate falls within this frequency band. In [1] it is proposed to analyze the respiratory rate and to use this to define a more dynamic HF band which is centered around the respiratory rate. This still presents the issue that when a subject is at rest, the respiratory rate will fall below 0.15 Hz and the HF band will overlap with the LF band leading to an overestimation of the sympathetic activity and an underestimation of the parasympathetic activity. Another issue is that when the subject is speaking, the respiratory rate becomes undefined and this frequency differentiation can not be used anymore.

In [5] the RSA is quantified by decomposing the HRV into two components; one respiratory component, denoted $\boldsymbol{Y}_{\boldsymbol{X}}$, describing all variations in the HRV due to respiration and one residual component $\boldsymbol{Y}_{\perp}$. These components are constructed using orthogonal subspace projections and the result of this method is that the analysis of the LF band can now also be performed when the respiratory rate overlaps with the LF band. The decomposition also works for any spectra of the respiratory rate, so this system can also be used while the subject is speaking. This method has not been used in literature for stress. The hypothesis is that this

decomposition is a good method to quantify the RSA, and that when combining this with respiratory features, a good working stress detection system can be made.

1.3 Document Structure

In Figure 1.1 the structure of this thesis can be found. First the ECG pre-processing steps will be discussed in Chapter 3. Next, features will be extracted from the processed data in Chapter 4. Before moving on towards the speaking and stress detection machine learning algorithms, the dataset that is used is discussed based on the information gathered in Chapters 3 and 4, and a significant issue is presented, in Chapter 5. A solution is presented to split the system into two layers; a speaking detection layer and subsequently a stress detection layer. The design of those algorithms is presented in Chapter 6. Finally the complete system is tested on a different dataset in Chapter 7.

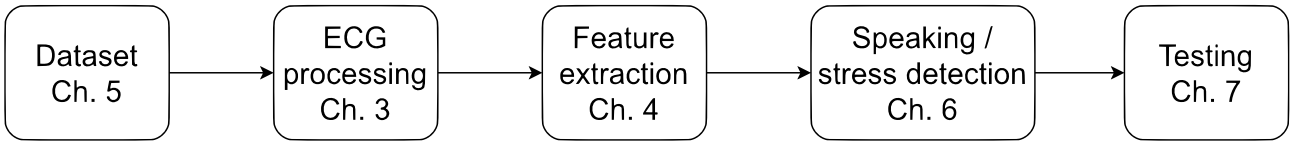

Figure 1.1: Overview of the final system with the corresponding chapters.

Chapter 2

Program of Requirements

This chapter details the program of requirements. The goal of the system that is to be designed is fairly simple; to detect stress, using the ECG and respiratory effort signals. This is what makes up the functional requirements. The non-functional requirements places constraints on how the system will achieve this. These constraints are all measurable against some metric, or are binary constraints. Then there are some trade-off requirements. These requirements are more difficult to exactly quantify, and some trade-off will always exist between them.

2.1 Mandatory requirements

2.1.1 Functional requirements

1. The algorithm should classify stress and stress only.
2. The stress detection should be based on features derived from the heart rate variability and the respiratory signal.

2.1.2 Non-Functional Requirements

1. The system should detect stress with an average accuracy of at least 80%.
2. The system should use short term ($<$ 2 minutes) windows.
3. The computation time of the complete system on a wearable should be faster than the chosen segment length.
4. The system should detect stress with machine learning.

2.2 Trade-off requirements

1. Memory usage should be low enough that implementation on a wearable system is possible.
2. Energy usage should be low enough that integration into some type of wearable is possible.
3. The machine learning algorithm with the highest accuracy will be chosen.

Chapter 3

ECG Processing Steps

As discussed in Section 1.2 the ANS influences the heart, so the HRV can be used to analyze the activity of the ANS. To generate the HRV from the ECG signal, some processing steps have to be done first. A typical ECG has multiple fundamental elements, but the R peak, which is the maximum amplitude peak of a single heart beat, is used to generate the HRV. How the R peaks are detected from the ECG, is described in [7]. The HRV is further analyzed using time frequency analysis tools and features can then be extracted. The steps that are taken before features extraction are described.

3.1 Heart Rate Variability generation

Once the R peaks have been detected, the signal now has to be converted into the HRV signal. After R peak detection a time series of the intervals between subsequent heartbeats is known. As the interval between subsequent heartbeats is varying, this is a non-equidistant sampled signal. To use time frequency analysis tools that are discussed in Section 3.3, an equidistant sampled signal is needed. So the first step is to resample and interpolate the generated R peak signal at 2.56 Hz, resulting in an instantaneous heart rate signal $d_{\mathrm{HR}}(n)$, where n is the sample number. In Section 3.3.3 an explanation for this sampling frequency is given.

This instantaneous heart rate signal $d_{\mathrm{HR}}(n)$ is defined in [1] as

$$d_{\mathrm{HR}}(n) = \frac{1 + \Re(n)}{T(n)}, \tag{3.1}$$

Where $T(n)$ is the mean HR, which is slowly time variant, and $\Re(n)$ is the information from the ANS modulated onto the heart rate, with frequency content > 0.04 Hz, as explained in Section 1.2. Following the methodology in [1] $\Re(n)$ can be extracted.

First $d_{\mathrm{HR}}(n)$ is low pass filtered with a filter with cutoff frequency of 0.03 Hz, which filters out $\Re(n)$, resulting in $d_{\mathrm{HRM}}(n)$:

$$d_{\mathrm{HRM}} = \frac{1}{T(n)} \tag{3.2}$$

This filtering step has been implemented in MATLAB by a 4th order elliptical IIR LPF. The HRV signal, $d_{\mathrm{HRV}}(n)$, and the modulating signal, $\Re(n)$, can then both be calculated:

$$d_{\mathbf{HRV}}(n) = d_{\mathrm{HR}}(n) - d_{\mathrm{HRM}}(n) \tag{3.3}$$

$$\Re(n) = \frac{d_{\mathrm{HRV}}(n)}{d_{\mathrm{HRM}}(n)} \tag{3.4}$$

In Figure 3.1 the time domain and frequency domain representations of d_{HR}, d_{HRM} and $\Re$ can be seen for an ECG segment recorded for a subject during stress. It is clear that the slowly time variant mean HR is filtered out, and the HRV information with frequency content > 0.04 Hz is extracted.

3.2 HRV decomposition

The HRV is modulated by the ANS and the respiration, of which the latter is called the RSA. Both the ANS and RSA are influenced by stress and it is desirable to inspect both of these influences separately. To obtain these

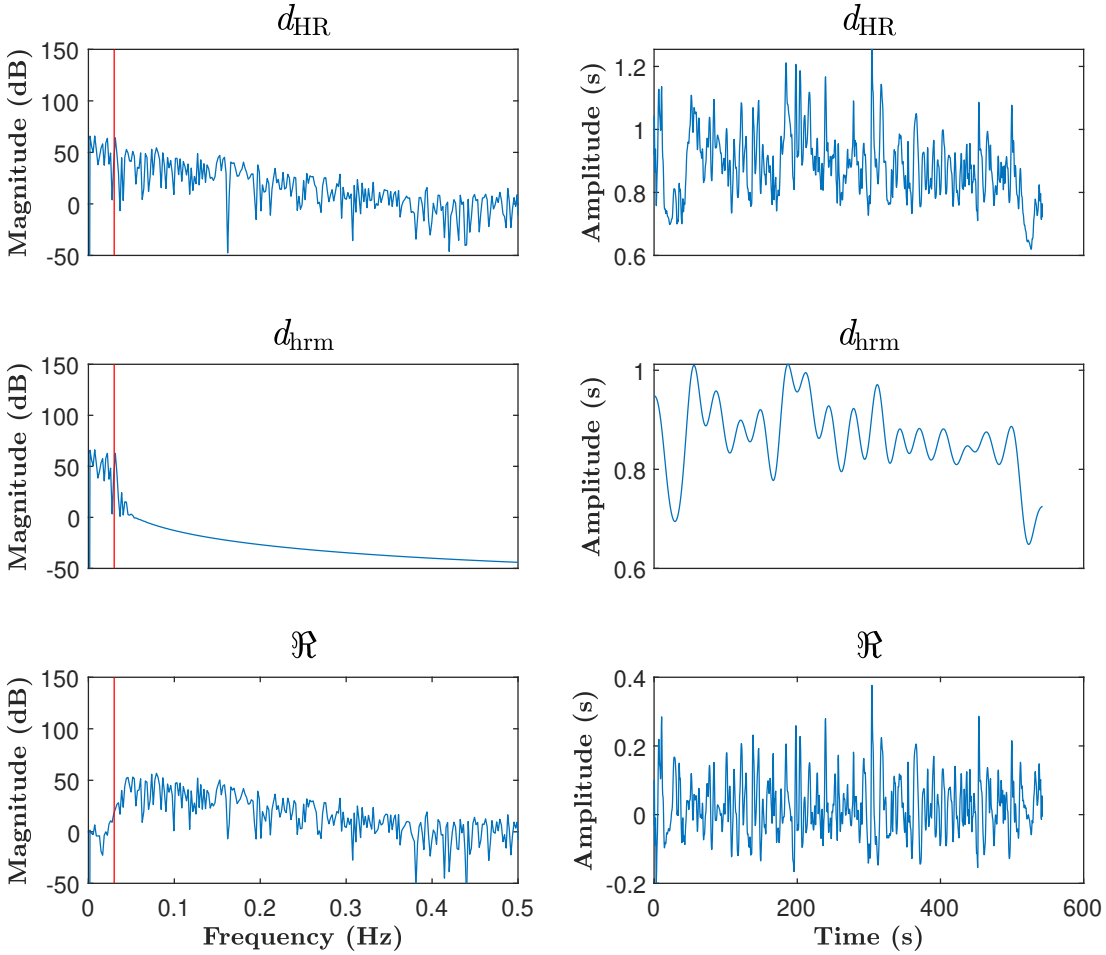

Figure 3.1: (Left) Frequency domain signals of $d_{\mathrm{HR}}(n)$, $d_{\mathrm{HRM}}(n)$ and $\Re(n)$. (Right) Time domain signals of $d_{\mathrm{HR}}(n)$, $d_{\mathrm{HRM}}(n)$ and $\Re(n)$

two signals the HRV is decomposed into two components: a respiratory component and the residual component. The respiratory component describes the linear influence of respiration mechanisms on the HRV signal. Every influence on the HRV signal apart from the respiration, is defined as the residual component. This residual component might still contain nonlinear influences from the respiration, as this decomposition only handles linear relations. The decomposition is done by means of the orthogonal subspace projections (OSP) discussed in [5]. Considering two signals $\mathbf{X}$ and $\mathbf{Y}$ the OSP creates a subspace $\mathbb{V}$ from delayed versions of $\mathbf{X}$. After creating the subspace $\mathbb{V}$ the influence of $\mathbf{X}$ on $\mathbf{Y}$ is obtained by projecting $\mathbf{Y}$ on the subspace $\mathbb{V}$. This subspace is constructed by a matrix $\mathbf{V} = [\mathbf{X_0}, \mathbf{X_1}, ..., \mathbf{X_m}]$, with $\mathbf{X_b} = [x_{\mathrm{b}+1}, x_{\mathrm{b}+2}, ..., x_{\mathrm{N\text{-}m+b}}]$, $b = 0, ..., m$, where m is the model order as defined in [5]. Since the respiration influences the HRV, $\mathbf{Y}$ is the HRV signal and $\mathbf{X}$ is the respiratory signal. The HRV signal is projected onto the respiratory subspace $\mathbb{V}$ by

$$\mathbf{Y_{resp}} = \mathbf{PY} \tag{3.5}$$

where $\mathbf{P}$ is defined as

$$\mathbf{P} = \mathbf{V}(\mathbf{V^T V})^{\mathbf{-1}}\mathbf{V^T} \tag{3.6}$$

After the projection, $\mathbf{Y_{resp}}$ contains the linear influence from the respiration on the HRV. Because $\mathbf{Y_{resp}}$ is linearly related to $\mathbf{Y}$, the residual component $\mathbf{Y_{residual}}$ is defined as $\mathbf{Y_{residual}} = \mathbf{Y} - \mathbf{Y_{resp}}$.

3.3 Time Frequency Analysis

3.3.1 Short Time Fourier Transform

One approach to perform frequency analysis on a signal is by using the Fourier transform. The Fourier transform is intended for stationary signals, and thus results in no time resolution. To gain time resolution, the Fourier transform can be applied to a sliding window; this is also known as the short time Fourier transform (STFT). A drawback of this is that it has fixed resolution; increasing time resolution will decrease frequency resolution and vice versa, but these resolutions are fixed once the window length has been chosen. To get good enough accuracies of the low frequency content of the HRV signal (0.04 - 10.15 Hz), a window of at least 5 minutes is typically necessary [8]. This leads to a time resolution of only 5 minutes for the LF band, but for higher frequencies this window of 5 minutes is not necessary, so additional time resolution is lost. The wavelet transform on the other hand allows for high frequency resolution at low frequencies, and high time resolution at higher frequencies. In [9], the good performance of the wavelet transform on ultra short windows (27s) is shown. The goal of this project is ultra short term analysis, so windows $<$ 2 min (Program Of Requirements). For windows that are this short of length, time frequency analysis tools like the Smoothed Pseudo Wigner Ville distribution (SPWVD) or the wavelet transform are then better options than the STFT.

3.3.2 Wigner Ville Distribution

The Wigner Ville distribution (WVD) takes the fourier transform of the instantaneous autocorrelation function of the signal, and does well to maximimze time and frequency resolution. A disadvantage of the WVD is that when there are multiple frequency components in the signal, cross terms will form. The Smooth Pseudo Wigner Ville distribution tries to minimize these, but still suffers from them. And despite this effect of reduction the method is sensitive to any interference in the frequency bands of interest.

3.3.3 Wavelet Transform

The wavelet transform computes the similarity between a mother wavelet $\psi(t)$ and the signal that is being inspected. The mother wavelet is scaled by a factor a and shifted with a factor b along the signal that is being inspected:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{3.7}$$

If a and b can vary continuously, the transformation is called the continuous wavelet transform. One would then expect that the discrete wavelet transform (DWT) simply discretizes the steps of a and b to make this implementable in computers. This is actually not the case; the DWT makes use of filter banks [10] to decompose the original signal into different levels of coefficients, where the different levels are a measure of scale/detail/frequency. In Figure 3.2 a schematic representation of this is given for 3 levels, $h[n]$ and $g[n]$ are high and low pass filters respectively based on the chosen mother wavelet [10]. The d coefficients are called the detail coefficients and the a coefficient is the approximation coefficient. Due to downsampling by a factor of 2 after each stage, time resolution halves after each stage. Inversely, due to only inspecting half of the previous stages spectrum after each stage, frequency resolution increases. The detail coefficient $d1$ has the most time resolution and represents the highest frequency of the signal. With each lower level (with $a3$ being the last coefficient for 3 levels), each coefficient has a lower time resolution but offers a higher frequency resolution. Figure 3.3 indicates the spectra covered by the coefficients, with 5 levels of decomposition and a sampling frequency of 2.56 Hz. The first coefficients offer high time resolution, but cover a wide spectrum resulting in low frequency resolution. The last coefficients and the approximation coefficient offer low time resolution, but they cover a small spectrum resulting in high frequency resolution.

As mentioned previously, the advantage of the wavelet transform is that it allows for a multi resolution analysis of the HRV signal: for higher frequencies, a higher time resolution is achieved. This means that high frequency features can be updated faster and this leads to the ability to analyze these non-stationary features throughout time; e.g. some statistical features like variance or entropy can now be used.

A disadvantage of using a discrete wavelet transform is that the frequency bands that are obtained are not the same as the typical LF and HF bands, often used to analyze the HRV. Though after using the HRV decomposition described in Section 3.2 the strict separation of these bands is of less importance. This is because the interest in the HF band is mostly due to the modulation of the RSA, which is now contained in $Y_{\mathbf{X}}$, and separated from all other modulators of HRV, which might overlap with the HF band.

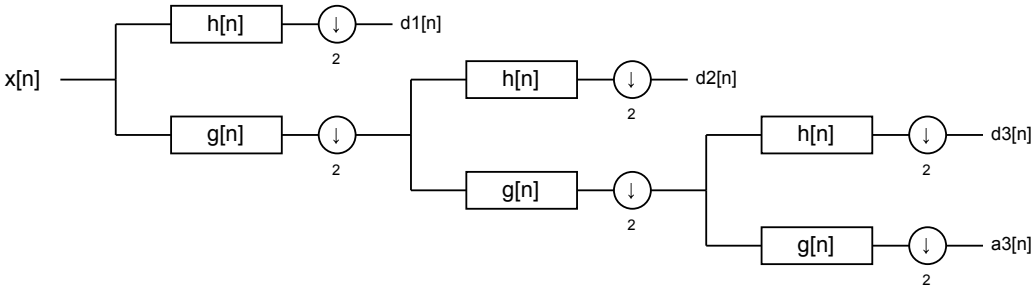

Figure 3.2: Schematic representation of the discrete wavelet transform filterbank for 3 levels.

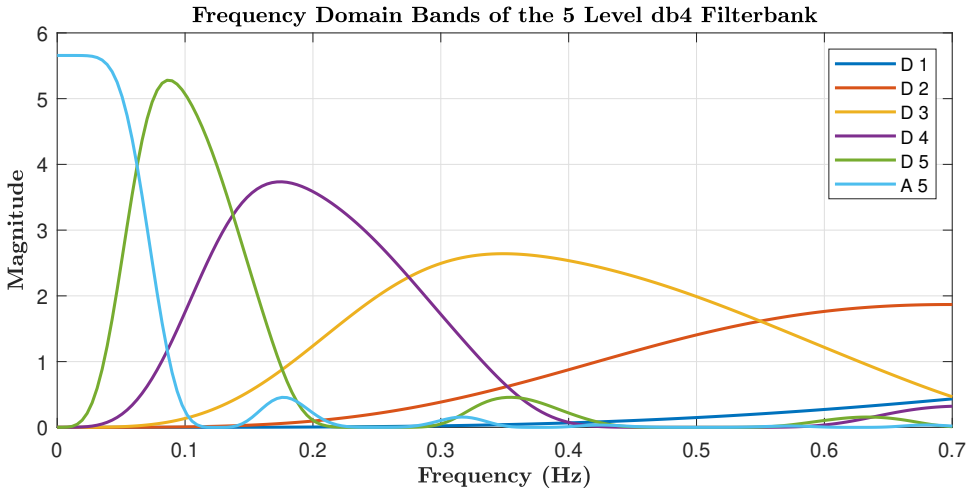

Figure 3.3: Spectra that the different coefficients of the daubechies 4 (db4) wavelet cover, for a sampling frequency of 2.56 Hz and 5 levels of decomposition.

Time complexity of the discrete wavelet transform

If the length of the filters are of constant length L, a single filter operation will take $\mathrm{T(N)} = \mathrm{L} \cdot \mathrm{N}$, where N is the length of the input signal, which leads to the following recurrence relation for the whole filterbank:

$$\mathrm{T(N)} = 2\mathrm{L} \cdot \mathrm{N} + \mathrm{T(N/2)} \tag{3.8}$$

Using a geometric series expansion, the highest order term will still be N, thus leading to O(N) time complexity, compared to the FFT which has O(N log N).

Different discrete wavelet transforms

There are many families of different (discrete) wavelet transforms. The Daubechies family is often used in HRV analysis [11], [12]. High level Daubechie wavelets are smoother, leading to better frequency localization, and the lower levels lead to better time localization. The db4 is a nice trade-off and is also used by other researchers in HRV analyis, for example in [11]. The db4 wavelet is used in this project.

Application of the db4 wavelet

If the signal is sampled at 4Hz, the frequency bands depicted in Table 3.1 can be defined. In Figure 3.3 it can be seen that the bands have significant overlap with each other. Due to the decomposition of the HRV signal, the exact separation of the HF and LF bands is of less importance, as the RSA information is now contained in a different signal. Nevertheless, the orthogonal subspace projection only handles linear relations. Assuming that the non-linear RSA components lie in the HF band, the correct separation of frequency bands is still important. In the right column of Table 3.1 the separation of bands for a sampling frequency of 2.56 Hz can be found. Now, both the HF and LF consist of 2 detail coefficients, namely d2, d3 and d4, d5 respectively. The lower bound of d5 coincides with the traditional LF lower bound, and the bound between LF and HF is only 0.01 Hz off of the traditional bound of 0.15 Hz. This is a definite improvement over the 4 Hz sampling frequency, so 2.56 Hz is used.

Coefficient	Frequency band (Fs = 4 Hz)	Frequency band (Fs = 2.56 Hz)
d1	2.00 - 1.00 Hz	1.28 - 0.64 Hz
d2	1.00 - 0.50 Hz	0.64 - 0.32 Hz
d3	0.50 - 0.25 Hz	0.32 - 0.16 Hz
d4	0.25 - 0.13 Hz	0.16 - 0.08 Hz
d5	0.13 - 0.06 Hz	0.08 - 0.04 Hz
a5	0.06 - 0.00 Hz	0.04 - 0.00 Hz

Table 3.1: Frequency ranges of the wavelet transform

Chapter 4

Feature Extraction and Interpretation

To use machine learning algorithms applied in this report, the data has to be compressed into a set of features. In this chapter, a list of features is proposed, based on features used in literature. Next, some tools that can be used to evaluate the performance of those features are introduced.

4.1 Feature List

In Table 4.1 the feature list that is used is shown. The first 7 features are time domain ECG features that are commonly found in literature [2, 3, 13–16]. The heart rate and RR interval are inversely proportional to each other, which means that using both is redundant. The heart rate was included as a test to check if heart rate values were reasonable. The next 7 features are frequency domain features ($\mathbf{Y}$ denotes the HRV signal). The normalized LF band and sympathovagal balance are defined as

$$\text{LFn} = \frac{\text{LF}}{\text{LF} + \text{HF}} \qquad \text{and} \qquad \mathbf{SB} = \frac{\text{LF}}{\text{HF}} \tag{4.1}$$

As explained in Section 1.2, the HF band is related to the parasympathetic activity, and the LF band is related to both parasympathetic and sympathetic activity. The sympathovagal balance is thus an attempt to quantify the relationship between the sympathetic activity and parasympathetic activity; a high sympathovagal balance indicates low parasympathetic activity and high sympathetic activity, indicating that the subject is experiencing stress, and vice versa for a low sympathovagal balance.

The next 2 sets of 7 features are created from the HRV decomposition from [5]. Feature 29-31 are also created with this decomposition, where the unconstrained version of the sympathovagal balance [5] is defined as

$$\mathbf{SB}_u = \frac{\text{LF}_\perp}{\text{LF}_\mathbf{X} + \text{HF}_\mathbf{X}}, \tag{4.2}$$

And the relative powers are defined as

$$\mathcal{P}_\mathbf{X} = \frac{\mathcal{P}_\mathbf{X}}{\mathcal{P}_\mathbf{X} + \mathcal{P}_\perp} \qquad \text{and} \qquad \mathcal{P}_\perp = \frac{\mathcal{P}_\perp}{\mathcal{P}_\perp + \mathcal{P}_\mathbf{X}} \tag{4.3}$$

Next, different features are extracted from the respiratory signals. The first 6 are the respiratory power in different bands, calculated using the db4 DWT. The upper 4 Hz bound is similar to what the authors of [3] use for their machine learning model. The lower bound of 0.1 Hz is because that is the slowest respiration rate a normal person will have in rest [17]. The final feature is the respiratory rate, which is supplied by the pre-processing group [7].

4.2 Feature analysis tools

4.2.1 Wilcoxon test

Table 4.1 displays possible features that can be extracted from the HRV, respiratory and RR peak signals. However, stress does not influence these features in the same way, some features might change significantly when under stress while others barely change. For every segment from the training set, the values of the features are determined. To determine the influence of stress on these features, the Wilcoxon test is applied

to values of the features obtained from the training set. These values are divided into two datasets: values from the baseline and values from stress stages. The Wilcoxon test checks if the median of the stress dataset differs significantly from the baseline dataset. When the Wilcoxon test is performed on two datasets it returns a p-value, which indicates how much the median of the stressed data differs from the baseline data. A lower p-value indicates a larger difference between the medians, often in literature a p-value below 0.003 indicates that the two medians differ significantly.

4.2.2 Minimum Redundancy Maximum Relevance

The MATLAB function *fscmrmr* ranks a set of features S based on the Minimum Redundancy Maximum Relevance (MRMR) algorithm. Within the MRMR algorithm the relevance (V_{x}) and redundancy (W_{x}) of a feature are defined as:

$$V_{\mathrm{x}} = I(x, y) \qquad \text{and} \qquad W_{\mathrm{x}} = \frac{1}{|S|} \sum_{z \epsilon S} I(x, z) \tag{4.4}$$

where $|S|$ is the number of features in S and I is the mutual information of the random variables X and Y. I is defined as:

$$I(X, Z) = \sum_{i,j} P(X = x_{\mathrm{i}}, Z = z_{\mathrm{j}}) log \frac{P(X = x_{\mathrm{i}}, Z = {}_{\mathrm{j}})}{P(X = x_{\mathrm{i}}) P(Z = {}_{\mathrm{i}})} \tag{4.5}$$

However only the values of the features are known per segment and not the probability with which they occur or the conditional probabilities between the different features. To counter this problem the MATLAB function uses an estimator for the mutual information described in [18]. With V_{x} and W_{x}, the mutual information quotient(MIQ) is defined as:

$$MIQ_x = \frac{V_x}{W_x} \tag{4.6}$$

With these properties, the features are ranked according to the following steps.

1. Find the feature with the largest relevance in S. Add this to the empty set F and remove the feature from S
2. Find features that have a zero redundancy with F and a nonzero relevance. From these features add the feature with the highest relevance to F and remove from S.
   Repeat this step until not a single feature satisfies the conditions then proceed to step 3.
3. For the remaining features in S find the largest MIQ_x value with nonzero relevance and nonzero redundancy and add the particular feature to F.
4. Once the remaining features in S have zero relevance add them in random order to F.

The full MRMR is described in more detail in [19]. This paper also shows the importance of reducing the redundancy in a feature set. In the paper, feature sets that used the MRMR algorithm to rank their features and reduce the dimension of the feature set, outperformed feature sets containing every feature from their feature sets.

Nr.	Features	Description	Category
1	MeanRR	Mean of the RR interval	Time Domain
2	SDRR	Standard deviation of the RR interval	
3	RMSSD	Root mean square of the differences between successive RR intervals	
4	pNN50	The number of pairs of successive RR intervals that differ by more than 50 ms, divided by total number of RR intervals.	
5	pNN20	Similar to pNN50, only with a 20ms threshold	
6	MeanHR	Mean heart rate	
7	SDHR	Standard deviation of the heart rate	
8	$\mathbf{P_Y}$	Total power of $\mathbf{Y}$	Frequency Domain
9	$\mathbf{P}_{LF}$	Power in the low frequency band (d5 + d4) of $\mathbf{Y}$	
10	$\mathbf{P}_{HF}$	Power in the high frequency band (d3 + d2) of $\mathbf{Y}$	
11	$\mathbf{P}_{LFn}$	Normalized power in the low frequency band of $\mathbf{Y}$	
12	**SB**	Sympathovagal balance of $\mathbf{Y}$	
13	σ_{LF}	Standard deviation of the LF band of $\mathbf{Y}$	
14	σ_{HF}	Standard deviation of the LF band of $\mathbf{Y}$	
15	$\mathbf{P_X}$	Total power of $\mathbf{Y_X}$	Respiratory Component (OSP)
16	$\mathbf{P}_{\mathbf{X}\|LF}$	Power in the low frequency band (d5 + d4) of $\mathbf{Y_X}$	
17	$\mathbf{P}_{\mathbf{X}\|HF}$	Power in the high frequency band (d3 + d2) of $\mathbf{Y_X}$	
18	$\mathbf{P}_{\mathbf{X}\|LFn}$	Normalized power in the low frequency band (d5 + d4) of $\mathbf{Y_X}$	
19	$\mathbf{SB_X}$	Sympathovagal balance of $\mathbf{Y_X}$	
20	$\sigma_{\mathbf{X}\|LF}$	Standard deviation of the LF band of $\mathbf{Y_X}$	
21	$\sigma_{\mathbf{X}\|HF}$	Standard deviation of the HF band of $\mathbf{Y_X}$	
22	$\mathbf{P}_\perp$	Total power of $\mathbf{Y}_\perp$	Residual Component (OSP)
23	$\mathbf{P}_{\perp\|LF}$	Power in the low frequency band (d5 + d4) of $\mathbf{Y}_\perp$	
24	$\mathbf{P}_{\perp\|HF}$	Power in the high frequency band (d3 + d2) of $\mathbf{Y}_\perp$	
25	$\mathbf{P}_{\perp\|LFn}$	Normalized power in the low frequency band (d5 + d4) of $\mathbf{Y}_\perp$	
26	$\mathbf{SB}_\perp$	Sympathovagal balance of $\mathbf{Y}_\perp$	
27	$\sigma_{\perp\|LF}$	Standard deviation of the LF band of $\mathbf{Y}_\perp$	
28	$\sigma_{\perp\|HF}$	Standard deviation of the HF band of $\mathbf{Y}_\perp$	
29	$\mathcal{P}_\mathbf{X}$	Relative respiratory HRV power	OSP Relative
30	$\mathcal{P}_\perp$	Relative residual HRV power	
31	$\mathbf{SB}_u$	Unconstrained version of the sympathovagal balance [5]	
32	$\mathbf{P}_{\mathbf{R}\,p1}$	Power in the 1.95 Hz - 3.91 Hz frequency band of $\mathbf{R}$	Respiratory Features
33	$\mathbf{P}_{\mathbf{R}\,p2}$	Power in the 0.98 Hz - 1.95 Hz frequency band of $\mathbf{R}$	
34	$\mathbf{P}_{\mathbf{R}\,p3}$	Power in the 0.49 Hz - 0.98 Hz frequency band of $\mathbf{R}$	
35	$\mathbf{P}_{\mathbf{R}\,p4}$	Power in the 0.24 Hz - 0.49 Hz frequency band of $\mathbf{R}$	
36	$\mathbf{P}_{\mathbf{R}\,p5}$	Power in the 0.12 Hz - 0.24 Hz frequency band of $\mathbf{R}$	
37	$\mathbf{P}_{\mathbf{R}\,p6}$	Power in the 0.06 Hz - 0.12 Hz frequency band of $\mathbf{R}$	
38	$\mathbf{P}_{\mathbf{R}\,hl}$	$(\mathbf{P}_{\mathbf{R}\,p1} + \mathbf{P}_{\mathbf{R}\,p2} + \mathbf{P}_{\mathbf{R}\,p3})/(\mathbf{P}_{\mathbf{R}\,p4} + \mathbf{P}_{\mathbf{R}\,p5} + \mathbf{P}_{\mathbf{R}\,p6})$	
39	$\mathbf{P}_{\mathbf{R}\,tot}$	Sum of the 6 power bands defined above	
40	$\mathbf{P}_{\mathbf{R}\,peak}$	The maximum value of all the detail coefficients of the 6 frequency bands defined above.	
41	pfRESP	Peak factor of the respiratory signal: $\mathbf{P}_{\mathbf{R}\,peak}/\mathbf{P}_{\mathbf{R}\,tot}$	
42	pf2RESP	Similar to pfRESP, only this time the peak and total power are calculated with the FFT.	
43	RESPRATE	Respiratory rate.	

Table 4.1: Features selection

Chapter 5

Data Handling

Before moving on and selecting the features that best classify the data, it is important to take a step back and analyze what it is exactly that those features quantify:

A problem with detecting stress is that it is difficult to make sure that it is actually stress that is detected, and not something else, e.g. movement or speech. To ensure that the algorithm is trained on detecting stress and stress only, it is crucial to completely understand the dataset that will be used to train the machine learning algorithm. First, the dataset and the data collection protocol will be described. Second, possible issues with this data collection will be discussed (and improvements will be selected, if so).

5.1 Dataset Analysis

The dataset was collected at the University of Zaragoza and the Autonomous University of Barcelona, Spain, and both Ethics Committees approved the protocol for data collection. The ECG and the respiratory effort were recorded from 46 volunteers, of which 18 were male and 28 were female (mean age of 21.76 $\pm$ 4.48 years). The sampling frequency of the ECG and respiration were 1 kHz and 250 Hz respectively. Stress was induced by means of a modified Trier Social Stress Test [1], which consists of the following phases:

- Baseline (BL). The subjects listen to relaxing audio for about 10 minutes and are asked to completely relax.
- Storytelling (ST). Three stories are told to the subjects with a lot of detail. The subjects are asked to remember as many details as possible.
- Memory task (MT). The subjects are asked to recite the previously told stories in front of a camera, with 30s for each story.
- Stress anticipation (SA). The subjects are asked to wait for the evaluation of the MT stage, which will take about 10 minutes.
- Video Exposition (VE). For each story told in the ST stage two videos will be shown: first of an actor, repeating the story perfectly. The goal is to make the subjects believe that it is normal to do this task perfectly. Next, the video of the subject telling the story is displayed.
- Arithmetic task (AT). The subject has to count down from 1022 in steps of 13. Whenever a mistake is made the subject has to start over from 1022, and the subject is told that there is a 5 minutes time limit. No subject completed the countdown.

The first stage, BL, is classified as a relaxed state, the other states are classified as stressed states. This was confirmed by multiple psychometric scores [1]. Of the 46 volunteers, 12 were excluded as some stages were corrupted by technical artefacts.

This test seems like a good way to induce stress, which is confirmed by the psychometric evaluation. But there is one significant issue. In some of the stressful states the subjects are speaking, but there is no baseline for which this is the case, as illustrated in Figure 5.1. This leads to the following statement:

If a subject is speaking, the subject is always in a stressed stage.

If then the best features are simply selected by means of the Wilcoxon test for example, it is almost guaranteed that the algorithm is not only detecting stress, but also detecting if someone is talking or not. This conflicts

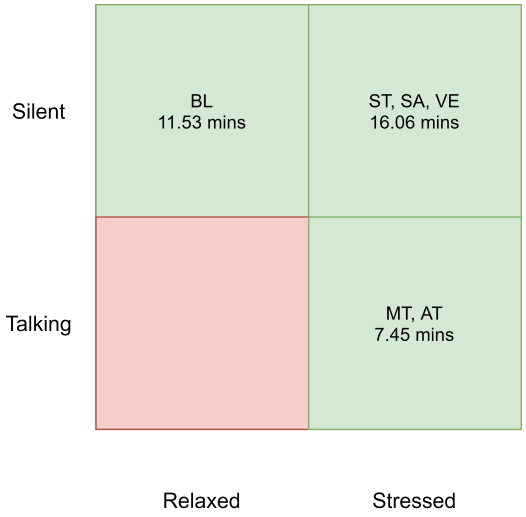

Figure 5.1: Illustration of the lack of speaking baseline, with the average time in minutes per person.

with the Program of Requirements defined in Section 2. In Figure 5.2 an example of a feature that classifies talking instead of stress can be seen. The silent stress state shows no significant (Wilcoxon $p < 0.003$) difference with the baseline, but the talking stress state does, leading to an overall significant difference between stressed and relaxed (as can be seen in Table 5.1). So if this feature is selected based on the fact that there is a significant ($p < 0.003$) difference between the stress stages and the baseline, the algorithm will not only be trained on stress but also on whether someone is speaking. A solution to this problem could be to only select a feature if there is a significant difference with respect to baseline for all three cases: silent stress, talking stress and total stress. But this can still run into the same problem, as depicted in Figure 5.3. It is clear (also confirmed by the Wilcoxon test with $p < 0.003$) that when the subject is not talking and stressed, there is a significant difference with respect to baseline. But it can also be seen that when the subject is talking, this difference is even bigger. This poses a dilemma; keep the feature which will lead to an artificially boosted classification accuracy, or remove this feature, at the cost of losing information related to stress.

Silent stress	**Talking stress**	**Total stress**
0.287	0.000	0.000

Table 5.1: P-values with respect to baseline for the mean heart rate (corresponding to Figure 5.2)

.

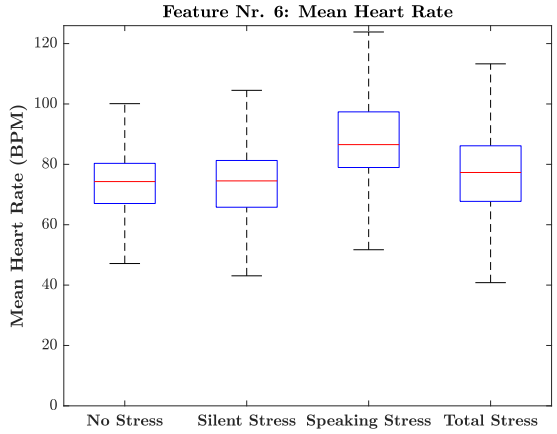

Figure 5.2: Boxplot of the mean heart rate

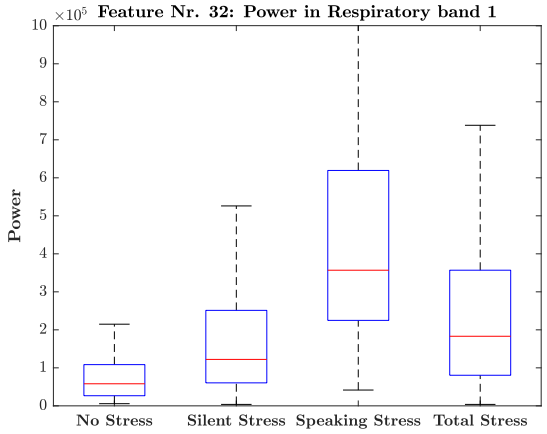

Figure 5.3: Boxplot of the respiratory feature $\mathbf{P_R}_{p1}$

5.2 Two Layer Approach

Due to this dilemma, the choice was made to split the classification into two layers, to make sure that stress and only stress is detected (Program of Requirements): The first layer will detect if the subject is talking or not. If the subject is not talking, the second layer will determine if the subject is stressed or not. The result of using this two layered approach is that now the classification accuracy of the second (stress detection) layer is much more representative of the algorithms ability to classify stress. Of course this is only for silent states, but future research could focus on speaking states as well.

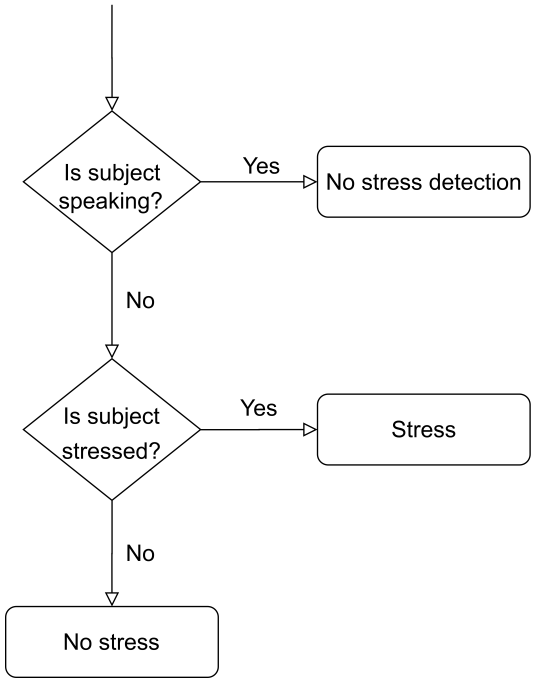

Figure 5.4: Schematic representation of the two layer approach.

5.2.1 Test and training data.

The dataset that is used consists of 34 subjects. Before any feature analysis was done, the dataset was split up into a test and training set. For the test set, 5 subjects were randomly selected and excluded from the training set.

Chapter 6

Stress and Speaking Detection

As the main purpose of this project is stress detection, this is designed first. After that, the speaking detection algorithm is designed.

6.1 Segment length selection

First the segment length on which the final algorithm will be trained is a parameter that has to be selected. In literature many studies use 5 minute windows [3]. This is because the lowest frequency content of the HRV signal is 0.04 Hz, which means that a single period is 25 seconds, and if the STFT is used, windows of 5 minutes are needed to capture LF information accurately [20]. In [9], the good performance of the wavelet transform on ultra short (27s) windows with HRV analysis is shown. As the wavelet transform is used in this project, shorter segment lengths thus may be viable. The advantage of a longer segment length is better preservation of the LF content and lower variance of all the features in general, the disadvantage is twofold: Firstly, the longer segment leads to an increase in stress detection latency. The second disadvantage is that there will be less data to train on, as there are now fewer segments in total. To select the optimal segment length, some analysis on the features for different segment lengths had to be done.

First, the MT and AT stages (in which people were speaking) were excluded as to only analyze the effect of changing the segment length on stress detection. Next, the dataset was segmented using different segment lengths; ranging from 20 to 120 seconds with 10 seconds interval, leading to 11 different sets. For each of those sets, features were calculated for all the segments.

Then, the Wilcoxon signed rank test was performed on all of those sets, testing the difference between relaxed and stressed states. The result of this can be seen in Figure 6.1. Each cell represents the p-value of a feature for a certain segment length. The base 10 log has been taken of the inverse of the p-value for improved readability. Note that $-log_{10}(0.003) \approx 2.5$. A few things can be observed in Figure 6.1: First a gradient from left to right, indicating that features generally have lower p-values for smaller segment lengths. This is due to the fact that there is much more data for shorter segment lengths (as more segments can be formed), leading to a lower p-value. This gradient is the most evident in the respiratory features (feature nr. 32-43) and seems to mostly disappear for ECG features. This can be explained by the previously mentioned disadvantage of using shorter segments; the loss of accurate frequency information for LF components, because only a few wavelengths of the low frequencies can fit in the shorter segments. An interesting observation is that the HF band of $\mathbf{Y_X}$ (feature nr. 17) clearly suffers from this effect as well.

Because of this gradient, it becomes difficult to see which segment length would be optimal, so more information is needed.

Thus the next analysis method is the Minimum Redundancy Maximum Relevance (MRMR) algorithm. This was implemented by using the *fscmrmr* MATLAB function. As explained in Section 4.2.2, this function not only checks the relevance of the features, but penalizes features that have high correlation with other features. In Figure A.1, which is included in the appendix, the result of this test can be seen. As the MRMR algorithm also accounts for correlation among the features, the best performing features change very abruptly among different segment lengths. While the MRMR algorithm is a good method to select the best features from a feature-set for one segment length, it is still not immediately clear how the different segment lengths compare with each other. More analysis is necessary.

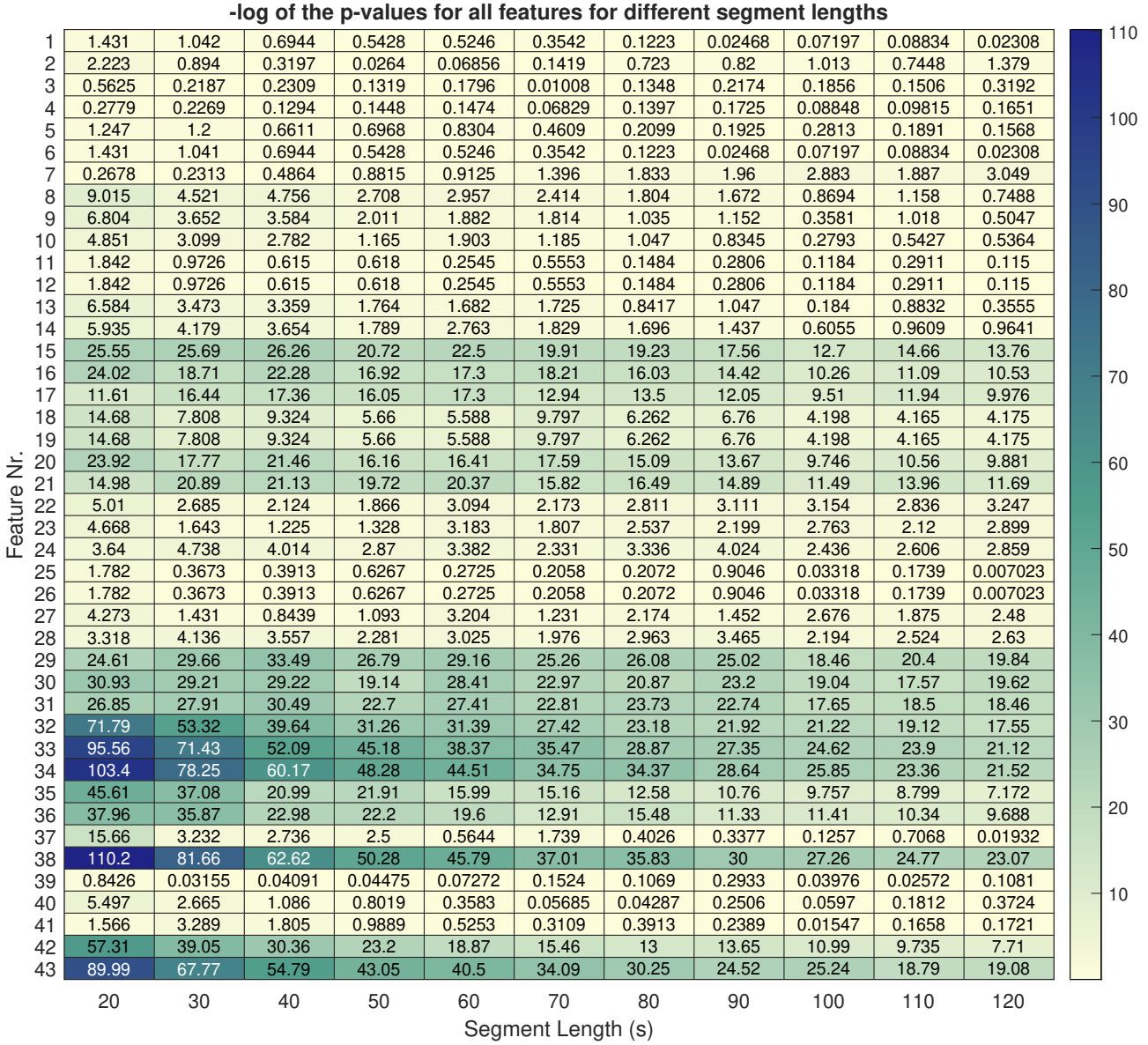

Figure 6.1: Base 10 logarithm of the inverse of the p-value for each feature for each segment length.

To compare different segment lengths the following methodology was used: First, a pre-selection was done using the Wilcoxon test; all features with a p-value > 0.003 were discarded. Next the 10 best features were selected by means of the MRMR algorithm. These features were then used to train a Random Forest (called Bagged Trees in MATLAB) machine learning algorithm. The reason why Random Forests were used is explained next.

6.2 Random Forests

Random Forests were first proposed by Breiman in [21]. Random Forests consists of multiple decision trees; each of those trees is trained on a bootstrapped data-set and each node split uses only a limited number of features. A bootstrapped data-set is made by taking L random samples of the original dataset, where L is the length of the original dataset. This leads to certain observations appearing more than once, and about a third of the observations being left out of the bootstrapped data-set.

This inherent randomness of Random Forests protects against overfitting, which is something that often occurs in single decision trees. If enough trees are used to build the Random Forest, the accuracy starts to converge. These two factors combined makes Random Forests a suitable algorithm for the evaluation of different segment lengths.

For each segment length the 10 best features were used to train a Random Forest. The number of learners was

set at 500 and the maximum number of splits was set at 256, in Section 6.3 further analysis of these parameters is performed. Figure 6.2 shows the results of this evaluation. On the Y-axis the average classification is plotted. This has been calculated by taking the average of the specifity and sensitivity. Both the specifity and sensitivity were within 3% of the average classification accuracy. In the plot it is clear that for shorter segment lengths, the accuracy starts dropping. This can be explained by the fact that these segments are not long enough to capture the relevant information that is used for some features, like for instance the LF content. And the drop in accuracy for the longest segments might be due to the reduced number of data points to train. The optimal segment length then appears to be 80 seconds, which is the segment length that was chosen. A segment length of 100 seconds may have a bit higher accuracy, but it was determined that this small possible gain does not warrant a 25% segment length increase. A segment length of 80 seconds also achieves an accuracy $> 80\%$ which is in line with the Program of Requirements defined in Section 2.

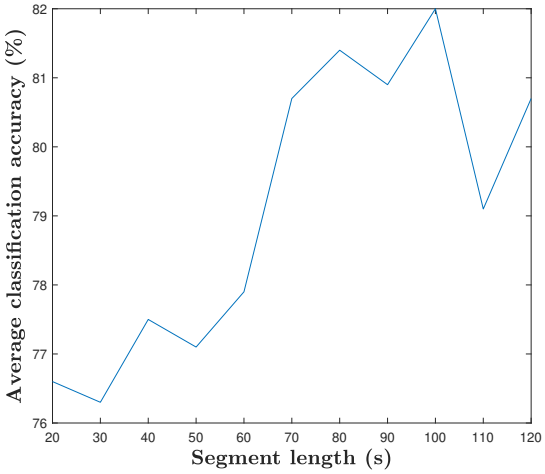

Figure 6.2: Comparison of the average classification accuracy for different segment lengths.

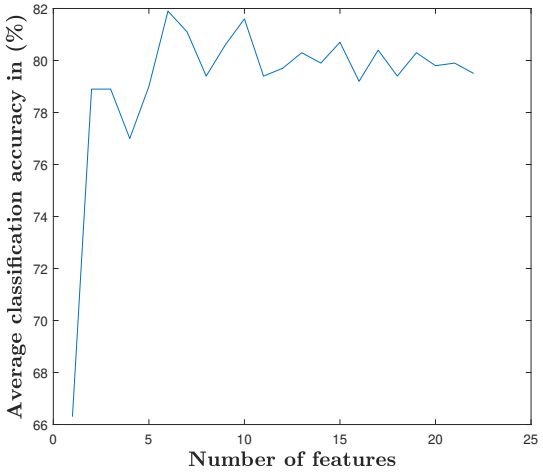

Figure 6.3: Comparison of the classification accuracy for different number of features.

6.3 Machine Learning Optimization

The goal is to keep the machine learning model simple to ensure low computation times when using the model to determine stress. With the bagged trees algorithm there are four parameters that can be optimised, namely: number of features, maximum number of splits, number of learners and the misclassification cost matrix. For the misclassification cost matrix a setting is found where the true positive and true negative percentages are within 2% of each other, which results in a balance between sensitivity and specificity. To obtain comparable results this balancing of the sensitivity and specificity will be done for every iteration. In order to reduce the variance of the accuracy from the trained machine learning models, 250 learners are used to determine the number of features and maximum number of splits. With 250 learners the final prediction of the random forest might be more consistent compared to 30 learners. The number 250 is chosen since this should be well above the number of learners for which the accuracy converges.

6.3.1 Optimization of the number of features

A rule of thumb for the maximum number of features is given in [22]. The rule states that for every feature, there should be ten observations in the smallest class. When using segments of 80 seconds, the baseline is the smallest class and has 229 observations. Thus a maximum of 22 features can be used. However, the goal is to find the model with the simplest parameters and thus the least number of features needed. To obtain the minimum number of features leading to maximum accuracy, the accuracy is determined for 1 up until 22 features. As described in Section 4.2.2, the features can be ranked by mean of the MRMR algorithm. This ranked list will be used when evaluating the effect of the number of features. If six features are used by the machine learning model, the first six features from the ranked list are taken. In Figure 6.3, 250 learners and 256 maximum number of splits are used. In the figure it can be seen that the accuracy converges after about

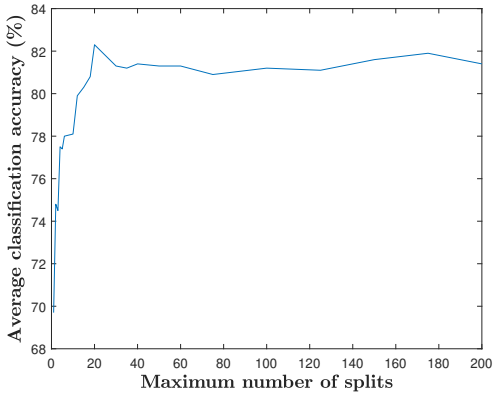

Figure 6.4: Comparison of the classification accuracy for different numbers of maximum splits.

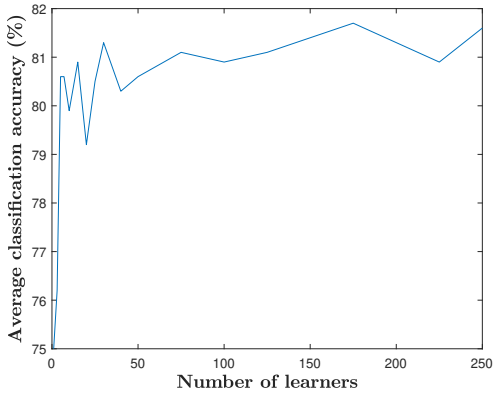

Figure 6.5: Comparison of the classification accuracy for different number of learners.

six features. To keep the machine learning model as simple as possible, six features are used. From now on, every optimization step will be performed with the six selected features.

Besides the convergence of the accuracy, the figure also shows the variance of accuracy once the convergence level has been reached. This could be explained by the fact that random forests are used where the trees created are of random nature and thus the accuracy varies per iteration.

6.3.2 Optimization of the maximum number of splits

Optimizing the maximum number of splits is done by varying the maximum number of splits from 1 until 200. Furthermore the number of features used is six and the number of learners is 250. By varying the maximum number splits the accuracy is influenced as seen in Figure 6.4. From the figure, it becomes clear that the accuracy converges at a low amount of maximum splits. After 20 maximum splits, it clear that the model has reached convergence since the variations are within 1%. Again, to obtain a simple machine learning model, the maximum number of splits is set to 20. This value will be used in the optimization to determine the number of learners.

6.3.3 Optimization of the number of learners

The number of learners specifies how many decision trees are made within the bagged tree model. When more decision trees are generated more results are computed, which benefits the final result. Therefore the number of learners is varied between one and 250 to see where the accuracy converges, resulting in Figure 6.5. Furthermore the number of features used is six and the maximum number of splits is equal to 20. Just as with the previous optimization parameters, the accuracy converges with a low number of learners, in this case with about 75 learners. Thus the number of learners is set to 75 to keep the machine learning model as simple as possible.

6.3.4 Final parameters

Looking at the above sections, the final parameters for the bagged trees model are: 6 features, a maximum of 20 splits and 75 learners. The misclassification cost for predicting stress in the baseline stage is set to 1.4, which balances the sensitivity and specificity to be within 1% of each other. With these parameters, the trained model is tested on the test set.

6.4 Testing the stress detection model

With the parameters of the machine learning model finalized, a model is exported. This model is tested on the five people that were excluded from the training set. The data of the test set is segmented into segments of 80 seconds and processed in the same way as explained in Chapter 3.

In Table 6.1 the accuracies of the model can be found. The accuracies are close to what is expected after looking at validation results, but there is a very high variance; the results differ a lot between subjects.

A possible cause for this high variance is that some of the features that are used are measured absolutely instead of relatively. For example, the respiratory rate may change when someone is stressed, but different

Model	Baseline Accuracy	Stress Detection Accuracy	Average Classification Accuracy
First Model	83 ± 25 %	76 ± 26 %	80 ± 9%
Second Model (Normalized Features)	75 ± 37 %	86 ± 24 %	80 ± 17 %

Table 6.1: Accuracies of the two different machine learning models.

subjects have different respiratory rates as well. So when making a general machine learning model, if someone breathes faster or slower than the training population mean, this immediately introduces a bias that is not connected to that subject's stress levels. A more relative feature for example is feature nr. 30; the relative power in the residual HRV component, denoted $\mathcal{P}_{\perp}$. The power in this component is calculated relatively to the total HRV power, making it less susceptible to biases.

As a first step, some features from Table 4.1 were improved. Namely the power in the respiratory bands (feature nr. 32-37) were normalized with respect to the total power in the respiratory bands. Next, a second model was trained using only features that were measured relatively or that were normalized in some way (the respiratory rate was excluded for example). The same optimization steps as described previously in Section 6.3. were used. The final algorithm used 25 maximum splits, 125 learners and 6 features. The features that met the normalization criteria and that were used after optimization will be analyzed in section 8.2. The figures that were generated for optimizing the parameters can be found in Appendix B.

This algorithm was tested on the same test dataset, the results of which are also listed in Table 6.1. It is clear that nothing much has changed, even though exclusively normalized features have been used. This leads to the question as to which extent human physiological features can be normalized.

6.4.1 Individuality of physiological features

In [23] the individuality of breathing patterns is discussed. The author establishes that breathing patterns are highly individual, not only in terms of features like tidal volume and respiratory frequency but also in terms of airflow shape. This means that even after normalizing respiratory features with respect to total power, the individuality of breathing patterns makes these features still very prone to personal biases. The article does state that these individual respiratory patterns are time persistent; a subject's breathing pattern barely changes throughout time. This means that a personalized approach might be able to work very well.

All of the ECG/HRV features that perform well (see Figure 6.1), use the HRV decomposition discussed in Section 3.2. This means that all of those features are influenced by the respiratory effort and the individuality that comes along with it. On top of that, in [24] it is concluded that differences also exist between individuals in how the RSA control system responds to different breathing periods. So not only are those HRV features influenced by the individuality of the respiration, but also by the system that modulates the respiration.

6.4.2 Personalized model

To account for these intersubject differences and reduce the variance for detection some sort of personalized model should be used for the best results. One simple method is to subtract the median of a subject's baseline with the median of the test set baseline. This shift can then be used for all the subjects segments. This is the simplest way to account for biases due the individual nature of the features. This method was tested, but did not result in any improvements.

6.5 Speaking Detection

	Silent	Speaking	Average Classification
Speaking Detection	98 %	100 %	99 %

Table 6.2: Accuracy of the speaking detection model on the test dataset.

The speaking detection layer was optimized using the same approach as in Section 6.3. The best features were chosen from the updated feature list with the normalized respiratory features. The Figures that were generated for optimizing the parameters are listed in Appendix C. The final model used 5 features, 50 maximum number of splits and 75 learners.

The 5 features that were used are listed in Table 6.3.

Feature Nr.	Feature
37	$\mathbf{P}_{\mathbf{R}\,p6}$
6	MeanHR
38	$\mathbf{P}_{\mathbf{R}\,hl}$
41	pfRESP
22	$\mathbf{P}_{\perp}$

Table 6.3: Features used for the speaking detection alogrithm.

The results of testing the model on the test set are listed in Table 6.2. As can be seen from the Figures in Appendix C, the validation accuracy was around 85%. The sudden increase in performance on the test set is thus notable. This can again be explained by the individuality of features, leading to a high variance in average classification accuracy per subject.

Chapter 7

Testing on a Seperate Dataset

To really evaluate how well the algorithm performs in the real world, it is also tested on a completely different dataset; the Drivers dataset.

7.1 Drivers dataset

This dataset was generated by letting subjects drive around Boston. Subjects would spend the first 15 minutes and last 15 minutes with their eyes closed in a garage, which serves as the baseline. The time in between these phases, was spent driving around, which took approximately 50 to 90 minutes. A full description of the dataset is given in [25].

The dataset consists of 16 subjects. For each subject, the data was divided into the different parts of the drive: the pre-drive period, the driving period and the post drive period. For each of these parts, the data was segmented into 80s segments and the preprocessing steps, speaking and stress detection were performed. The speaking detection results are listed in Table 7.1.

	Pre-Drive Speaking	**Driving Speaking**	**After Drive Speaking**
Speaking detection	$3.4 \pm 13.6\%$	$6.11 \pm 7.7\%$	$4.3 \pm 12.3\%$

Table 7.1: Speaking percentage per stage as detected by the algorithm.

This is in line with the expectation that people were either barely speaking or not speaking at all. As in the original paper describing the dataset [25], it is mentioned for the baselines that people sat with their eyes closed, which probably means no speaking. And while driving, there was another person in the car monitoring everything, but doing so from the back seat as to not disturb the driver. This again indicates that there was probably barely any speaking. Of the 927 segments, there were 49 segments classified as speaking. Of those 49 segments, 15 belonged to subject number 1. Also in the first baseline only subject number 1 was classified to have been speaking, and in the second baseline only subject number 1 and 15 were classified to have been speaking. This causes the high standard deviations in Table 7.1. Either these subjects were speaking a lot, or this is a result of the individuality of features, as discussed in 6.4.1.

Algorithm	**Pre-Drive**	**Driving**	**After Drive**
All Features with Normal Respiration	$94 \pm 11\%$	$32 \pm 21\%$	$91 \pm 15\%$
Normalized features only	$77 \pm 27\%$	$76 \pm 24\%$	$63 \pm 30\%$

Table 7.2: Accuracies of the two machine learning algorithms on the Drivers dataset.

Next the stress detection was performed on the segments in which subjects were not speaking. The classification accuracies for stress detection are listed in Table 7.2. The first machine learning model that uses non-normalized respiration features and ECG features is very inaccurate. There is a clear bias towards rest detection. This is because the power in the respiratory bands are not normalized, which means that these features are influenced by the use of different sensors. This is confirmed when inspecting the data; the power in the respiratory bands of the drivers dataset is more than a 1000 times higher than the power in the respiratory bands of the training set.

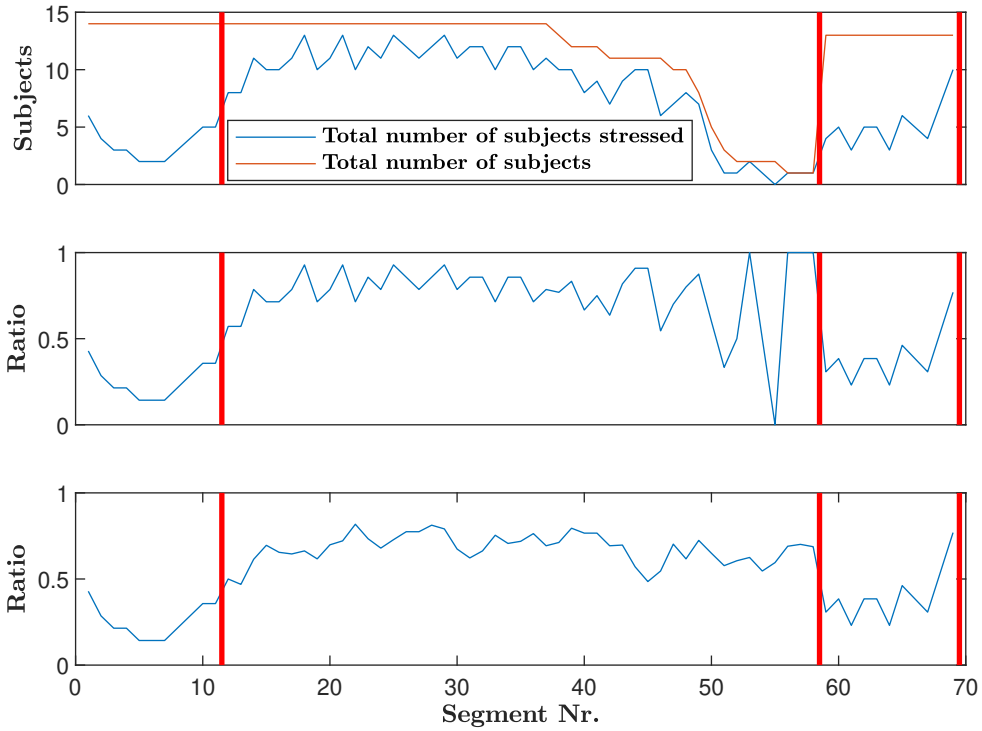

Figure 7.1: Progression of subject's stress levels throughout driving. Top Figure shows number of subjects with that segment and number of subjects stressed. Middle Figure shows the ratio of stressed over total number of subjects. Bottom Figure shows the same as the middle Figure, but all the driving parts are interpolated to be the same length for all subjects.

The second machine learning model that exclusively uses normalized features however, performs very well, obtaining similar accuracies to the test set. One thing to note is the lower after drive accuracy. This can be explained by people still being somewhat stressed after a long drive.

In Figure 7.1 the progression of stress levels throughout the experiment is depicted. The different stages are separated by the red lines. Because the duration of the driving part varies, the number of subjects that have a certain segment starts to decrease at the end of the driving part. This is depicted by the orange line in the top Figure in 7.1. The blue line shows the number of subjects that are stressed. The middle Figure shows the ratio of the two. As the number of people decreases near the end of the driving stage, this ratio becomes highly variant. To overcome this, the stress/no stress vectors that were generated for each subject for the driving part, were interpolated to the same length. The average of these vectors is plotted in the bottom Figure. Two subjects had a few segments of which the sensor data was corrupted, so these subjects were not used to generate this Figure.

This Figure really shows that the system works. In the first baseline, the lowest stress levels are right in the middle of the 15 minute segment; showing that people are a bit more nervous at the start, as all the sensors have just been connected. Towards the end of the first baseline, people could get more stressed in anticipation of the test starting. Then once the driving has started, stress levels rise up considerably, and stay consistently high until the end. Once the subjects are back in the garage, stress levels immediately drop back down again. But the mean stress levels in the second baseline still exceed the ones in the first baseline, possibly due to the body still being in a stressed state because of the stressful driving. The higher variance and the peak at then end can perhaps also be attributed to this fact, though the latter can also just be a result of the high variance of the system in general.

7.2 Performance Analysis

In the program of requirements in Section 2 it is stated that the system should be implementable into some type of wearable, meaning that the computation time should be fast enough. For the whole drivers dataset,

Process	Time
Total Time	0.1278 ± 0.0176 (s)
HRV gen	0.0011 ± 0.0014 (s)
OSP	0.0049 ± 0.0009‘(s)
Feature gen	0.0109 ± 0.0014 (s)
Stress Detection	0.0672 ± 0.0094 (s)
Talking Detection	0.0435 ± 0.0065 (s)

Table 7.3: Computation times of the different part of the system.

there were 927 segments of 80s for which both stress detection and speaking detection was done in MATLAB.

It was run on a laptop with a dual core Intel i7-5500U @2.4 GHz CPU. This processor was first released in 2015, making it 5 years old. This computation time includes all of the pre-processing steps described in Chapter 3, then calculating the 10 features needed, and subsequently running both the speaking and stress detection algorithms. These times can be found in Table 7.3. The first 4 segment times have been removed in calculating these times, as these took much longer than the rest. If it would actually be implemented in a wearable, these start up/caching issues would not be present.

Extrapolating this data to wearable computation time, energy and memory usage is difficult and is something that is looked at in more detail in the System Design thesis in [26]. But the factor of 626 difference between 80 seconds and the computation time of 0.128 s, while always running both the stress and speaking detection algorithms, and all this on a 5 year old laptop, proves hopeful that the requirements with respect to performance are fulfilled.

Chapter 8

Discussion

8.1 Overall System Discussion

In the end, the stress detection system achieved an average classification accuracy of 80% on the test dataset, and of 77% on the drivers dataset if only considering the first baseline. If no distinction is made between pre-drive and post-drive baseline, the accuracy of the baseline would be 70%, resulting in an average classification accuracy of 73%. A few examples of similar studies in literature are compared.

In [3] a 71.8% accuracy using random forests was achieved. A deep learning approach was also implemented, achieving an accuracy of 83.9%. But not only did this study not account for the effect speaking has on features, subjects were *always* speaking during the stressful stages. The segment length was a bit shorter at 50 seconds.

In [15] a talking baseline is included and shows that most misclassifications for the rest state were situated there. An average classification accuracy of 84.6% was achieved using Bayesian networks. Other than ECG and respiratory signals, body temperature and galvanic skin response (GSR) signals were also used. GSR measures the conductivity of the skin, which is an indicator for how much a subject is sweating. The final algorithm used 4 features, of which only 1 was an ECG feature and the rest were GSR features, which indicates that using GSR is a very good way to detect stress. A segment length of 30 seconds was used.

In [27] a 84.5% classification accuracy was achieved, by using ECG, body temperature and GSR. The mean GSR was by far the best feature, again indicating the good performance of GSR.

In [14] a 74.5% accuracy was achieved, using ECG, respiration, GSR and electromyography (EMG) of the trapezius muscle. EMG measures the electrical activity of certain muscles. A segment length of 120 seconds was used.

Compared to these 4 studies, the performance of the algorithm designed in this project is comparable with the state of the art. Similar accuracies are achieved, often while using less physiological signals and while also accounting for the influence of speaking. The algorithm also works on a completely different dataset, showing it's applicability in real world situations.

8.2 Features Discussion

The following features are used in the final machine learning algorithm, they are ordered by relevance, as indicated by the mRMR algorithm. The definitions of these features can be found in Table 4.1. Boxplots of these features can be found in Figure 8.1.

Feature Nr. 38 - $\mathbf{P_R}\, hl$
This is defined as the ratio of sum of the high respiratory frequency bands (3.9-0.5 Hz) over the low frequency bands (0.5 - 0.1 Hz). As can be seen in Figure 8.1 this ratio increases for stressful states. This means that the power in the higher frequency bands of the respiratory rate increase with respect to the power in the lower frequency bands, which indicates that breathing patterns are less smooth when people are under stress.

Feature Nr. 29 - $\mathcal{P}_{\mathbf{X}}$
This is the relative power in the respiratory component of the HRV. The hypothesis was that this is a good quantification of the RSA, which in turn quantifies the parasympathetic activity which is inversely proportional to a subject stress levels. This is confirmed by the boxplots in Figure 8.1, as it is clear that the relative power in this band decreases when subjects are stressful.

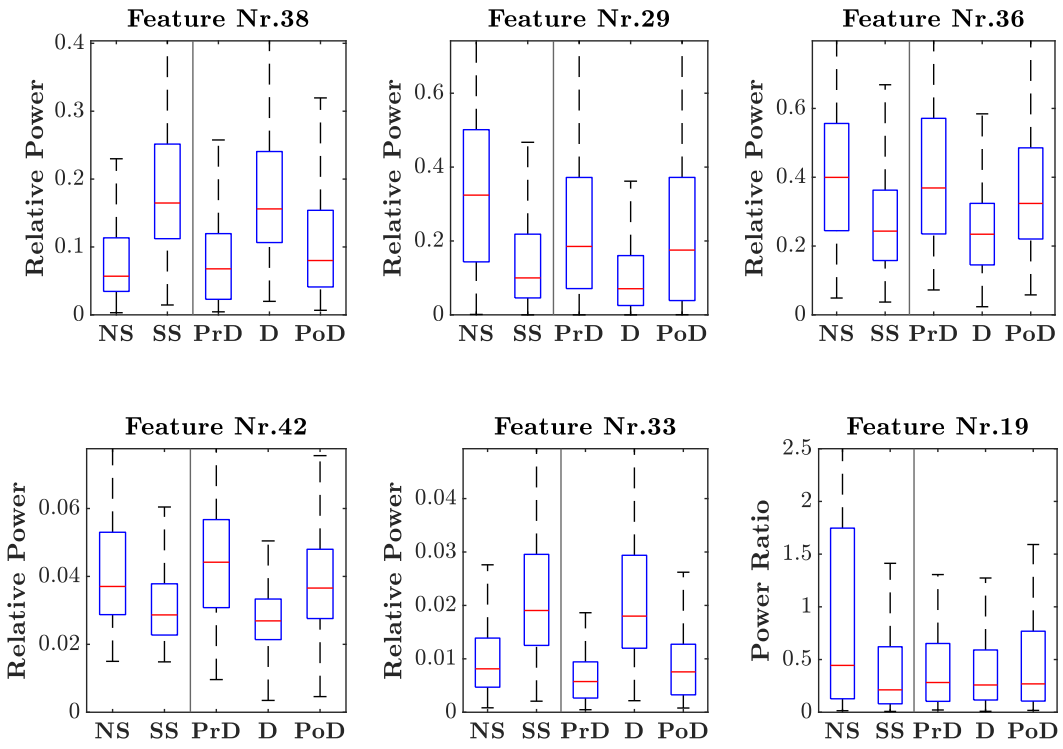

Figure 8.1: Boxplots of the 6 features that are selected for the final machine learning model. NS = no stress, test dataset; SS = silent stress, test dataset; PrD = pre-drive drivers dataset; D = driving, drivers dataset; PoD = post-drive, drivers dataset

Feature Nr. 36 - $\mathbf{P_R}\, p5$
As can be seen in Figure 8.1 the power in this respiratory band decreases under stress. This is the power in the 0.12-0.24 Hz band. Normal respiratory rates lie between 0.1 and 0.5 Hz [17], so the decrease of power in this band indicates that subjects start breathing at a rate above this band.

Feature Nr. 42 - pf2RESP
This indicates how much power resides in the fundamental respiratory frequency relative to total respiratory power. It is clear from Figure 8.1 that this ratio decreases when under stress. This again indicates that breathing patterns under stress are less smooth.

Feature Nr. 33 - $\mathbf{P_R}\, p2$
As can be seen in Figure 8.1 the power in this respiratory band increases. This is the power in the 0.98-1.95 Hz band. This is well above normal respiratory rates, so this again indicates a less smooth breathing pattern under stress.

Feature Nr. 19 - $\mathbf{SB_X}$
The sympathovagal balance is defined as the ratio of the LF band over the HF band. In Figure 8.1 it is clear that this feature changes for the first dataset, but is fairly constant for the drivers dataset. This means that this feature is not very robust and should be excluded in future systems.

As 4 out of 6 features are respiratory features, it is clear that respiration information is more important than HRV information to detect stress. Furthermore, the 2 features that do use the HRV data, also use the respiration for the decomposition. This also further establishes the importance of a 2 layer system, as respiration is very much influenced by speaking.

Chapter 9

Conclusion

The hypothesis that the HRV decomposition is a good method to quantify the RSA, and that when combining this with respiratory features, a good working stress detection system can be made, is confirmed. Comparing the system with literature, it is clear that it performs up to or above the level of the state of the art. The real world applicability of the system is also proved by the non diminishing performance on a completely different dataset.

A crucial finding of this report is how important it is to separate the influence of speaking from stress, as many physiological features are influenced by both. This is especially true of course for respiratory features, which prove to be very useful when classifying stress in silence.

The main novelty of this work relies on the development of a stress detector system based on easy-to-record signals, namely the ECG and the respiratory signal. If these easy-to-record signals are recorded with a wearable, stress can be detected in a work environment, for example an office. In such environments, people often speak and with the proposed system false stress due to speaking is minimized. Therefore future stress detectors should be based on the proposed two-layered approach.

9.1 Future Work

Future work should focus on obtaining a dataset in which speaking is present in both the baseline and the stressful stages of the test. Because there will always be a correlation between speaking and stress in many features extracted from ECG and respiratory, a two layered approach could be the way to go for stress detection systems.

Furthermore additional layers could be added to a two layered approach. If someone is exercising or moving intensely, the features extracted from the respiration and ECG are probably less correlated to stress compared to a person not moving. By combining the ECG and respiration data with an accelerometer, a detector for movement might be constructed. If intense movement is detected, a different stress detector might be used to obtain better results.

Bibliography

[1] A. Hernando et al. "Inclusion of Respiratory Frequency Information in Heart Rate Variability Analysis for Stress Assessment". In: *IEEE Journal of Biomedical and Health Informatics* 20.4 (2016), pp. 1016–1025.

[2] S. Betti et al. "Evaluation of an Integrated System of Wearable Physiological Sensors for Stress Monitoring in Working Environments by Using Biological Markers". In: *IEEE Transactions on Biomedical Engineering* 65.8 (2018), pp. 1748–1758.

[3] Wonju Seo et al. "Deep ECG-Respiration Network (DeepER Net) for Recognizing Mental Stress". In: *Sensors (Basel, Switzerland)* 19 (2019).

[4] Eunsoo Won and Yong-Ku Kim. "Stress, the Autonomic Nervous System, and the Immune-kynurenine Pathway in the Etiology of Depression". In: *Current neuropharmacology* 14 (Dec. 2015).

[5] C. Varon et al. "Unconstrained Estimation of HRV Indices After Removing Respiratory Influences From Heart Rate". In: *IEEE Journal of Biomedical and Health Informatics* 23.6 (2019), pp. 2386–2397.

[6] Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology. "Heart Rate Variability: Standards of measurement, physiological interpretation and clinical use". In: *Circulation* 93.5 (1996), pp. 1043–1065.

[7] Enes Kinaci and Talha Kuruoglu. "Thesis Pre-processing". In: (2020).

[8] M. Malik et al. "Heart rate variability. Standards of measurement, physiological interpretation, and clinical use". English. In: *European Heart Journal* 17.3 (1996), pp. 354–381. ISSN: 0195-668X.

[9] U. Wiklund, M. Akay, and U. Niklasson. "Short-term analysis of heart-rate variability of adapted wavelet transforms". In: *IEEE Engineering in Medicine and Biology Magazine* 16.5 (1997), pp. 113–118.

[10] Martin Vetterli and Cormac Herley. "Wavelets and Filter Banks: Theory and Design". In: *IEEE Transactions on Signal Processing* 40.9 (1992), pp. 2207–2232.

[11] Szi-Wen Chen. "A wavelet-based heart rate variability analysis for the study of nonsustained ventricular tachycardia". In: *IEEE Transactions on Biomedical Engineering* 49.7 (2002), pp. 736–742.

[12] Vincent Pichot et al. "Wavelet transform to quantify heart rate variability and to assess its instantaneous changes". In: *Journal of Applied Physiology* 86.3 (1999), pp. 1081–1091.

[13] Rajiv Singh, Sailesh Conjeti, and Rahul Banerjee. "A Comparative Evaluation of Neural Network Classifiers for Stress Level Analysis of Automotive Drivers using Physiological Signals". In: *Biomedical Signal Processing and Control* 8 (Nov. 2013), pp. 740–754.

[14] J. Wijsman et al. "Wearable Physiological Sensors Reflect Mental Stress State in Office-Like Situations". In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. 2013, pp. 600–605.

[15] Elena Smets et al. "Comparison of Machine Learning Techniques for Psychophysiological Stress Detection". In: *Pervasive Computing Paradigms for Mental Health*. Ed. by Silvia Serino et al. Cham: Springer International Publishing, 2016, pp. 13–22. ISBN: 978-3-319-32270-4.

[16] R. Castaldo et al. "Detection of mental stress due to oral academic examination via ultra-short-term HRV analysis". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016, pp. 3805–3808.

[17] Mason Laura. "Signal processing methods for non-invasive respiration monitoring". PhD thesis. University of Oxford, 2002.

[18] G. A. Darbellay and I. Vajda. "Estimation of the information by an adaptive partitioning of the observation space". In: *IEEE Transactions on Information Theory* 45.4 (1999), pp. 1315–1321.

[19] CHRIS DING and HANCHUAN PENG. "MINIMUM REDUNDANCY FEATURE SELECTION FROM MICROARRAY GENE EXPRESSION DATA". In: *Journal of Bioinformatics and Computational Biology* 03.02 (2005), pp. 185–205.

[20] Rossana Castaldo et al. "To What Extent Can We Shorten HRV Analysis in Wearable Sensing? A Case Study on Mental Stress Detection." In: *EMBEC & NBC 2017*. Ed. by Hannu Eskola et al. Singapore: Springer Singapore, 2018, pp. 643–646. ISBN: 978-981-10-5122-7.

[21] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32.

[22] A. K. Jain, R. P. W. Duin, and Jianchang Mao. "Statistical pattern recognition: a review". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000), pp. 4–37.

[23] Gila Benchetrit. "Breathing pattern in humans: diversity and individuality". In: *Respiration Physiology* 122.2 (2000), pp. 123–129. ISSN: 0034-5687.

[24] Samia Ben Lamine et al. "Individual differences in respiratory sinus arrhythmia". In: *American Journal of Physiology-Heart and Circulatory Physiology* 286.6 (2004), H2305–H2312.

[25] J. A. Healey and R. W. Picard. "Detecting stress during real-world driving tasks using physiological sensors". In: *IEEE Transactions on Intelligent Transportation Systems* 6.2 (2005), pp. 156–166.

[26] Geert Jan Meppelink and Yavuzhan Mercimek. "Thesis System Design of a Telehealth System". In: (2020).

[27] Lucio Ciabattoni et al. "Real-time mental stress detection based on smartwatch". In: Jan. 2017, pp. 110–111.

Appendix A

mRMR results for every feature per segment length

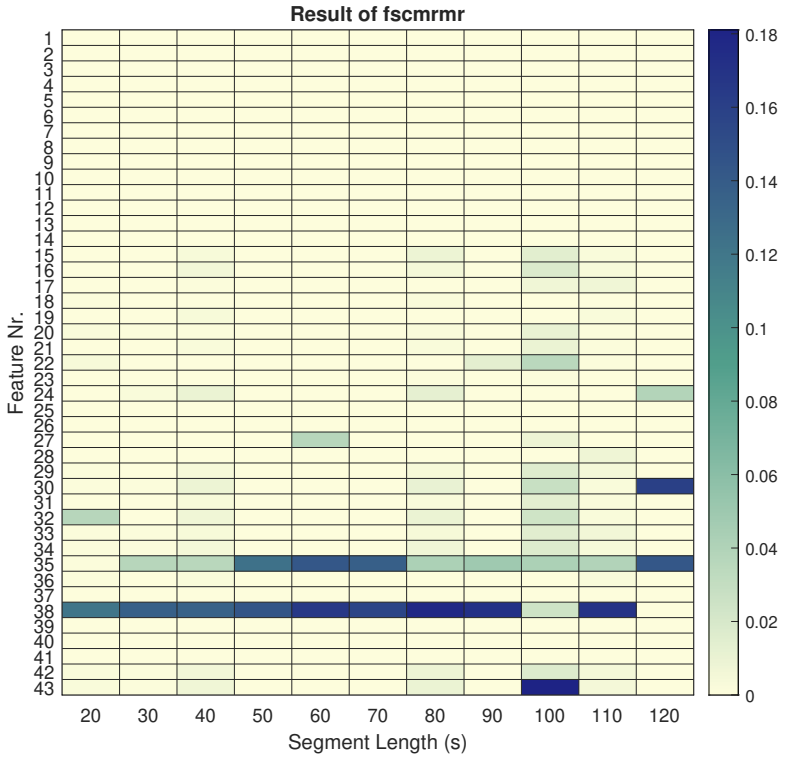

Figure A.1: Result of the mRMR algorithm for each feature for each segment length.

Appendix B

Normalized optimization

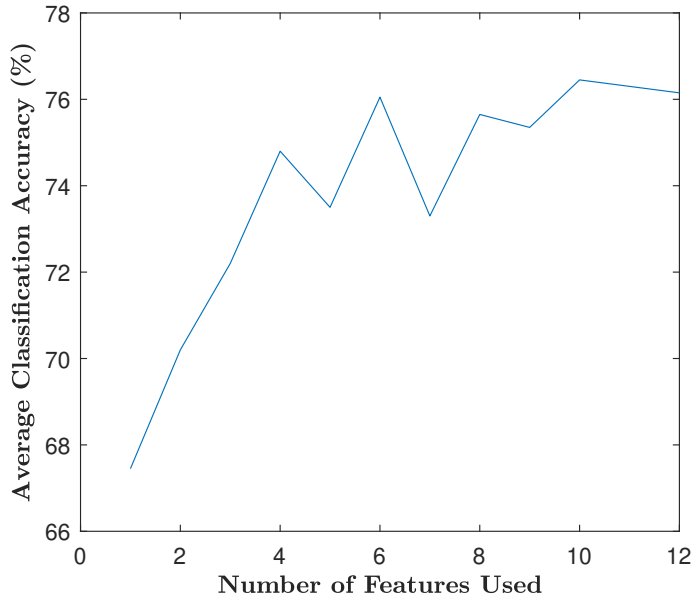

Figure B.1: Optimization of the number of features for the machine learning algorithm with normalized features. Number of learners is set at 250 and the maximum number of splits is set at 256.

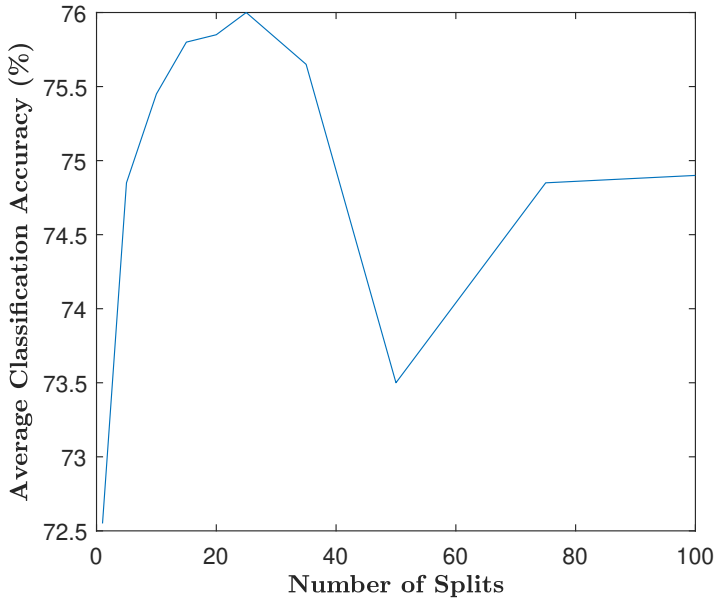

Figure B.2: Optimization of the number of splits for the machine learning algorithm with normalized features. The best 6 features are used and the number of learners is set at 250.

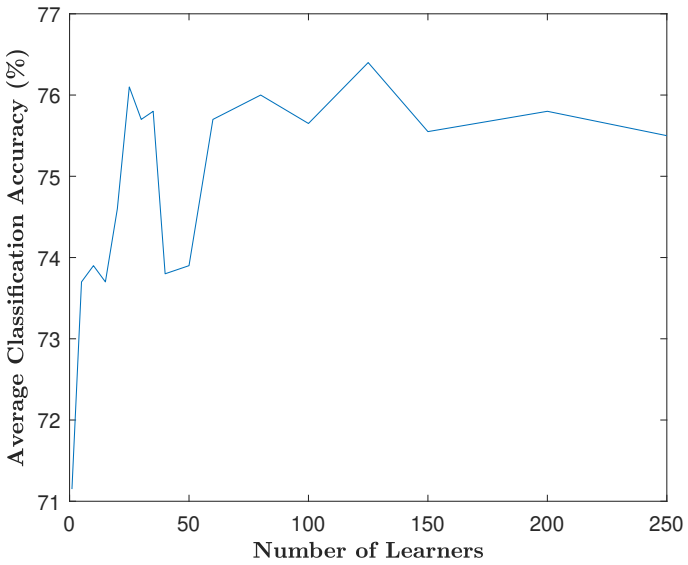

Figure B.3: Optimization of the number of learners for the machine learning algorithm with normalized features. The best 6 features are used and the maximum number of splits is set at 25.

Appendix C

Speaking Optimization

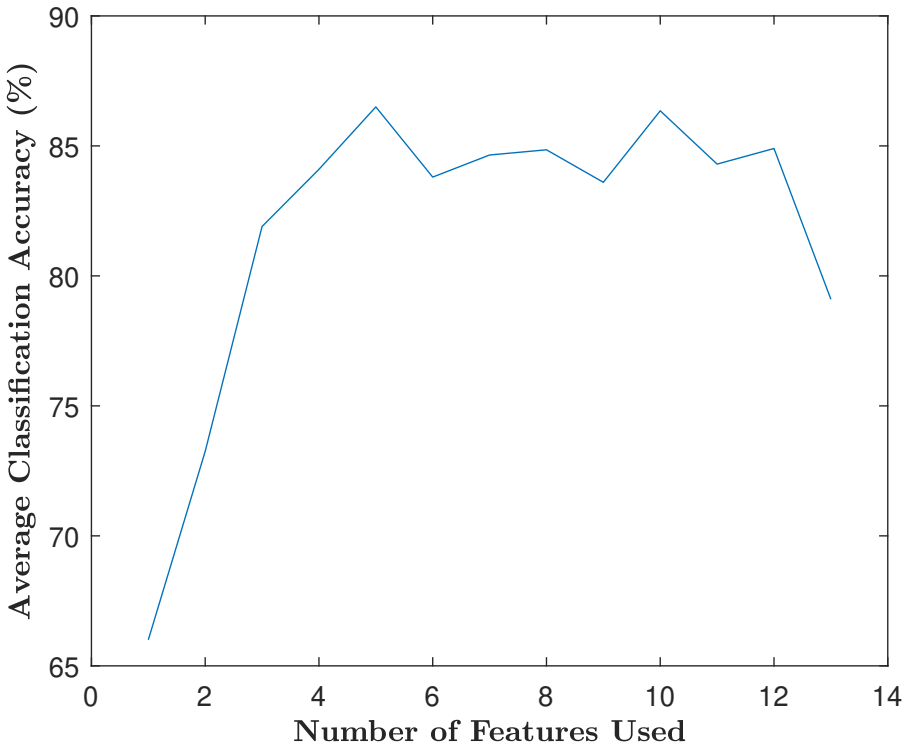

Figure C.1: Optimization of the number of features for the speaking detection algorithm. Number of learners is set at 250 and the maximum number of splits is set at 256.

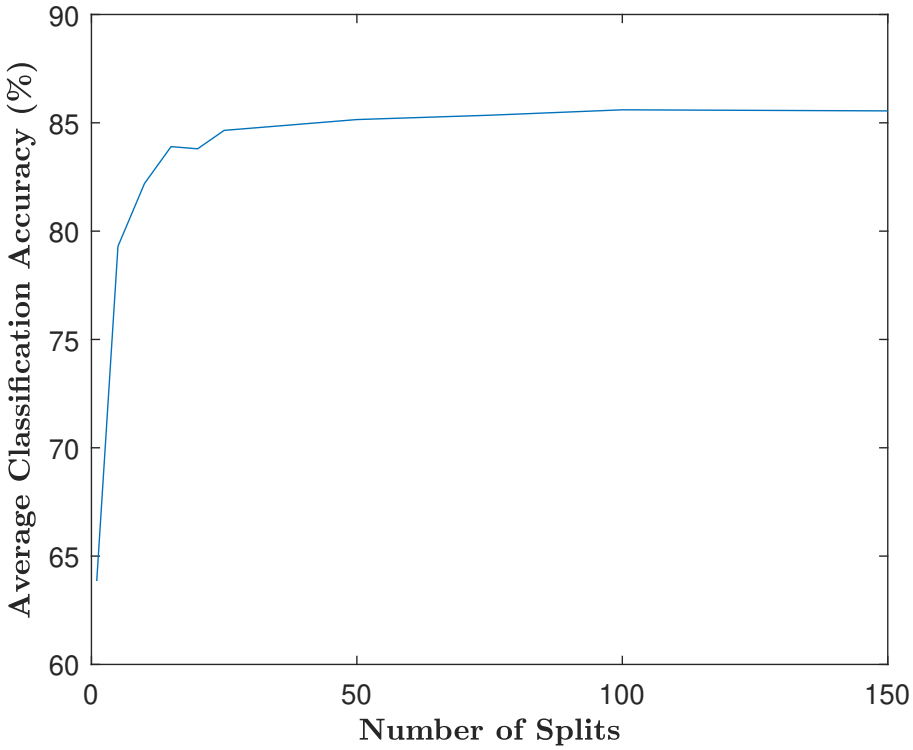

Figure C.2: Optimization of the number of splits for the speaking detection algorithm. The best 5 features are used and the number of learners is set at 250

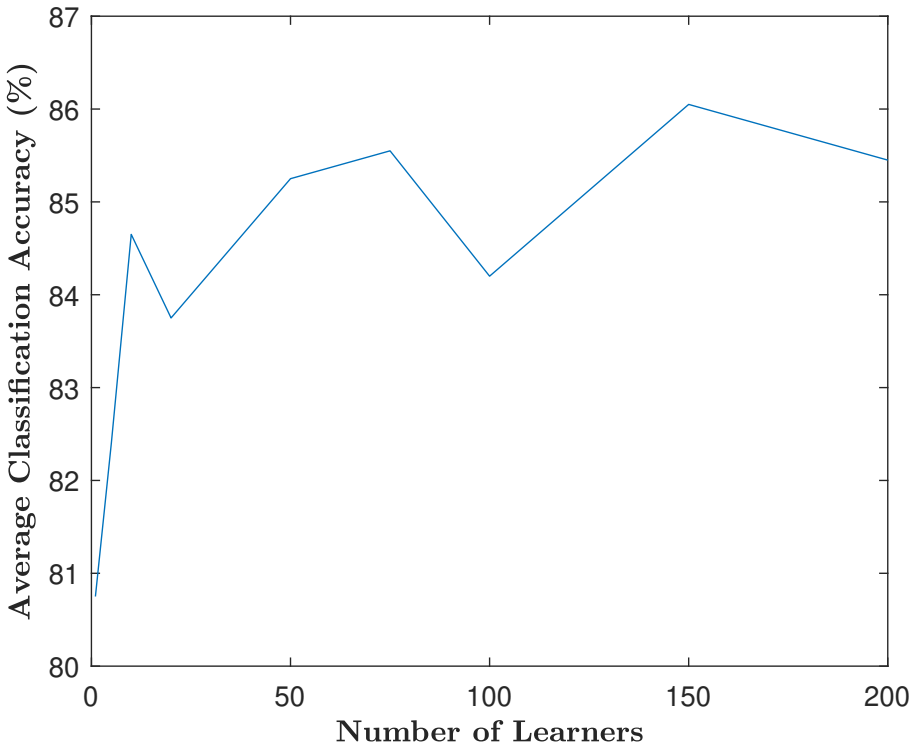

Figure C.3: Optimization of the number of learners for the speaking detection algorithm. The best 5 features are used and the maximum number of splits is set at 50.