INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Um sistema multimodal para a deteção de stress

HUGO ANDRÉ VIANA CORREIA
Outubro de 2022

POLITÉCNICO
DO PORTO

A multimodal system for stress detection

Hugo André Viana Correia

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas de Informação e Conhecimento

Supervisor: Doutora Fátima Rodrigues

Porto, October 2022

Abstract

Stress is the physiological or psychological response to internal or external factors, which can happen in short or long terms. Prolonged stress can be harmful since it affects the body, negatively, in several ways, thus contributing to mental and physical health problems.

Although stress is not simple to properly identify, there are several studied approaches that solidify the existence of a correlation between stress and perceivable human features.

In order to detect stress, there are several approaches that can be taken into consideration. However, this task is more difficult in uncontrolled environments and where non-invasive methods are required. Heart Rate Variability (HRV), facial expressions, eye blinks, pupil diameter and PERCLOS (percentage of eye closure) consist in non-invasive approaches, proved capable to accurately identify the mental stress present in people.

For this project, the users' physiological signals were collected by an external video-based application, in a non-invasive way. Moreover, data from a brief questionnaire was also used to complement the physiological data.

After the proposed solution was implemented and tested, it was concluded that the best algorithm for stress detection was the random forest classifier, which managed to obtain a final result of 84.04% accuracy, with 94.89% recall and 87.88% f1 score. This solution uses HRV data, facial expressions, PERCLOS and some personal characteristics of the user.

Keywords: Stress, Machine Learning, Classification, Heart Rate Variability, Facial Expressions, Eye Blink, Pupil Diameter, PERCLOS

Resumo

O stress é a resposta fisiológica ou psicológica a fatores internos ou externos, o que pode acontecer a curto ou longo prazo. O stress prolongado pode ser prejudicial uma vez que afeta o corpo, negativamente, de várias formas, contribuindo assim para problemas de saúde mental e física.

Embora o stress não seja simples de identificar corretamente, existem várias abordagens estudadas que solidificam a existência de uma correlação entre o stress e as características humanas percetíveis.

De forma a detetar o stress, existem várias abordagens que podem ser tidas em consideração. No entanto, esta tarefa é mais difícil em ambientes não controlados e onde são necessários métodos não invasivos. A variabilidade da frequência cardíaca (HRV), expressões faciais, piscar de olhos e diâmetro da pupila e PERCLOS (fecho ocular percentual) consistem em abordagens não-invasivas, comprovadamente capazes de identificar o stress nas pessoas.

Para este projeto, os dados fisiológicos dos utilizadores são recolhidos a partir de uma aplicação externa baseada em vídeo, de forma não invasiva. Além disso, serão também utilizados dados recolhidos a partir de um breve questionário para complementar os dados fisiológicos

Após a implementação e teste da solução proposta, concluiu-se que o melhor algoritmo de deteção de stress foi o *random forest classifier*, que conseguiu obter um resultado final de 84,04% de *precision*, com 94,89% de *recall* e 87,88% de f1 *score*. Esta solução utiliza dados de HRV, expressões faciais, PERCLOS e certas características pessoais do utilizador.

Palavras-chave: Stress, Machine Learning, Classification, Heart Rate Variability, Facial Expressions, Eye Blink, Pupil Diameter, PERCLOS

Contents

List of Figures

List of Tables

Acrónimos e Símbolos

Acronym List

AHP	*Analytic Hierarchy Process*
API	*Aplication Programming Interface*
ANS	*Autonomic Nervous System*
AUROC	*Area Under the Receiver operating characteristic*
AVNN	*Average of NN intervals*
BMC	*Business Model Canvas*
BPM	*Beats Per Minute*
CI	*Consistency Index*
CR	*Consistency Ratio*
ECG	*Electrocardiogram*
EDA	*Electrodermal Activity*
EU	*European Union*
FAST	*Function Analysis System Technique*
FP	*False Positive*
FN	*False Negative*
GSR	*Galvanic Skin Response*
HF	*High Frequency*
HRV	*Heart Rate Variability*
LF	*Low Frequency*
MAE	*Mean Abosulte Error*
MLP	*Multilayer Perceptron*
NCD	*New Concept Development*
PERCLOS	*Percentage Eye Closure*

PNN20 *Percentage of adjacent NN intervals that differ from each other by more than 20 milliseconds*

PNN25 *Percentage of adjacent NN intervals that differ from each other by more than 25 milliseconds*

PNN50 *Percentage of adjacent NN intervals that differ from each other by more than 50 milliseconds*

PPG *Photoplethysmogram*

PSS *Perceived Stress Scale*

RBF *Radial Basis Function*

REST *Representational State Transfer*

RI *Random Index*

RMSE *Root Mean Square Error*

RMSSD *Root mean square of successive differences between normal heartbeats deviation of the average NN intervals*

SDNN *Standard deviation of the NN intervals*

SDNN_RMSSD *Ratio of SDNN to RMSSD*

SVM *Support Vector Machine*

TP *True Positive*

TN *True Negative*

UC *Use Case*

UI *User Interface*

ULF *Ultra Low Frequency*

VLF *Very Low Frequency*

VSM *Vector Space Model*

Symbol List

λ Lambda

1 Introduction

This chapter starts with a contextualization of the project, as well as the definition of the underlaying problem, followed by the desired objectives of the proposal.

1.1 Context

Stress is, unfortunately, a common feeling in every people’s day-to-day life, specially at work. Although there are perceivable aspects of a stressed person that can be easily identifiable, there also exists underlaying body responses that need proper analysis in order to be seen, which can mean that a person might not be aware of his/her own stress.

There are several situations that force a person to continuously stay stressed, the most common one being at work. The majority of people work most of the days, around 8 hours a day, 5 days a week. This means that if people are under a lot of stress in their work time, it can be considered as a long-term stress exposure.

Being stressed for a long period of time causes several health problems. These can affect the body physically, manifesting as cardiovascular diseases or diabetes, as well as psychologically, in the case of depression, burnouts or even bipolar disorders (Ganster & Rosen, 2013).

This dissertation is within the scope of the Mad@Work ITEA project[1], that focuses on detecting and mitigating poor mental health conditions such as work stress and burnout, which have not yet resulted in a diagnosed mental health disorder (Mad@Work, 2021).

[1] Mental Wellbeing Management and Productivity Boosting in the Workplace (Mad@Work), POCI-01-0247-FEDER-046168, Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico (SI&IDT), 09/2020 a 06/2023.

1.2 Problem

Stress happens when a person cannot handle high pressure situations. At work, stress is most often present, affecting the person's quality of life, as well as the work performed. Severe stress at work can contribute to several diseases, including both mental and physical conditions. This concerns both the employees and the companies, since the workers' productivity and motivation are often negatively affected, which can also lead to sick leaves.

Although it is possible to perceive stress, there is a lack of proper solutions to mitigate mental health issues in these scenarios. Therefore, an approach capable of detecting a person's stress level in the workplace non-invasively becomes necessary, in order to combat it.

1.3 Objectives

The focus of this thesis is to develop Machine Learning models to detect stress by using physiological signals, such as, HRV, facial expressions, eye blink frequency, pupil diameter and PERCLOS. To that end, the objectives to fulfill will be as follows:

- Retrieving the required data from the test subjects, in a work environment. HRV, facial expressions, eye blinks, pupil diameter and PERCLOS will be obtained through an already existing video plethysmography-application that analyzes the person's face and retrieves the desired signals.
- The development of a pop-up application to remind the workers to fill a self-report questionnaire, whose main goal is to complement the data collected by the video-based application and serve as ground truth.
- Defining the most relevant features extracted from the subjects
- Determining the most adequate machine learning algorithms to develop stress detection models
- Analyze and test the developed models in order to conclude which one has the highest predictive power to identify stress.
- Implement the best model into an application, which will notify the user about their stress levels.

1.4 Methodology

The stress detection will be done through Machine Learning models developed with multimodal data from several physiological signals, using the CRISP-DM (Wirth and Hipp, 2000) methodology, which consists in a structured and robust approach for the planning of projects involving Machine Learning and data analysis.

After the extraction and pre-processing of the data, machine learning models will be developed in order to detect stress, which can be classified as "stress" or "not stress" (binary classification).

All the developed models will then be properly evaluated, using the most appropriate metrics, to assess which model performs the best in each scenario.

Finally, an application will be developed. The best machine learning model obtained will be used through this application to identify stress in the users.

1.5 Value Analysis

Work related stress is becoming a serious threat in the workplace. The increase in demanding jobs is one of the main causes for this issue. In the European Union (EU), there are several reports of poor mental health conditions related to work, namely, mental stress, which concerns both the companies and the employees. By analyzing this opportunity, it's possible to unveil a solution to attract this market segment, which consists in a multimodal, non-intrusive, stress prediction system for work environments. This solution can be used in the form of an application to detect and inform users about their stress, in order to combat the mental health hazards, present in the workplace.

Although the system uses four different physiological signals to predict stress, the HRV signal needs to be pre-processed in order to retrieve derived features as stress markers. As such, a multi criteria analysis was performed by comparing three Python libraries used to extract several metrics derived from HRV.

A function analysis for this system was also developed in order to help visualize all the required functions and the relationships between them.

1.6 Contributions

In this project, the author describes an end-to-end pipeline that allows the detection of stress in people at work in a non-invasive manner. Our approach comprises three parts: i) data acquisition, ii) modelling, and iii) evaluation. Each of these parts are associated with contributions detailed as follows:

i) Data Acquisition. Several physiological signals, such as, HRV, facial expressions, eye blinks, pupil diameter and PERCLOS will be collected during working periods by the same video plethysmography application which guarantees the synchronization between the different signals collected. Also, there will be end-user confirmation through a pop-up application that retrieves the perceived stress in order to label the physiological data.

ii) Modelling. Different machine learning models with individual and combined physiological signals will be created, in order to explore all the available alternatives in this scenario.

iii) Evaluation. Measurement of a series of evaluation metrics for each of the developed Machine Learning models to determine which one is the optimal solution to be implemented in the proposed system.

1.7 Document Structure

This document is divided in 5 chapters. First, in this section, the project is contextualized in the area it is inserted in. Then, the present problem is briefly explained, followed by the description of the objectives to be fulfilled with this project and the respective methodologies used to properly achieve them. This chapter is concluded with the project's contributions to the field in which it is situated.

In the State of the Art, the most relevant terms such as, stress and respective physiological responses are introduced. A comparative analysis is also performed between approaches already implemented in the area.

In chapter 3 a value analysis is developed for this project.

In chapter 4 the proposed design of the application is presented.

The fifth chapter consists in the implementation and evaluation of the solution. Here, all of the steps for the development and evaluation of the machine learning models are thoroughly described.

In the last section, a conclusion for this document is presented, including the main insights and results obtained with in this work.

2 State of the Art

In this chapter, the main terms and concepts used in this thesis are presented and briefly explained. The focus will be in the definition and contextualization of the physiological signals and machine learning techniques to be used. In the end, there will be an analysis of several related solutions in the area, followed by a comparison between them.

2.1 Physiological Signals of Stress

Stress can be described as the feeling of being overwhelmed with mental or emotional pressure, provoking both mental and physical strain (Mental Health Foundation, 2021). These stress responses can be measured with self-report measures, behavioral coding or via physiological measurements (Crosswell and Lockwood, 2020). In fact, studies in the area of psychophysiology utilize physiological data (e.g., Cardiovascular and electrodermal activity) to assess affective states, such as mental stress (Sioni and Chittaro, 2015).

In order to produce a predictive system for stress in work environments, there's a need to consider non-invasive approaches. To that end, this project will use three studied physiological markers that proved being able to accurately detect mental stress in people.

2.1.1 Heart Rate Variability

Heart Rate Variability (HRV) consists in a measure of the variation in time between beat to beat and is controlled by a part of the nervous system called Autonomic Nervous System (ANS) (Harvard Health Publishing Staff, 2021). HRV is a well-studied metric in cardiovascular diseases and has been shown to be related to changes in neurohormonal activation (Bilchick and Berger, 2006). It was also proved through several studies that HRV features changes according to induced stress (Kim *et al.*, 2018).

As proved by (Dalmeida and Masala, 2021), HRV-derived features can be good stress indicators, since the best performing model developed managed to achieve a Recall of 80%. Their work used the time interval between heartbeats, also named RR (Moody, 2002), to extract the HRV features for the machine learning models. The term NN intervals refer to the intervals between normal (sinus) beats, which are the ones often used (Moody, 2016), since abnormal beats are removed (Shaffer and Ginsberg, 2017). However, the terms NN and RR are almost always used as synonyms.

A visual representation of HRV with the corresponding RR intervals is shown in Figure 1.

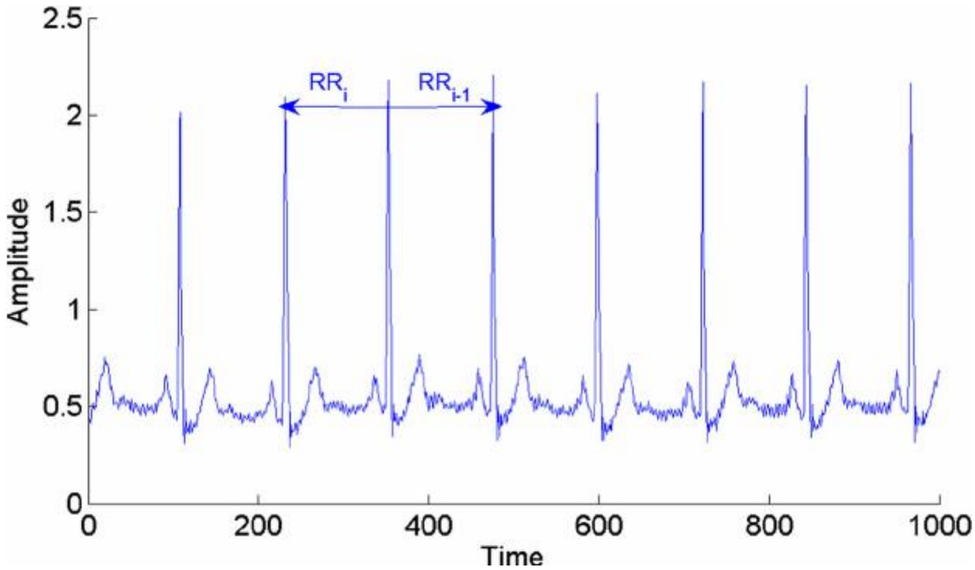


Figure 1 – Visual representation of the HRV with measured RR intervals (Chen, Zhang and Wang, 2015)

The article (Shaffer and Ginsberg, 2017) describes the metrics associated with HRV, dividing them into “time-domain”, “frequency-domain” and “non-linear features”. This subdivision of the HRV is also supported in the earlier mentioned article (Dalmeida and Masala, 2021), although not elaborating on “non-linear features”. Some of the defined metrics are the following:

- Time Domain Metrics: Standard deviation of the NN intervals (SDNN), Standard deviation of the average NN intervals (SDANN), Average of NN intervals (AVNN), Root mean square of successive differences between normal heartbeats (RMSSD) (Shaffer and Ginsberg, 2017) and the percentage of adjacent NN intervals that differ from each other by more than 50 milliseconds (pNN50) (Shaffer and Ginsberg, 2017)
- Frequency Domain Metrics: Total power, Low frequency (LF), High frequency (HF), Very low frequency (VLF), Ultra low frequency (ULF) and LF/HF ratio

In their study (Dalmeida and Masala, 2021), certain features such as LF/HF ratio, LF and HF were proved to be the least significant predictors across all the performed tests.

2.1.2 Facial Expressions

According to the American Psychological Association, a facial expression is a form of nonverbal signaling using the movement of facial muscles (American Psychological Association, 2020). Facial expressions can also be considered as configurations of different small muscle movements in the face that are used to deduce a person's emotional state (Harley, 2016).

As proposed by Paul Ekman, the six basic human emotions consist in fear, anger, joy, sadness, disgust, and surprise (Ekman, 1992; Chen *et al.*, 2019; Keltner *et al.*, 2019). This list of emotions is the most widely acknowledged in terms of basic emotions (Kowalska and Wróbel, 2017).

Several studies affirm that facial expressions can consist in a body's response to stress. According to an investigation, facial expressions such as fear, anger and disgust reflect biological responses to stress. Results show that, when exposed to stressors, the subject's fear increased accordingly with their cortisol and cardiovascular responses to stress. However, when anger and disgust responses were higher, the cortisol and cardiovascular responses were lower (Lerner *et al.*, 2007).

Another recent study's results states that anger and disgust are good stress indicators, producing systems with high predictive power. In this experiment, the earlier mentioned facial expressions are the basic emotions considered to be related to stress. Facial expressions such as: fear, happiness, neutral, sadness and surprise were also used in order to train the models (Gao, Yüce and Thiran, 2014).

2.1.3 PERCLOS

PERCLOS, which consists in an acronym for "percentage eye closure" (Trutschel *et al.*, 2011), is a video-based driver drowsiness monitoring technology which assesses drowsiness by measuring slow eyelid closure and estimating the proportion of time for which the eyes are closed over specified time intervals (Li and Chung, 2014).

According to an investigation related to drowsiness-detection technologies and measures, the PERCLOS was concluded to be the most reliable approach to identify a driver's alertness level (David F. Dinges and Grace, 1998). Moreover, another study also states that, according to the results obtained, the PERCLOS value is higher when the driver is drowsy, and lower when the driver is alert (Junaedi and Akbar, 2018).

A study has been also made where the assessment of mental stress is also based on eye related area. Among the other measures, it was concluded that the PERCLOS value decreases with increase in the mental workload. The subjects proved to be more attentive at high difficulty levels. A relaxing state of mind reflected in high PERCLOS values (Petkar *et al.*, 2009). As stress is also related with the person's alert state, the possibility of an existing link between the stress and the PERCLOS is tested in this thesis.

2.1.4 Pupil Diameter

The iris of the eye is responsible for the variation in the pupil diameter, which controls the amount of light allowed to pass into the eye (JONES, 2005). This diameter has been proven to be related to the age of the person, while being independent of gender, refractive error, or the iris color (Winn *et al.*, 1994).

Several emotional events, such as examining photographs, listening to sounds, or being threatened with shock, cause the pupil width to increase (Henderson, Bradley and Lang, 2018). Investigations on stress classification have already established a relationship between mental stress and the pupil diameter size (Pedrotti *et al.*, 2014; Torres-Salomao, Mahfouf and El-Samahy, 2015). Additionally, according to a recent study, the eye dilation decreases while the subjects are under increased workload (Othman and Romli, 2016). This way, the pupil diameter can be used to build an accurate stress prediction model in work environments.

2.1.5 Eye Blink Frequency

A blink is a temporary closure of both eyes, involving movements of the upper and lower eyelids (Blount, 1927). Humans make approximately 13,500 spontaneous blinks each day, occurring 15 times per minute, in average (Peshori *et al.*, 2001). Our eye blinks can be classified in three different types: involuntary/spontaneous blinks, which occur unconsciously, voluntary blinks, which are generated consciously, and reflex blinks, which are induced by different kinds of stimuli (Abe *et al.*, 2013).

Eye blinks are also a useful indicator of cognitive processing, demonstrating sensitivity to demands in several tasks (Irwin and Thomas, 2010).

An experiment was performed in order to detect a correlation between the number of eye blinks and the occurrence of stressful events, and the results indicate that the eye blinks are increased in the stressful scenario, suggesting that a link between eye blinks and stress exists (van den Haak *et al.*, 2010). In fact, other studies already conclude that a strong correlation between eye blink frequency and stress exists, in humans, stating that a higher frequency of eye blinks occurs in stressful situations (Haak *et al.*, 2009). Thus, eye blink frequency analysis is considered as a non-invasive method to predict stress.

2.2 HRV Toolkits

In order to properly obtain insights from the HRV, the signal must be pre-processed to extract the necessary features. This process is facilitated through the use of HRV toolkits that perform the required measurements. Thus, three reliable HRV python libraries were introduced and briefly analyzed.

2.2.1 pyHRV

pyHRV is an open-source HRV toolbox for the python programming language (Gomes, Margaritoff and Silva, 2019). This library was developed in order to aid researchers and developers with a trustworthy approach on HRV and provides several methods to execute several important tasks, the most important being the computation of the relevant time/frequency domain features.

The time and frequency domain modules contains a wide variety of functions. All of the most relevant features, according to the earlier mentioned article (Dalmeida and Masala, 2021), can be extracted from pyHRV. This library also provides several alternative methods to compute the frequency domain measures.

All the features are properly documented in an easily comprehensible way so that the use of this toolkit can be straightforward.

2.2.2 HeartPy

HeartPy is a HRV analysis toolkit designed to process PPG data extracted from a Photoplethysmogram (PPG) or camera sensors (van Gent *et al.*, 2019b, 2019a). This library provides easy access to frequency and time domain measures. It contains various measurements for time and frequency domain, such as SDNN, RMSSD and pNN50 for time domain and LF, HF and HF/LF ratio for frequency domain.

This toolbox also provides pre-processing functions which can aid the data visualization. Although not as extensive as the pyHRV documentation, HeartPy also provides a well-documented guide for its use, as well as pre-built example notebooks that demonstrate how to perform several analytical tasks with the toolkit.

2.2.3 hrvanalysis

hrvanalysis is a Python module for HRV analysis of RR-Intervals (Champseix, Ribiere and le Couedic, 2021). It provides pre-processing functions such as outlier-removal methods from the RR-intervals and ectopic beat filtering to improve the quality of the data obtained. Although having a reduced documentation with few examples, it provides a wide range of features while also being simple to use.

2.3 Machine Learning

Machine learning is a growing field of computing algorithms that aim to mimic human intelligence by learning from their surroundings (el Naqa and Murphy, 2015). This way, computers can make predictions based on pre-acquired knowledge (Bülbül and Ünsal, 2011).

Supervised learning consists in the use of labeled datasets to properly train algorithms (IBM Cloud Education, 2020). This helps the classification algorithms finding a correlation between the input data and the target variable, in order to predict it.

In machine learning, classification is a subset of supervised learning problems where the target variable consists in a categorical instance (Sen, Hajra and Ghosh, 2020). The main purpose of classification algorithms is to group data in the required structure based in common characteristics (Bülbül and Ünsal, 2011). In the scope of this project, the target variable is “stress”. For it to be categorical, it could either be classified as Stressed/not stressed or with different labels such as: “Mildly stressed, extremely stressed, not stressed”. To answer this problem, there are several known classification algorithms that can be used, such as (Huang, Chen and Wang, 2007): K-nearest Neighbors, Decision Tree, Support Vector Machines (SVM), Logistic Regression, Naïve Bayes, Gradient-Boosted Tree or even Random Forest.

2.4 Evaluation

The task of evaluating classification models is to measure the degree to which the classification suggested using the model corresponding to the actual classification of the case (Novaković *et al.*, 2017).

In order to properly evaluate the effectiveness of a classifier, a confusion matrix can be developed. This matrix consists in a table with the results obtained while testing the developed models. The machine learning models will predict a label and then compare that result with the actual (correct) result. In this case, a true positive (TP) would represent a positive value (“Stress”) that was correctly predicted by the model, while a true negative (TN) consists in a correct prediction of a “Not stress” label. The same logic applies to the false positive (FP) and false negative (FN) notations, which represent incorrect predictions made by the model. In this case, a FP consists in an incorrect prediction of an actual negative, being identified as a positive value (“Stress”), while a FN is prediction of a negative label (“Not stress”) that was an actual positive.

An example of a confusion matrix is demonstrated in Table 1.

Table 1 – Example of a confusion matrix

	Stress (Predicted)	Not stress (Predicted)
Stress (Actual)	(TP)	(FP)
Not stress (Actual)	(FN)	(TN)

With the information obtained from the testing phase of the machine learning models, several evaluation metrics can be calculated to give a deeper insight on the performance of the models. For this thesis, the metrics used will be the Accuracy, Balanced Accuracy, Precision, Recall and F1 Score. Although the accuracy is the more commonly used metric, it should be complemented with more insights. The recall metric serves as an example of a good complementary indicator, since it indicates the percentage of TP cases that the model can detect. The Balanced Accuracy can also be helpful to evaluate classification models when the classes are not balanced.

The equations for these metrics are exemplified below.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}\right) \tag{2}$$

$$Precision = \frac{TP}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

$$F1\ Score = 2 \text{ x } \frac{Precision \text{ x } Recall}{\mathrm{Precision} + \mathrm{Recall}} \tag{4}$$

In order to evaluate the machine learning models with the previous metrics, there's a need to obtain testing results, which will be used to build a confusion matrix. The two most popular methods used in the evaluation of machine learning models are the holdout method and the k-fold cross validation.

The holdout method is straightforward. In this case, the data is divided in two sets: The training set, which usually contains between 70% and 80% of the data, and the test set that contains the rest. The training set is solely used in the learning process of the models, while the test set is exclusively used for the model to make predictions on the test data. The predicted labels are compared alongside the actual labels to build a confusion matrix.

The k-fold, on the other hand, is used in order to do a more rigorous evaluation of the solution. This procedure divides a limited dataset into k non-overlapping folds. Each of the k folds will be used as a held back test set, while all the other folds are used as a training dataset (Brownlee, 2020). A representation of this method is presented in Figure 2

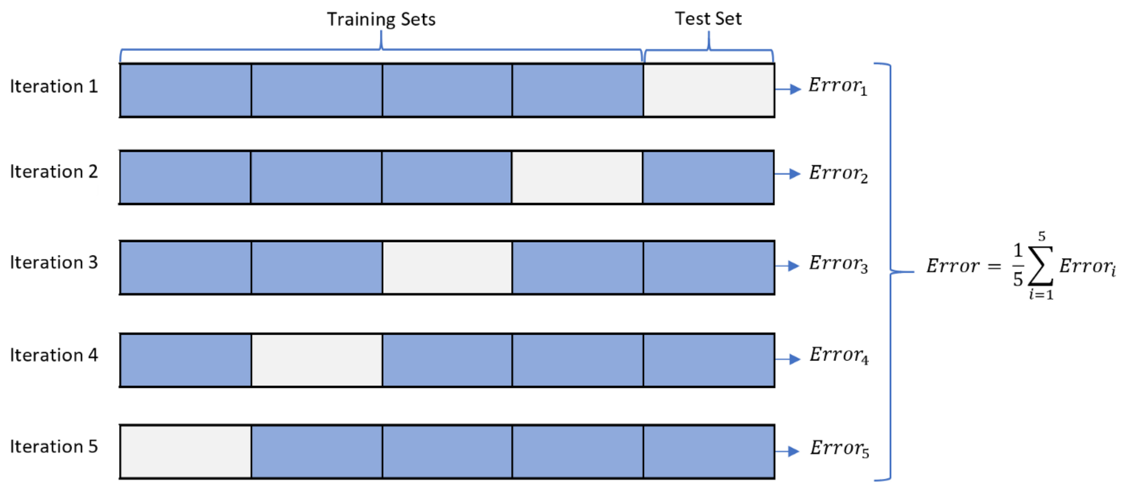


Figure 2 – Representation of a K-fold cross validation method (Patro, 2021)

2.5 Existing Solutions

There are several approaches on stress prediction. Several methods used consist in intrusive and non-practical approaches such as: Electrocardiogram (ECG), Galvanic Skin Response (GSR) or cortisol levels in saliva (VanBruggen *et al.*, 2011). As such, the existing solutions presented in this section should follow a non-intrusive approach on stress detection, in order to relate to the proposal of this thesis. The chosen solutions will also be the ones using similar biological signals to the ones proposed by this document, emphasizing on HRV.

The analyzed projects will be briefly explained, and their results will be demonstrated and succinctly discussed. In the end, a conclusion based on all the mentioned existing solutions' approaches and results is elaborated.

2.5.1 HRV Features as Viable Physiological Markers for Stress Detection Using Wearable Devices

This article (Dalmeida and Masala, 2021) presents an effective and non-invasive approach on stress detection. One of the goals was to classify stress mainly using HRV-derived metrics, obtained from wearable devices. The device utilized was the Apple Watch. By using the Apple Breathe App, it was possible to extract the Beats Per Minute (BPM) and so, calculate the RR intervals.

The programming language used was Python, which had access to useful libraries such as NumPy for time domain features and pyHRV (Gomes, Margaritoff and Silva, 2019) for frequency domain features.

From the several developed models, the SVM model with Radial Basis Function (RBF) kernel provided the best accuracy, with 83.33%. However, the chosen model was a Multilayer

Perceptron model (MLP) which attained a recall of 80% and an Area Under the Receiver Operating Characteristic (AUROC) of 75%. The recall in this study is considered important because of the high cost of a false negative score.

Lastly, a web application was built in order to test the developed models. The authors state that the application was able to predict stress conditions with a 71% prediction probability and 79% in relaxed states.

2.5.2 Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study

This study (Can *et al.*, 2019) presented an automatic stress detection system through non-obstructive wearable devices that could be used daily. The developed system contained heart activity data from a Photoplethysmogram (PPG) sensor, skin conductance from the GSR, accelerometer and temperature data. The PPG sensor is used to measure the blood flow, thus providing the RR intervals. The result was a three-class detection system that was capable of differentiate the stress levels of three known contexts: “Free day”, “lecture” and “contest” sessions.

In the three-class stress level detection, a 90.40% accuracy was obtained from the use of Empatica E4 devices that provided high data quality. Samsung S devices obtained an 84.67% accuracy. For person-specific models, as expected, the accuracy was higher, achieving a maximum value of 97.92%. The general models obtained a maximum of 88.20% accuracy. With a combination of heart activity with electrodermal activity, the maximum accuracy obtained was 92.15%, while individually, this value was only 86.27%. This proves that a mixture of approaches can beneficial.

Although this study uses GSR as a classifier, it was worth mentioning due to its non-invasiveness by extracting all the data required from practical non-obstructive wearable sensors.

2.5.3 Remote Measurement of Cognitive Stress via Heart Rate Variability

This article (McDuff, Gontarek and Picard, 2014) demonstrates a remote solution for cognitive stress measurement. This is achieved through a camera that is able to detect physiological signals, such as heart rate, breathing rate and HRV. In order to verify the remote signals, contact sensor measurements such as PPG and respiratory measurements were also collected.

The developed Support Vector Machine (SVM) classifier achieved an 85% accuracy. Moreover, the results also show that HRV components and breathing rates were the strongest predictors of cognitive stress.

2.5.4 Detecting emotional stress from facial expressions for driving safety

An experiment was conducted (Gao, Yüce and Thiran, 2014) in order to develop a monitoring system capable of detecting the emotional stress of drivers by analyzing their facial expressions. This system considers that the subjects are stressed if anger or disgust related expressions are constantly present for a certain period of time. For this research, two data sets were recorded. Set 1 was recorded in an office setting, consisting in 42 videos taken from 21 subjects. Set 2 was recorded 20 videos of 12 subjects in a car setting.

The best performing system managed to achieve an accuracy of 90.5%, with a recall of 0.860 and precision of 0.882 (F-measure = 0.871) for Set 1. For Set 2, the accuracy was 85% with a recall of 0.735 and 0.914 precision (F-measure = 0.815).

2.5.5 Prediction of Daily Mental Stress Levels Using a Wearable Photoplethysmography Sensor

This study (Park, Kim and Kim, 2018) aimed to uncover the viability of mental stress level prediction through the HRV attained from the PPG sensors in wearable devices.

The experiment involved the measurement of the subject's PPG signals for 30 seconds, three times a day, using the wearable device. At the end of the day, the participants evaluated their own mental stress using the Perceived Stress Scale (Cohen, Kamarck and Mermelstein, 1994) (PSS). Through a linear regression model, a correlation analysis between the PSS score of each participant and the LF/HF obtained from their HRV measurements was elaborated. Results show that the LF/HF feature was reasonably correlated with the subject's PSS scores, obtaining a mean correlation coefficient of 0.64.

Then, a linear regression model was developed to determine the possibility of predicting a person's stress levels from the HRV measures extracted from the wristbands. The results indicate that this method can be considered viable, since the implemented model was able to achieve an average accuracy of 86.35%.

2.5.6 Results comparison

The mentioned studies developed non-intrusive solutions for the stress prediction problem. Most of those works use wearable devices to retrieve the physiological signals, except for the articles in 2.5.3 and 2.5.4 (Gao, Yüce and Thiran, 2014; McDuff, Gontarek and Picard, 2014) which follow a remote approach, by camera. All the solutions are dependent of HRV measures, while the case in 2.8.4 (Gao, Yüce and Thiran, 2014) relies solely on facial expressions to detect stress.

For this comparison, in the case of (Can *et al.*, 2019), the person-specific models were not considered. The solutions and respective accuracies, signals and methods used are displayed in Table 2.

Table 2 – Comparison of the selected solutions

Solution	Best achieved accuracy	Physiological signal	Method
(Dalmeida and Masala, 2021)	83.33%	HRV	Wearable Sensor
(Can *et al.*, 2019)	92.15%	HRV	Wearable Sensor
(McDuff, Gontarek and Picard, 2014)	85%	HRV	Camera
(Gao, Yüce and Thiran, 2014)	90.5%	Facial Expressions	Camera
(Park, Kim and Kim, 2018)	86.35%	HRV	Wearable Sensor

Although the methods and signals used can differ between the projects, the obtained results were constantly good. The setting that had the most achieved accuracy was performed by (Can *et al.*, 2019), which worked with HRV extracted from wearable devices. This solution was followed by (Gao, Yüce and Thiran, 2014) which attained a 90.5% accuracy by analyzing the facial expressions of the subjects through a camera. The solution proposed by (McDuff, Gontarek and Picard, 2014) managed to achieve a model with 85% accuracy based on HRV extracted from a camera, while the solution developed by (Dalmeida and Masala, 2021) only obtained an accuracy of 83.33% by using HRV measures acquired from wearable devices. In this case, it proves that the combination of the method and physiological signal used is not necessarily related to the obtained results, indicating that other factors such as datasets, algorithms or hardware used can make a difference in the results.

Another note worth mentioning is that the use of the accuracy metric can mislead the perception of the results if the proportion of stress/non stress are imbalanced. Although not as mentioned, the recall of the models can also be beneficial to evaluate solutions related to stress, since it indicates the percentage of true positive cases that the model can detect.

3 Value Analysis

In this chapter, the project will be introduced with a Value Analysis approach. The Value Analysis of the project consists in an essential procedure to increase the value of an item or service at the lowest cost without sacrificing quality (Hughes and Chafin, 1996). The New Concept Development (NCD) Model (Koen *et al.*, 2001) will be the chosen model in this Value Analysis assessment and will be thoroughly defined and followed.

3.1 New Concept Development Model

The NCD model consists in a good practice to the development and innovation of products. It is divided in three key parts: The most noticeable one being the engine, which represents the leadership, culture and business strategies of an organization that powers the five key elements (Dewulf, 2013), the five key elements of the Front End of Innovation (Idea Genesis, Idea Selection, Concept & Technology Development, Opportunity Analysis and Identification) and the influencing factors, which refers to elements such as: the Organizational Capabilities, Business Strategy, Outside World (e.g. distribution channels, customers and competitors), and the Enabling Science that will be utilized (Koen *et al.*, 2001). This model is illustrated in Figure 3.

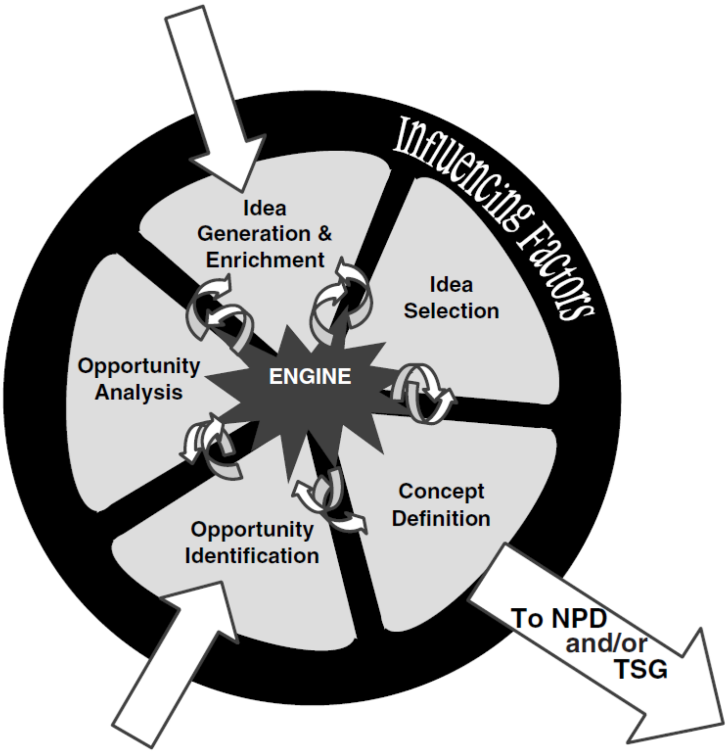


Figure 3 - New Concept Development Model (Koen *et al.*, 2001)

3.1.1 Opportunity Identification

Stress in workplaces has been growing globally mainly because of the rise of demanding jobs and increased pressure due to heavy workloads. It can also be caused by other factors such as: poor management in work, unpleasing working conditions or even lack of support from work colleagues and supervisors (World Health Organization, 2020).

The following graph shows the distribution of work-related stress, depression, or anxiety of the European countries, also distinguishing by gender. This study, performed by Eurostat, was based on data retrieved in 2020 and covers people from the European union, with age between 15 and 64 years (Eurostat, 2021).

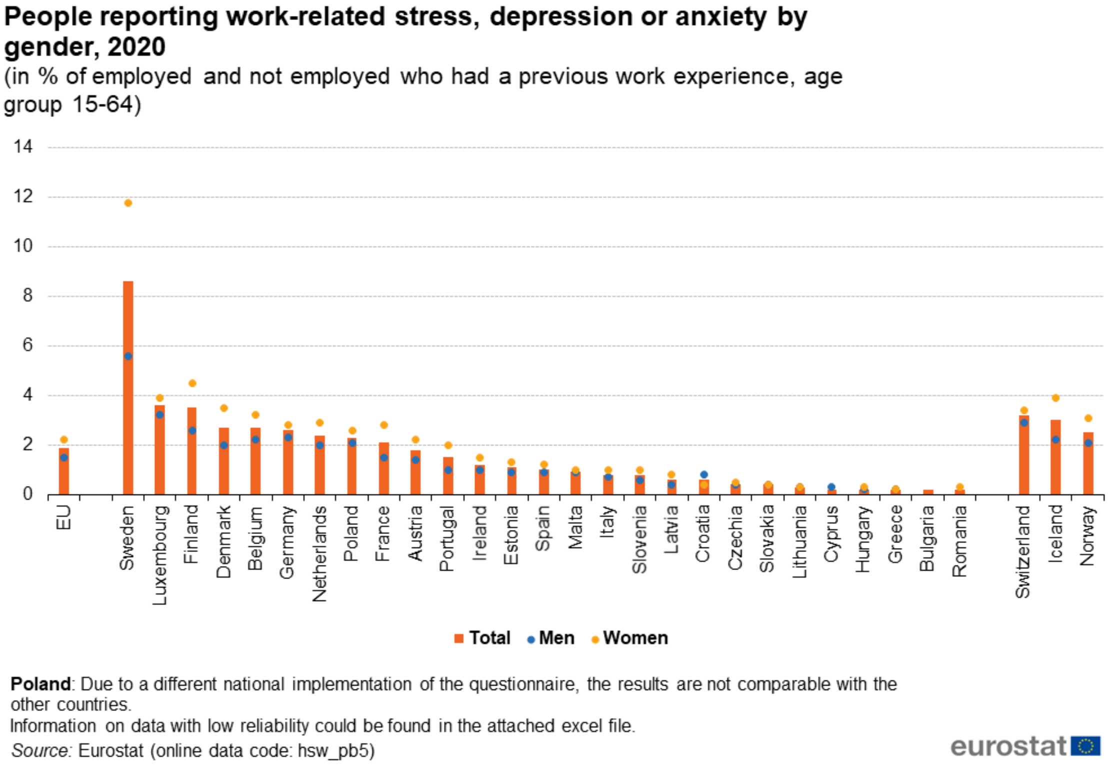


Figure 4 – People reporting work-related stress, depression or anxiety by gender in 2020 (Eurostat, 2021)

By analyzing Figure 4, it is noticeable that women, generally, have higher related percentages in comparison to men. It is also revealed that there are several countries that stand higher, comparatively with the EU, such as Sweden, Luxembourg, or Finland. Although Portugal stands a little below, it is still within the top ranked countries in this graph, which consists in a serious negative aspect.

As mentioned earlier, the effects of long-term exposure to stressful situations can be quite demanding on the body, provoking several types of serious health problems. Since most people spend a lot of hours working, this indicates that there's a high risk to their health if they tend to be stressed due to work. Another downside of this issue is that not only does it affect the employees' health, but also decreases their work performance, which affects the companies' products/services and forces them to either recruit new members or invest in occupational medicine.

Stress detection can be performed in several ways. Physiologically, the most used signals in stress detection are: Electrodermal Activity (EDA), Heart Activity and accelerometer (Can *et al.*, 2019). However, there is no defined test that can be done to detect stress in an individual, although being noticeable by analyzing potential symptoms or reactions displayed.

3.1.2 Opportunity Analysis

A recent study performed by the World Health Organization estimated that the common mental disorders in employees accounted for US$ 140 billion per year in costs related to lost productivity in the WHO European Region alone and US$ 1 trillion a year in the global economy (World Health Organization/Europe, 2017).

According to the European Agency for Safety and Health at Work (EU-OSHA), stress is the second most frequently reported work-related health problem and affects 22% of workers in Europe, as of 2005 (European Agency for Safety and Health at Work., 2010). It's also noticeable that there was a rise on work stress. Recent studies prove that there is an increase in long-term job strain from 1995 to 2015 (Rigó *et al.*, 2021).

In order to fight these concerns, there's a need to properly identify the stress in people, especially at work, so that countermeasures could be applied. To that end, the implementation of a platform capable of accurately detect stress and notify the users, can heavily benefit the workers by improving their lifestyle and combat health problems, as well as the companies/economy, given that the productivity of healthy and motivated employees is much likely to be higher.

3.1.3 Idea Generation & Enrichment

The idea generation phase of this project consisted in the investigation of several papers related to stress detection, in order to obtain insight on the different approaches already implemented. This search yielded useful information regarding the technologies necessary to create appropriate machine learning models, as well as the most predominant methodologies and physiological signals used in mental stress prediction.

3.1.4 Idea Selection

After an extensive search about stress prediction using physiological signals, it was possible to obtain an insight on the best practices, tools, and physiological responses to be used in this type of project.

HRV features were proved to be the most efficient way of predicting stress. Thus, this will be the most predominant predictor used in this thesis. The HRV features will be used in combination with eye blink frequency, pupil diameter and facial expressions to develop machine learning models.

Using a camera, an external program will retrieve the physiological signals and save them in value format. Based on the built machine learning models, these signals will be processed in order to produce stress predictions.

A dataset will be created by collecting data over the course of two months. Subjects will be studied, while working, in two different ways: First, an application will be analyzing the subject's face in order to extract the earlier mentioned signals. Then, another application will be also retrieving data by asking relevant questions to the subject, such as the perceived stress level.

3.1.5 Concept Definition

The proposed solution should be able to precisely identify stress in individuals, in a work environment, through a non-invasive approach, and consecutively notify them. The input data will be retrieved by an external application, using a camera to detect the physiological signals used.

3.2 Value Creation

In order to develop a successful product, there's a need to elaborate not only on the product itself, but also on how that product will be delivered to the customers and result in revenue for the creators.

The main purpose of this project is to present customers with a reliable application to detect stress and notify them about their stress levels. With this project, it's possible to define two stakeholders: the companies, which can adopt the system in their offices, and the employees. The companies can benefit from the implementation of this system due to the increased productivity that will result from the employees' better mental health, as well as a drop in costs from sick leaves. On the other hand, an employee that uses these systems will also be benefited by getting an improvement on their mental health, which can lead to an increased motivation and wellbeing, while also reducing the chances of contracting severe physiological diseases such as cardiac problems.

To properly organize the key concepts of this project, a business model was developed by recurring to a business model canvas (BMC). BMCs are intended to make business modelling more simple and easily comprehensible, while also capturing the complexity of how enterprises work (Qastharin, 2015).

In Figure 5 the value of the proposed project is presented through a BMC, according to (Osterwalder and Pigneur, 2010).

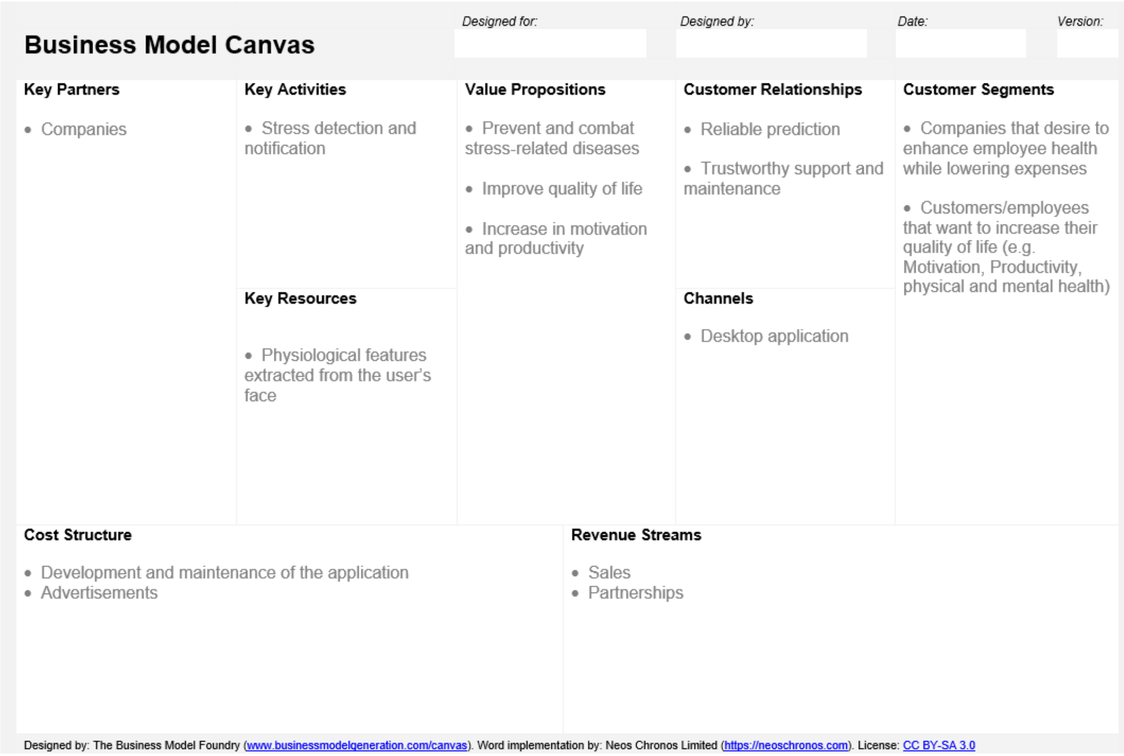


Figure 5 – Business Model Canvas

3.2.1 Value Proposition

A value proposition is an overall view of a company's bundle of products and services that are of value to the customer (Osterwalder, 2004). The value proposition canvas consists in a tool created by Alex Osterwalder, used to break down the problem of identifying the value proposition into discrete parts (Mansfield, 2019). A value proposition canvas was elaborated for this project, in order to complement the previously mentioned BMC, since it consists in a more in-depth vision of the "value propositions" block of the BMC. This value proposition canvas is demonstrated in Figure 6.

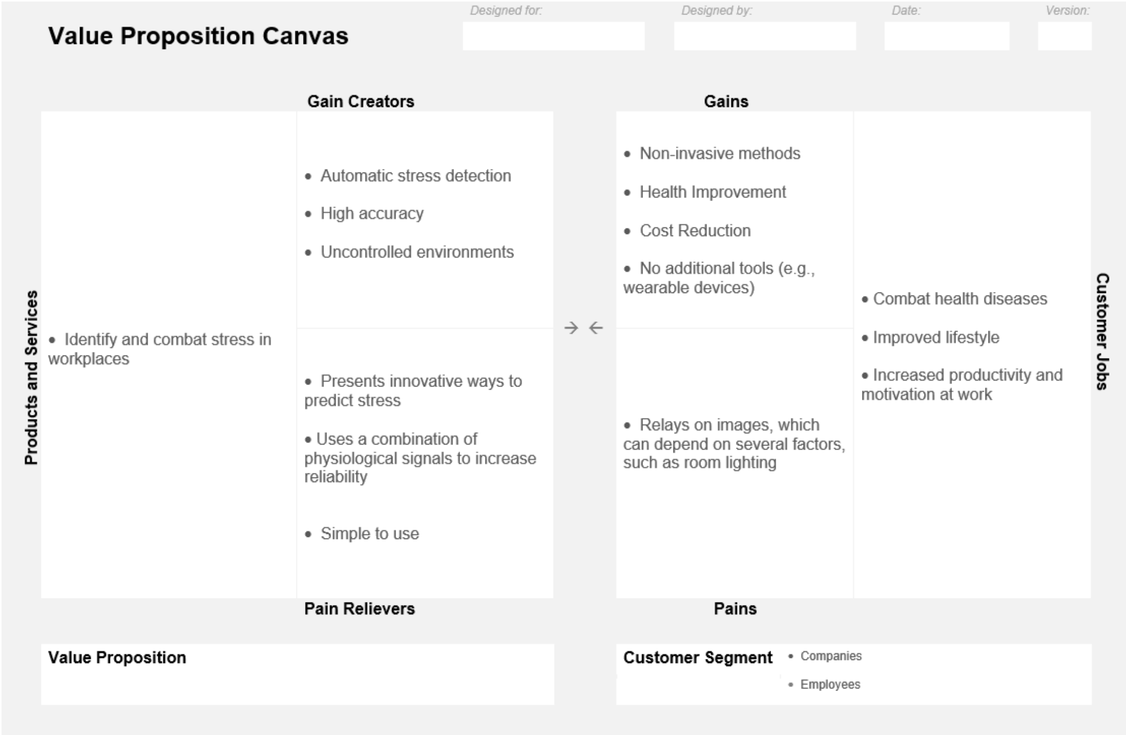


Figure 6 – Value Proposition Canvas

This model presents two sides: The customer (customer jobs) and the product (products and services). The customer is composed by the gains, which refers to the benefits and delights that are captured, and the pains, that represent the negative points of the product. On the other hand, the product is composed by the gain creators and pain relievers. The gain creators explain how the gains are delivered, while the pain relivers refer to the ways to attenuate the customer's pains (Mansfield, 2019).

3.3 Definition and Evaluation of Alternatives

The Analytic Hierarchy Process (AHP), proposed in 1980 by Thomas L. Saaty, is a model built to aid the decision-making problems. It consists in a three-part process which includes identifying and organizing decision objectives, criteria, constraints and alternatives into a hierarchy; evaluating pairwise comparisons between the relevant elements at each level of the hierarch; and the synthesis using the solution algorithm of the results of the pairwise comparisons over all the levels (Saaty, 1988).

The proposed alternatives to be utilized in this method consist in three libraries for the Python programming language. The chosen libraries are: pyHRV, HeartPy and hrvanalysis which are used to aid the development of solutions by offering several HRV-related functions. The defined criteria consist in:

- Documentation – The detail and legibility of the frameworks' documentation
- Usability – How easy it is to implement and use the frameworks
- Functionality – The different number of functions and methods available
- Reliability – How reliable are the frameworks to comply with the objectives of the project

The AHP method was performed based on the earlier mentioned alternatives and criteria, resulting in the following hierarchical tree:

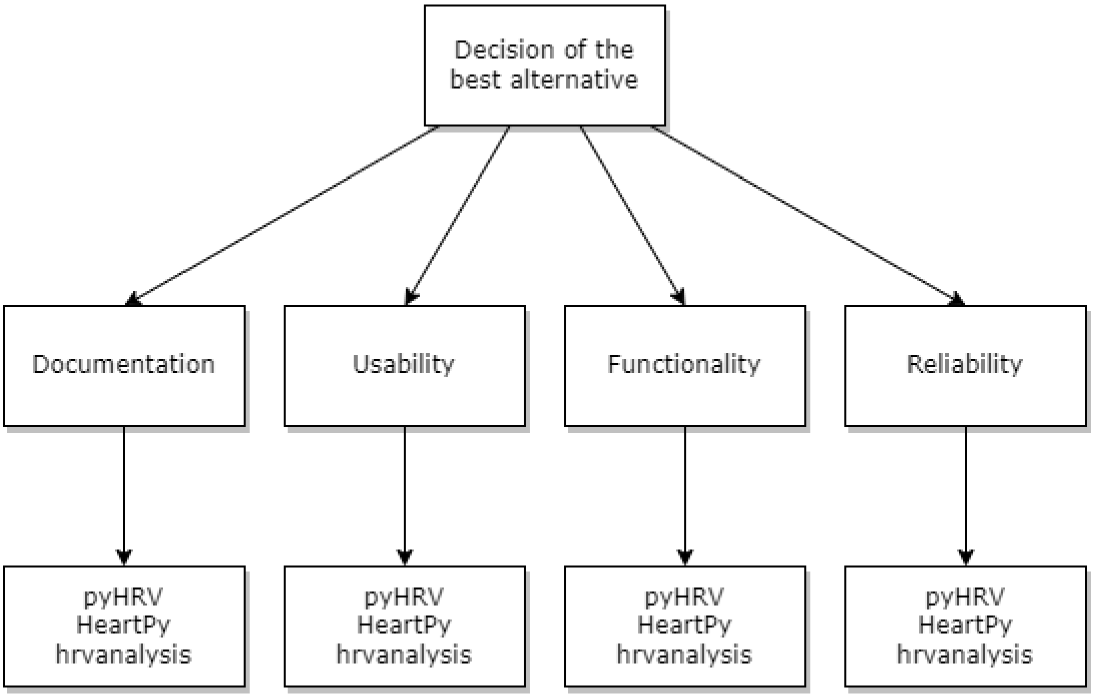


Figure 7 – AHP decision tree

The next step consists in defining the priorities between each criteria pair through a comparison matrix. In this task, the fundamental scale of absolute numbers (Saaty, 2008) will be used. The values on this scale range between 1 and 9, where 1 corresponds to an equal importance between criteria and 9 implies that a criterion is of extreme importance, over the other. This task is represented in Table 3.

Table 3 – Pairwise comparison matrix

	Documentation	Usability	Functionality	Reliability
Documentation	1	5	2	2
Usability	1/5	1	1/3	1/5
Functionality	1/2	3	1	1/2
Reliability	1/2	5	2	1
Sum	2.20	14	5.33	3.70

Then, it is necessary to normalize the values in the previous matrix and consequently calculate the relative priority of each criterion. The result consists in a preference vector that identifies the order of importance of each criterion.

To perform the normalization, the values from each column are divided by the sum. Then, the relative priority is obtained by calculating the arithmetic mean of the values of each row. The result of this process is demonstrated in Table 4.

Table 4 – Normalized matrix with calculated relative priority

	Documentation	Usability	Functionality	Reliability	Relative priority
Documentation	0.45	0.36	0.38	0.54	0.43
Usability	0.09	0.07	0.06	0.05	0.07
Functionality	0.23	0.21	0.19	0.14	0.19
Reliability	0.23	0.36	0.38	0.27	0.31

The relative priorities obtained in the previous step indicate that the most important criterion is the Documentation, followed by Reliability, Functionality and lastly, Usability.

In order to verify the consistency of the judgements, a consistency test is performed. Thus, the consistency ratio (CR) needs to be calculated. To obtain this value, it is necessary to obtain the λmax, which is obtained by multiplying the preference vector (Table 4) by each line; dividing the products obtained by the corresponding relative priorities; and calculating the mean value of the results obtained in the last operation. In this case, the λmax has a value of 4.06.

With the λmax already calculated, it is necessary to obtain the Consistency Index (CI), which is obtained by the following formula:

$$IC = \frac{\lambda \max - n}{\mathrm{n} - 1}$$

By applying the mentioned formula, the CI obtained had the value of 0.02.

The last step of the consistency test is to obtain the CR by using the CI and the Random Index (RI). The RI is a value that depends on the number of present criteria. In this case, since there are 4 criteria, the RI value is 0.9. The formula to calculate the CR is as follows:

$$CR = \frac{CI}{RI}$$

According to the mentioned formula, the CR obtained has a value of 0.02. In order to prove the consistency of the judgments, the CR must have a value of 0.1 or lower. In this case, the CR is lower than 0.1, which means that the defined values for the criteria are consistent.

To conclude the AHP method, a comparison matrix is created to compare the alternatives according to each criterion. The relative priorities obtained in the matrices will be calculated to decide which is the best alternative.

Table 5 – Comparison matrix for the Documentation criteria

Documentation	pyHRV	HeartPy	hrvanalysis	Relative Priority
pyHRV	1	0.33	3	0.25
HeartPy	3	1	6	0.65
hrvanalysis	0.33	0.17	1	0.1
Sum (Total)	4.33	1.5	10	

Table 6 – Comparison matrix for the Usability criteria

Usability	pyHRV	HeartPy	hrvanalysis	Relative Priority
pyHRV	1	0.5	0.25	0.14
HeartPy	2	1	0.5	0.29
hrvanalysis	4	2	1	0.57
Sum (Total)	7	3.5	1.75	

Table 7 – Comparison matrix for the Functionality criteria

Functionality	pyHRV	HeartPy	hrvanalysis	Relative Priority
pyHRV	1	3	4	0.62
HeartPy	0.33	1	2	0.24
hrvanalysis	0.25	0.5	1	0.14
Sum (Total)	1.58	4.5	7	

Table 8 – Comparison matrix for the Reliability criteria

Reliability	pyHRV	HeartPy	hrvanalysis	Relative Priority
pyHRV	1	4	6	0.69
HeartPy	0.25	1	3	0.22
hrvanalysis	0.17	0.33	1	0.09
Sum (Total)	1.42	5.33	10	

The last operation involves the multiplication of the relative priorities of each alternative by the relative priorities of each criterion, the results are the following:

$$pyHRV = (0.25 \times 0.43) + (0.14 \times 0.07) + (0.62 \times 0.19) + (0.69 \times 0.31) = 0.45$$

$$HeartPy = (0.65 \times 0.43) + (0.29 \times 0.07) + (0.24 \times 0.19) + (0.22 \times 0.31) = \ \ 0.41$$

$$hrvanalysis = (0.1 \times 0.43) + (0.57 \times 0.07) + (0.14 \times 0.19) + (0.09 \times 0.31) = 0.14$$

According to the AHP method, the best framework in this scenario is pyHRV. Thus, this will be the chosen alternative in the development of the project.

3.4 Function Analysis System Technique (FAST)

The FAST technique was improved in 1964 by Charles Bytheway, who provided a graphical representation and logical structure to the function analysis step of the Value Methodology. This representation, known as FAST Diagram, provides a visualization of how functions are linked or how they work together in a system, in order to deliver the intended service. To do that, the diagram organizes the required functions into a How?/Why? Relationship (Borza, 2011). The FAST diagram elaborated for this project is demonstrated in Figure 8.

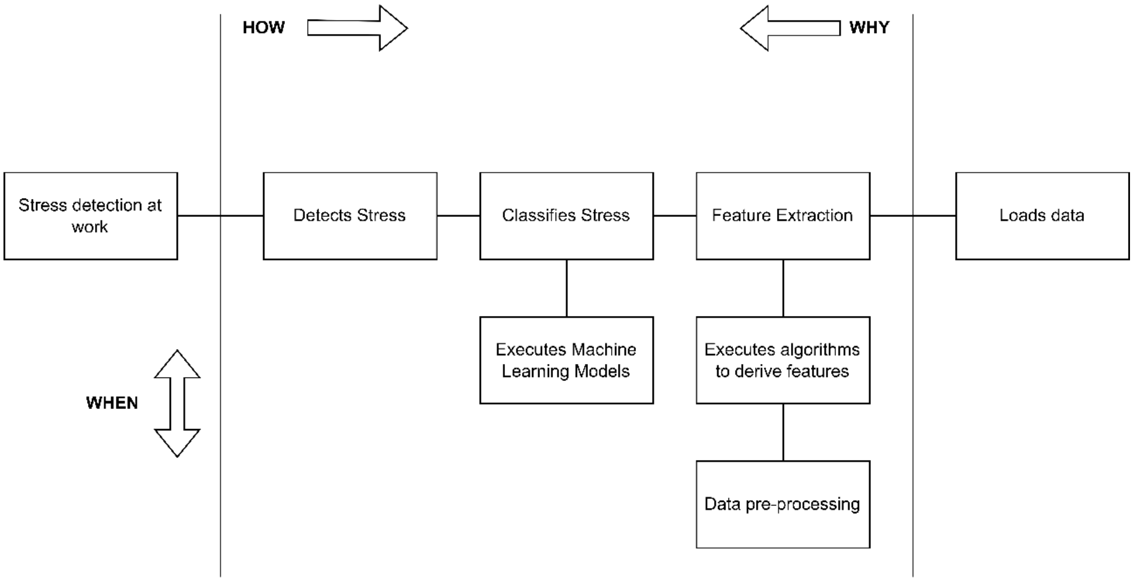


Figure 8 – FAST Diagram

As presented in the previous Diagram, the proposed system starts with the collection of the data as the input, on the right side, while the other end of the chart consists in the main goal of the system, which is the detection of stress at work.

Starting by the right side of the diagram, the function that relates to the collection of data (Loads data) is followed by a left-hand link to the "Feature Extraction" function, which indicates that the system loads the data to subsequently extract the features. In this stage, the "Feature Extraction" function has a downwards connection to another to functions. This shows when the functions are supposed to be executed. In this case, to properly obtain the desired features, the system first depends on the data pre-processing and the execution of algorithms to calculate physiological features. The same applies to the consequent functions of the system.

On the left side, the starting function is the detection of stress. This function establishes right-end links with the other functions to answer the "How" question posed. In this case, the relations suggest that, in order to detect stress, there's a need to classify and regress it first. Stress is considered as detected when the Machine Learning models predict a positive result for the stress label. The classification/regression also depends on the feature extraction, which itself first requires the loading of data from the external application.

4 Design

The proposed solution consists in four different main parts: data acquisition (Questionnaire and video applications), dataset creation (data pre-processing and feature extraction), stress prediction (implementation and evaluation of prediction models) and a stress detection web application. This solution involves the extraction and pre-processing of the data to further use in the prediction models, as well as the feature extraction from the raw data. Then, an additional step is required, which consists in the labeling of the data. After the necessary features are obtained, the prediction models are trained and evaluated. Lastly, the best performing model obtained in the stress Prediction component is then implemented into the web application to predict real cases of stress.

4.1 Requirements Analysis

In an early phase of the project, the fundamental parts of the solution were identified and properly addressed. In order to develop a final approach for this project, first, the requirements were subdivided into two categories: functional and non-functional requirements.

4.1.1 Functional Requirements

In this project's analysis, several functional requirements were identified. Each requirements are described through the use cases (UCs) found in the following table.

Table 9 – Identified UCs for the project

UC01	Collect physiological data with a video-based application
UC02	Collect data from a questionnaire application
UC03	Create dataset with physiological data labelled with stress/non-stress
UC04	Use the generated dataset to develop machine learning models for stress detection
UC05	Evaluate the obtained machine learning models
UC06	Implement the best performing model in a stress detection application

These UCs are also demonstrated through a UC diagram, in the following figure.

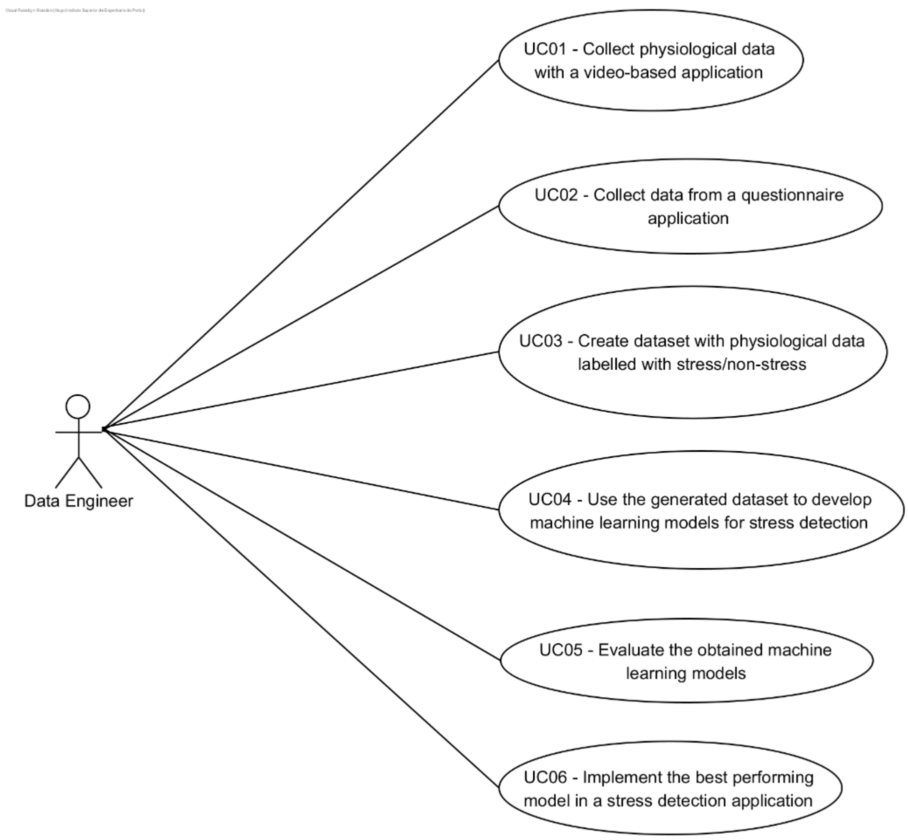


Figure 9 – UC diagram of the identified requirements

4.1.2 Non-functional requirements (FURPS+)

According to the FURPS+ acronym, the identified non-functional requirements for the system are the following:

Table 10 – Non-functional requirements (FURPS+)

Usability	• The UI of the applications should be user friendly
Reliability	• The stress detection application must be deployed on at least 2 different servers in order to reduce downtime in case of failure
Performance	• The classification of stress should be done quickly and effectively
Supportability	• The stress detection application should be implemented in a way to facilitate its maintenance and the addition of new features. • The stress detection application should be accessible from all web browsers
Implementation constraints	• The questionnaire application should be a desktop app

4.2 System Architecture

Once the requirement analysis of the project is complete, there’s a need to develop an appropriate solution for the project. Given the defined UCs, an approach was developed, which englobes all the necessary tasks in a simple, but effective manner. This approach includes the acquisition of the data, the creation of the dataset and the development and evaluation of stress classifiers.

The proposed architecture for this solution is demonstrated through the following component diagram.

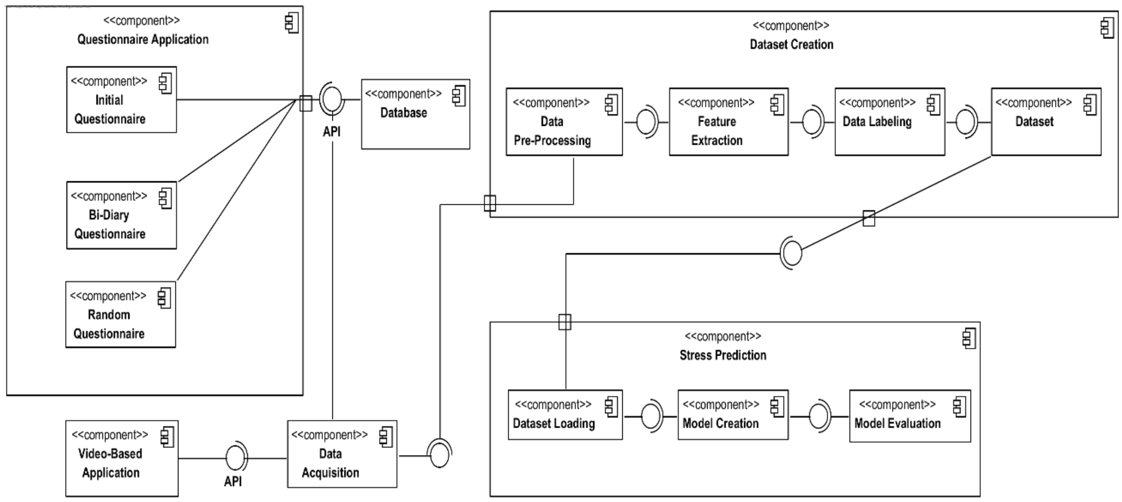


Figure 10 – Architecture of the proposed solution

This solution provides the necessary tools to develop a machine learning model for stress detection which is then implemented in a stress detection system.

4.2.1 Architecture Alternatives

For this project, the component diagram demonstrated in Figure 10 contains the architecture for the main part of this project. Therefore, an alternative design was elaborated for this architecture. The design presented in Figure 10 consists in a monolithic approach for the questionnaire application, which removes the backend application. This way, a single application will be responsible for both user interface (UI) and server, sharing the same programming language.

On the other hand, in the proposed alternative, the application is divided into two, the frontend and backend applications, which reduces the load on the user side. Moreover, it also facilitates the implementations of new features or bug fixes to the backend without the need of updating the application on the users' side. This alternative is represented in Figure 11.

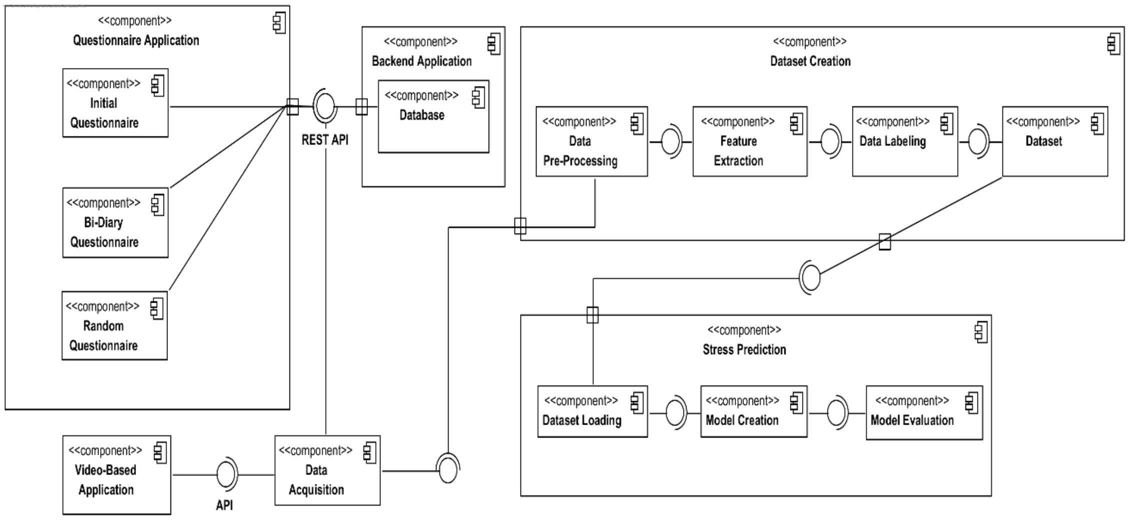


Figure 11 – Alternative architecture design

Although both design approaches are viable, for the mentioned reasons, this alternative (Figure 11) was chosen for this project.

4.3 Data Acquisition

The first part of the solution consists in the data acquisition from the participants of this study. In order to obtain trustworthy data to validate the stress prediction models, a desktop application was developed with the objective of collecting data, over the course of two months, from users while at work. A video-based application will also be running at the same time, with the objective of retrieving the physiological signals from the users, which will also be used to train the machine learning models. In this case, the questionnaires will complement the physiological data and serve as ground truth. All the data is sent to a database for further use. The application presents the users with three different types of questionnaires.

The first one only has personal questions, such as age, gender, weight, height, role in the company and marital status. Questions related to their habits are also presented. These include number of coffees or alcoholic beverages taken per day, cigarettes, sleep schedule, and so on.

Then, the bi-diary questionnaire is introduced. This form will appear as a pop-up at the lunch and leave hours, which are inserted by the user. Here questions are asked regarding their stress level during that part of the day, as well as emotional state and perception of their productivity. This form is concluded after the user answers the last question, which consists in a self-counting of their cardiac beats. In this task, the user presses the button to start the process and waits until the application reproduces a sound. Meanwhile, the user will count their beats silently, without manually confirming (e.g., Checking through wrist or neck) and will insert the number of beats counted when the sound is played. The time between the button press and the sound trigger is randomized between a pre-determined threshold (e.g., 8, 10 and 12 seconds), making the experiment more unpredictable and reliable.

Lastly, the third developed questionnaire consists in a simple pop-up that appears in the user's screen four times, surging two times in the morning and two times in the afternoon. This questionnaire asks the user about their stress level at the moment, which presents itself as a slider bar that ranges between 0 and 10, with the respective labels (e.g., "Not stressed", "Moderately stressed, "Extremely stressed"). This experiment also contains a certain degree of randomness. This is due to the fact that these pop-ups will appear at a random hour, between the user's entry time and the scheduled lunch/leave time, and the minimum difference between the generation of two questionnaires was set to 30 minutes. It was also imposed that these pop-ups may, at the most, appear 5 minutes before the bi-diary form.

The questionnaire data is accessed through a REST API that communicates with a backend application to connect to the database. A REST API consists in an application programming interface (API) that conforms to the constraints of representational state transfer (REST) architectural style and allows for interaction with RESTful web services (Red Hat, 2020). This was first defined in 2000 by computer scientist Dr. Roy Fielding in his doctoral dissertation (IBM Cloud Education, 2021).

In order to properly address the modules of this project, the Unified Modelling Language (UML) was used in the development of the diagrams for this project.

The modules for the data acquisition are exemplified through a component diagram in Figure 12.

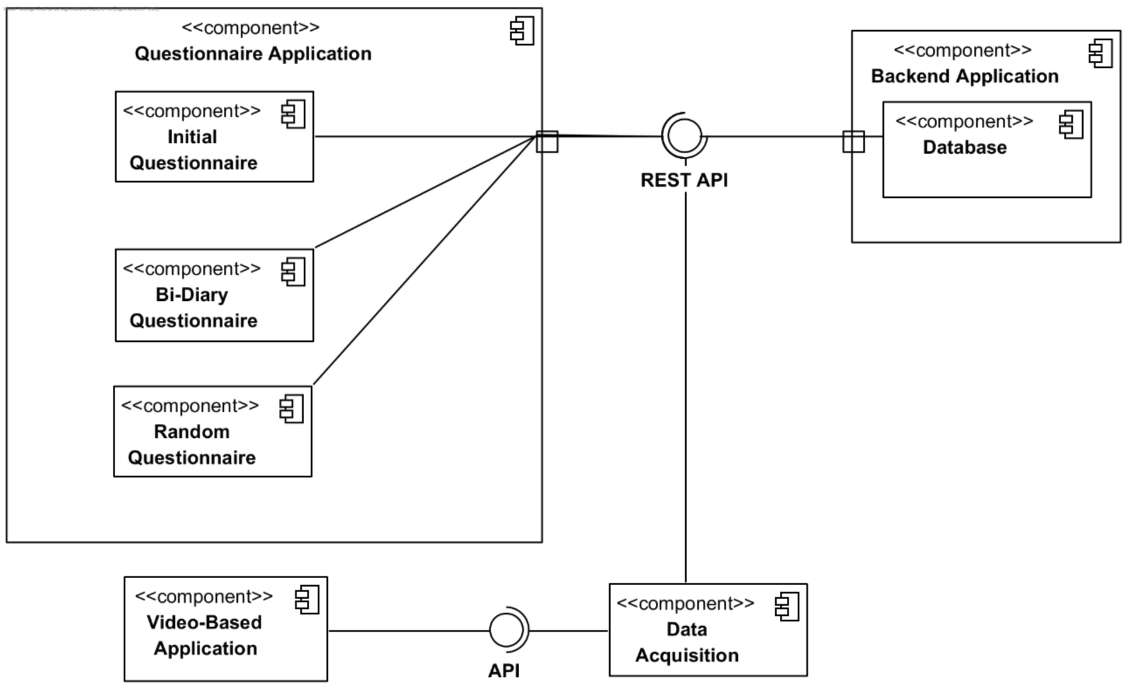


Figure 12 – Architecture for the data acquisition phase

4.4 Dataset Creation

The second part of this project consists in the creation of a dataset, which is going to be used in the training of the machine learning models. After the acquisition of the data from the participants, there are three steps required for this part: data pre-processing, feature extraction and data labeling. The data pre-processing for this collection of data consists in some simple steps, such as the removal of missing or incoherent values, deletion of unnecessary columns and data type conversion. To further improve the quality of the dataset, an algorithm was also developed to subdivide the physiological data files into shorter data. Since the retrieved data had uneven physiological recordings in terms of time duration, it was established that another division was necessary. Since the SDANN feature is calculated in segments of 5 minutes, it was established that the physiological data from the participants are subdivided into 5-minute readings, which benefits the correlation between the readings and the participants' stress. Since the SDANN feature is calculated in segments of 5 minutes, this was the value settled for the previously mentioned subdivision of the data.

Then, the feature extraction module is introduced. This part of the project consists in retrieving the necessary measures from the facial raw data obtained from the video application. The PERCLOS and facial expressions data are obtained directly from the files. However, the HRV data needs additional treatment. The retrieved files from the PPG application only provide the bpm readings of each second. So, these values are converted into NN intervals which are then used to extract the required HRV features. In order to remove incoherent data, it was also decided that the only accepted range for the bpm readings would be between 60 to 120.

After these steps comes the labeling of the dataset, which is done by merging the data from the video application with the labels from the questionnaire application. However, since the labels are not introduced at the exact same time as the measuring of the facial data, it's necessary that a label filling algorithm is implemented. So, the last step, before the dataset is created, consists in the labeling of the data by using a developed algorithm that fills the stress class with the nearest label available, based on the provided timestamps. Moreover, it is possible to define a maximum value of time difference accepted for this algorithm.

This part of the project is demonstrated through the component diagram in Figure 13.

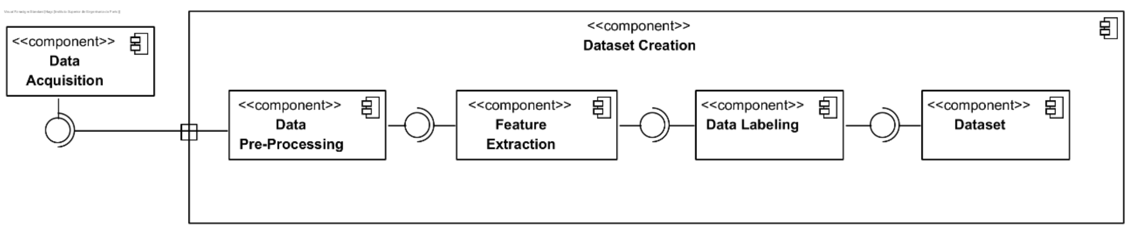


Figure 13 – Architecture of the dataset creation phase

4.5 Stress Prediction

The third part consists in the data mining phase, where the stress prediction models are created. This section interacts directly with the previously mentioned modules. First, the dataset is loaded. Then, several models are created with different attribute configurations and diverse algorithms (e.g., SVM, MLP). These models are properly evaluated and compared, in order to decide which alternative is the best for this scenario. The architecture for this part of the project is explained through a component diagram in Figure 14.

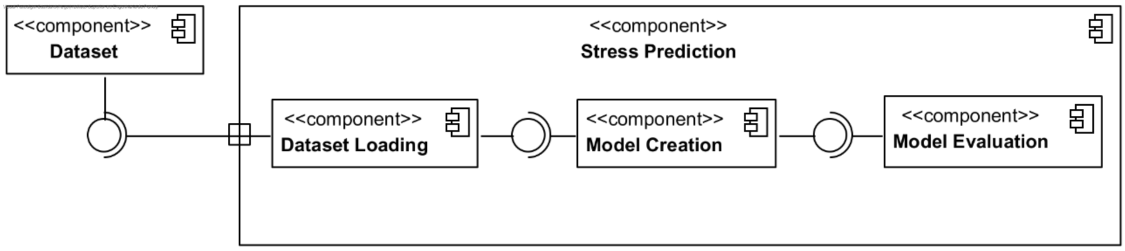


Figure 14 – Architecture of the data mining phase of the project

In the previously mentioned diagrams, the Questionnaire Application, Video-Based Application, Backend Application, Dataset Creation and Stress Prediction modules represent containers that include the components inside (if there are any).

4.6 Stress Detection Application

The last part of this project consists in the stress prediction application, which is used to test the developed models in real life cases. This application uses the machine learning models trained with the previously described dataset and an external video-based application that will collect the facial features of the users.

This application englobes the several modules presented earlier to effectively predict stress. First, the data is extracted and loaded into the app. Then, the data is pre-processed, and the required features are extracted. After the data is properly arranged, it is sent to the stress prediction module, which returns its result to the application, displaying it for the user. In this case, the prediction module will classify the stress with a binary classification, being either "stress" or "no stress".

According to the AHP method performed in section 3.3, the best toolkit for the HRV-related feature extraction is pyHRV. Thus, this will be the alternative used in the HRV feature extraction module.

This process is represented by a sequence diagram in Figure 15.

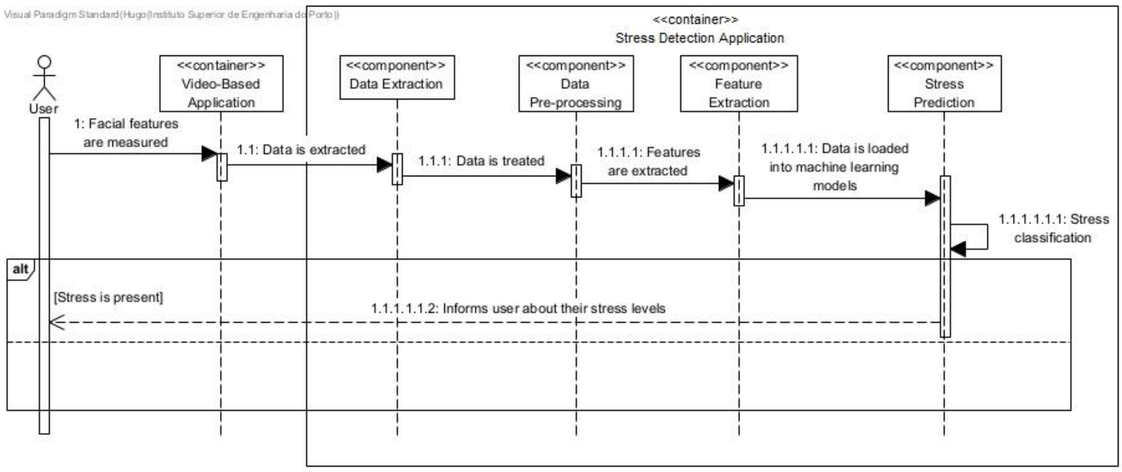


Figure 15 – Sequence diagram for the stress detection process

5 Implementation

In this chapter, all of the key implementations necessary for the development of this project are referred and thoroughly explained, being contextualized in this dissertation. The project englobes several main modules, such as the questionnaire application, machine learning model development and the stress detection web application.

5.1 Data Acquisition

In order to create a multimodal dataset for the machine learning models, two applications were used to collect the required information from the participants of this experiment. While the users are working, an external video application is silently storing the participants' physiological data. At the same time, a developed questionnaire application is also being executed in the background, which presents several questions to the user. The provided answers will later be used to label the physiological data and to predict the stress of the users.

The questionnaire application is composed by a frontend and backend application. The frontend was developed using Angular, Nodejs runtime and the Electron framework. With these tools, it is possible to develop a desktop angular application, which is extremely useful for this project. The respective backend was developed using the Elixir programming language and the Phoenix framework, hosted on the Gigalixir Platform (Gigalixir, 2022).

The DB for this application is non-relational and was developed using PostgreSQL and was divided in three tables, one for each form (initial, bi-daily and random questionnaires).

In the first interaction with the application, the inserted email will be used as an identifier of the user, which will be required to store the data in the database. The following image demonstrates the mentioned process.

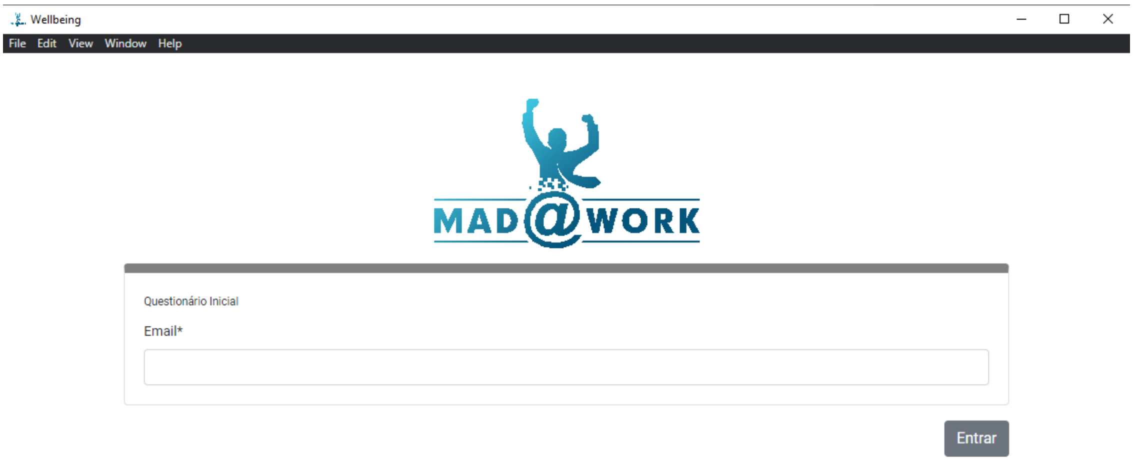


Figure 16 – User identification screen

After the email is inserted, the application checks if the associated email has a filled entry in the DB. If so, this questionnaire is skipped. If the user didn't fill the questionnaire previously, the questionnaire will appear in the next screen, according to the UI demonstrated in the next figure.

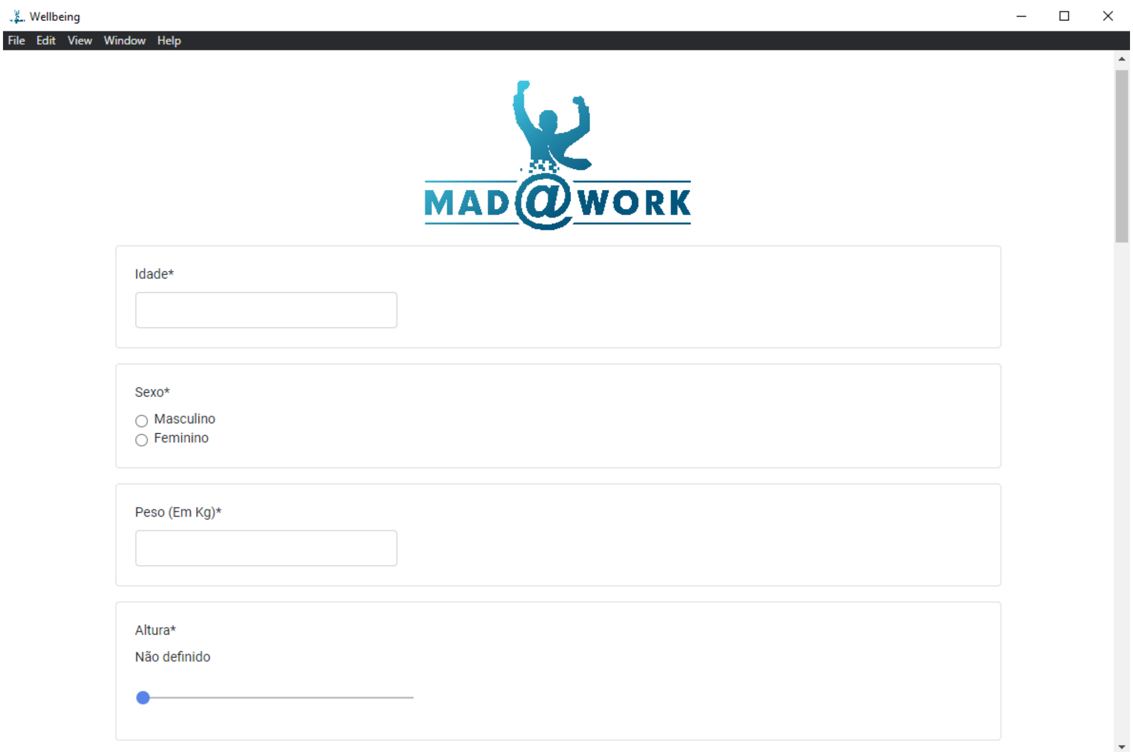


Figure 17 – Example of questions present in the initial questionnaire

The next screen presented consists in a UI where the user inserts the hours of their lunch break, as well as the leave hours. When all the fields are filled, the user saves the settings and can use the computer with the application being minimized. This screen is demonstrated in the following figure.

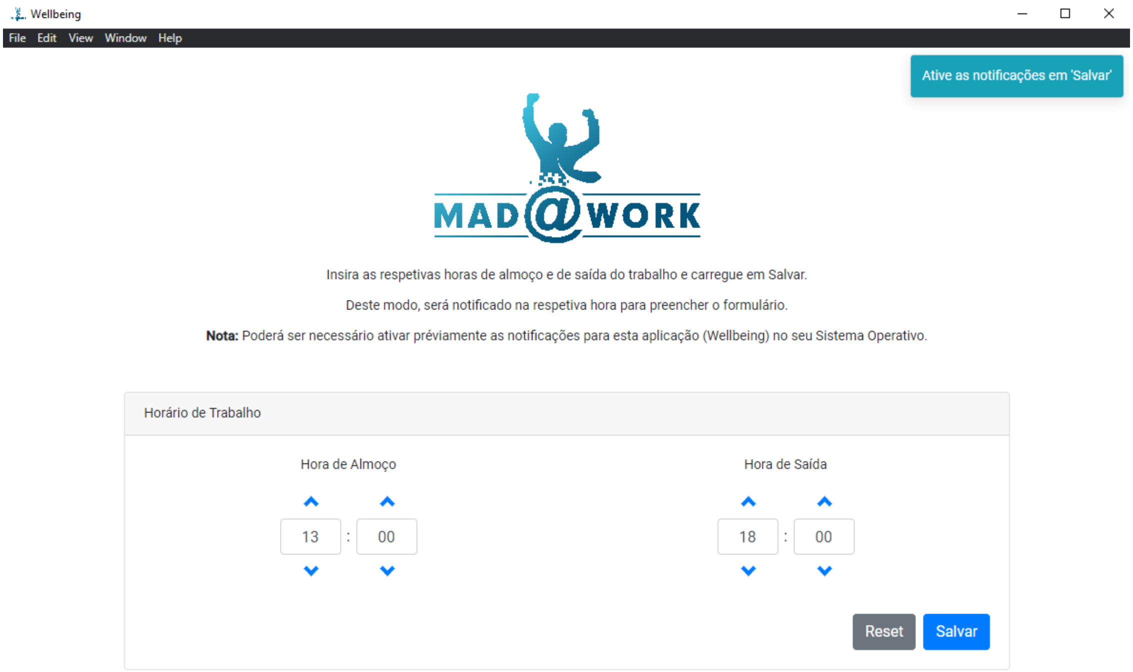
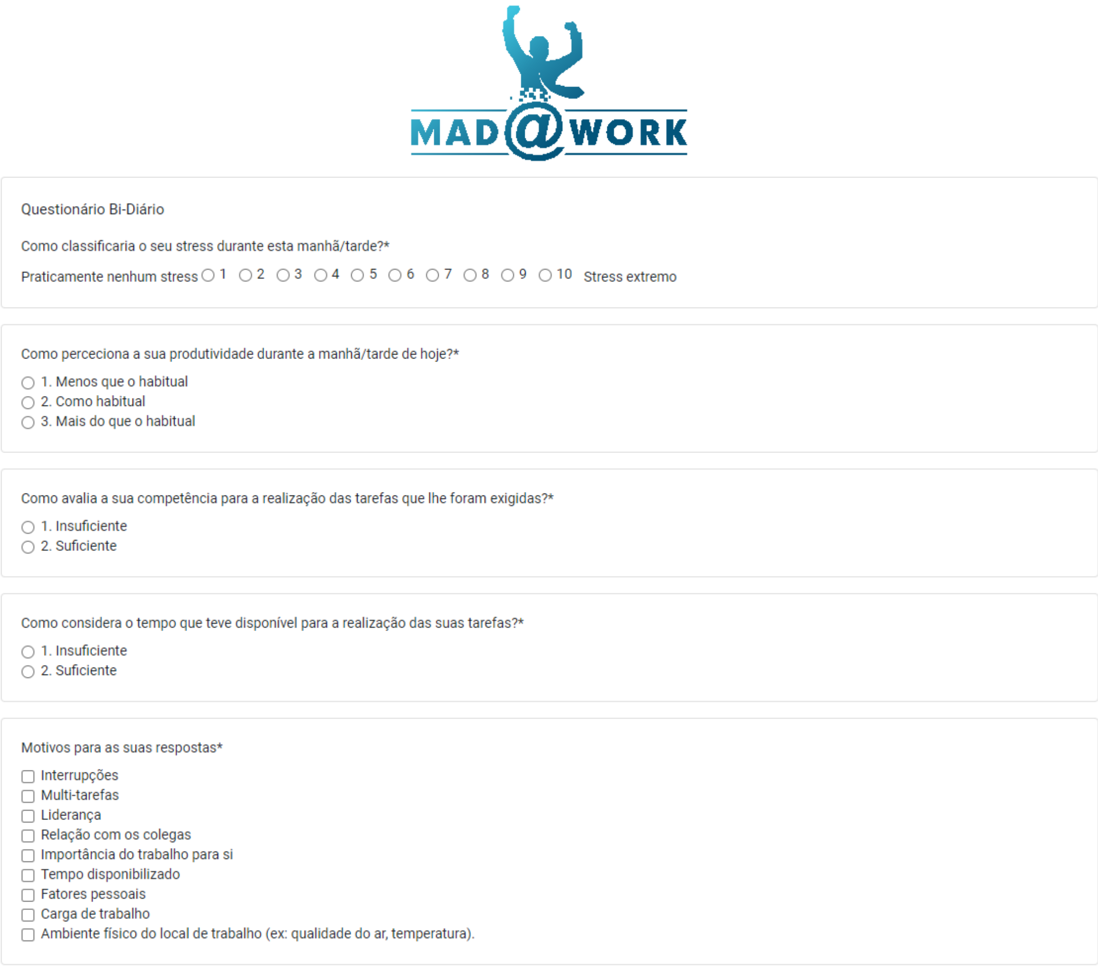


Figure 18 – UI for the lunch and leave hours

The next screen consists in a bi-daily questionnaire that is triggered at the lunch and leave hours, according to the information inserted by the user in the previous UI. This questionnaire presents the user with several questions related to their stress, productivity, and so on. An example of this questionnaire is demonstrated in Figure 19.



MAD@WORK

Questionário Bi-Diário

Como classificaria o seu stress durante esta manhã/tarde?*

Praticamente nenhum stress ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 Stress extremo

Como perceciona a sua produtividade durante a manhã/tarde de hoje?*

○ 1. Menos que o habitual
○ 2. Como habitual
○ 3. Mais do que o habitual

Como avalia a sua competência para a realização das tarefas que lhe foram exigidas?*

○ 1. Insuficiente
○ 2. Suficiente

Como considera o tempo que teve disponível para a realização das suas tarefas?*

○ 1. Insuficiente
○ 2. Suficiente

Motivos para as suas respostas*

☐ Interrupções
☐ Multi-tarefas
☐ Liderança
☐ Relação com os colegas
☐ Importância do trabalho para si
☐ Tempo disponibilizado
☐ Fatores pessoais
☐ Carga de trabalho
☐ Ambiente físico do local de trabalho (ex: qualidade do ar, temperatura).

Figure 19 – UI for the bi-daily questionnaire

Once the user saves their preferences, the application will also calculate four different hours for the scheduling of the random questionnaires, two of which will be before and the other two after lunch. These scheduling also takes into consideration the pre-requisite of the minimum interval between each questionnaire being thirty minutes. In order to avoid a conflict between the scheduling of these forms and the bi-daily form, the application ensures that the random form will always be triggered at least five minutes before the lunch/leave hours.

The random questionnaire only presents the user with a slider, which ranges from 0 to 10, relatively to the stress perceived by the user. Once the value is selected and the "OK" button is pressed, this window automatically closes, and the user resumes their work activities. The UI for this form is shown in Figure 20.

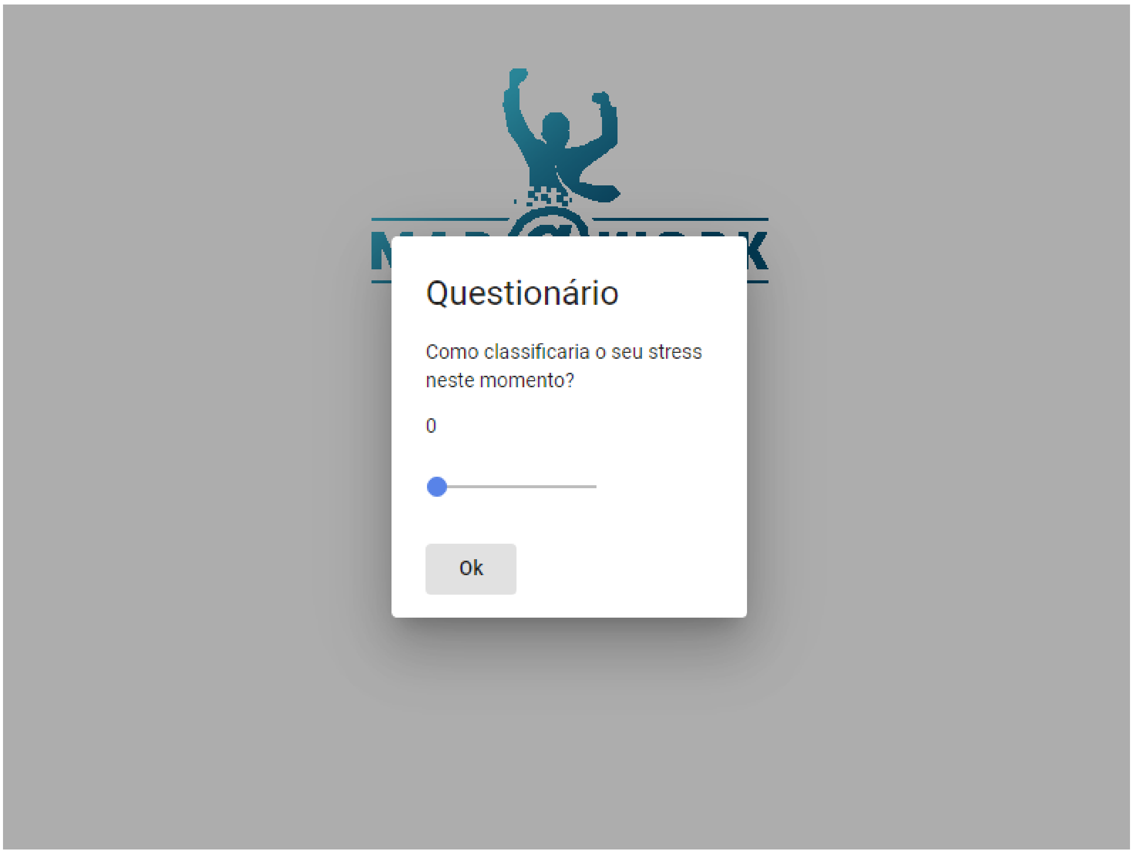


Figure 20 – UI for the random questionnaire

5.1.1 Security

The frontend application communicates with the backend application through http requests, which can sometimes lead to unauthorized use of the application by third parties, since it consists in a public API. To prevent the misuse of the application and to ensure that all the sensible data is well protected, several security features were added to the backend API.

The frontend application's goal is to send the users' questionnaire responses to the backend, in order to store them in the database. Therefore, the first security update would only allow the form submission methods in the API to be accessed by authorized users. In this case, only the desktop application will have access to submit responses. To perform this task, an API key was created and inserted in the environment of the desktop application of the participants. The key is sent in a specific header that is validated by the API.

The purpose of this study is to obtain relevant data from the participants, in order to develop machine learning models capable of detecting stress. However, to do this, we need to establish a secure connection to the database. To that end, the Secure Sockets Layer (SSL) encryption is enabled in the connection between the API and the database. Furthermore, the Hyper Text Transfer Protocol Secure (HTTPS) is always ensured, meaning that all the non-https requests are redirected to HTTPS. These changes significantly increase the security of the API by protecting the sensible data against potential attacks from outsiders.

SSL, and its successor, Transport Layer Security (TLS), are two protocols used for securing communication on the network by ensuring data confidentiality, data integrity and authenticity. The authentication of the communicating party and securing transfer of data is achieved through certificates, key exchange, and cipher suites (Sirohi, Agarwal and Tyagi, 2017). Thus, to further improve the security of the application, the API was redefined to only accept the latest TLS and SSL versions. In this case, the minimum TLS accepted version is TLS 1.2. Moreover, only the modern and strong ciphers are permitted when establishing a connection with the API.

The last implemented security feature consists in the introduction of an access control mechanism for the records in the database. To that end, the access to the database is protected behind a key validation, only generated for the administrators of the infrastructure. Also, only specific IP addresses are allowed to access the information in the database. These two mechanisms provide a substantial security level improvement.

5.2 Data pre-processing and Feature Extraction

For this study, after the data from the PPG and questionnaire applications are extracted, it is necessary to perform a thorough data treatment process on the collected information to ensure that the machine learning models will be trained using good quality data, which translates into well-performing models.

First, the data needs to be collected from all the sources, which consist in three different tables for the questionnaire data, and several comma-separated values (CSV) files, which contain the physiological readings of the users. Each file is associated with one user and only stores a certain amount of data. This means that a user can be associated with many of these csv files in one day, each referring to different time intervals of the day. The following images exemplifies how the csv files for a user are organized inside the folders.

2022-03-25_15-38-33	06/05/2022 16:07	Pasta de ficheiros
2022-03-25_15-44-47	06/05/2022 16:07	Pasta de ficheiros
2022-03-25_16-10-24	06/05/2022 16:07	Pasta de ficheiros
2022-03-29_14-05-17	06/05/2022 16:07	Pasta de ficheiros
2022-03-31_10-25-42	06/05/2022 16:07	Pasta de ficheiros
2022-03-31_10-25-58	06/05/2022 16:07	Pasta de ficheiros
2022-03-31_10-36-52	06/05/2022 16:07	Pasta de ficheiros
2022-03-31_11-20-58	06/05/2022 16:07	Pasta de ficheiros
2022-04-06_09-36-53	06/05/2022 16:07	Pasta de ficheiros
2022-04-07_08-48-40	06/05/2022 16:07	Pasta de ficheiros
2022-04-11_09-18-26	06/05/2022 16:07	Pasta de ficheiros
2022-04-14_10-11-41	06/05/2022 16:07	Pasta de ficheiros
2022-04-14_11-24-34	06/05/2022 16:07	Pasta de ficheiros
2022-04-27_09-34-36	06/05/2022 16:07	Pasta de ficheiros

Figure 21 – PPG data division in folders

blinksData	06/05/2022 15:04	Ficheiro de Valore...	1 KB
driverData	06/05/2022 15:04	Ficheiro de Valore...	48 KB
faceData	06/05/2022 15:04	Ficheiro de Valore...	52 KB
heartrate_signal	06/05/2022 15:04	Ficheiro de Valore...	22 KB

Figure 22 - Folder example relative to a user's PPG readings at a given time

After the data is loaded, the developed python script performs the basic pre-processing actions such as the removal of incoherent values, data normalization, data transformation, missing file/empty file verification, and so on.

As demonstrated in the previous images, the data obtained from the PPG application is subdivided in four different files for any given time interval. These files are stored in a folder which contains the starting date and time of the recording in its name.

By analyzing Figure 21 is possible to observe that the folders don't store the same amount of data. This can cause some incoherencies in the data, as some files contain readings for a large period of time. Using this data to extract the HRV features will result in less accurate readings. To that end, an algorithm was developed to re-organize the files by subdividing each csv file in 5-minute files, if possible. The result consists in much more files which only contain five minutes' worth of readings but improve the quality of the data used for the HRV feature extraction. The developed code for this task is exemplified in the following code extract.

```
interval = 5
while firstDate["TIME"].iloc[0]!=endOfFileDate:

 testeDataSet = testeDataSet[testeDataSet['TIME'].notnull()]
 firstDate["TIME"]=firstDate["TIME"].astype(str)
 nameOfFile = firstDate["TIME"].iloc[0].replace(" ","_")
 nameOfFile = nameOfFile.replace(":","-")

 testeDataSet["TIME"] =
 pd.to_datetime(testeDataSet["TIME"],errors="coerce")

 testeDataSetWithInterval =
 testeDataSet[testeDataSet["TIME"]<=finalDate.iloc[0]]

 testeDataSetWithInterval.to_csv(dividedByTimeFolder+"/"+id+"/"+nameOfFile+
 ".csv", sep=';')
 firstDate = testeDataSetWithInterval.tail(1) #last date of the previous
 file

 firstDate["TIME"] = pd.to_datetime(firstDate["TIME"],errors="coerce")

 finalDate = pd.to_datetime(firstDate['TIME'].astype(str)) +
 pd.DateOffset(hours=0, minutes=interval)

 testeDataSet =
 testeDataSet[testeDataSet['TIME']>firstDate["TIME"].iloc[0]]
```

Code 1 – Code relative to the file division by five minutes

To calculate the HRV features, there's a need to first obtain the NN intervals. Those were calculated by applying a simple formula to transform the beats per minute readings of every instance into NN intervals, in seconds. Moreover, a filter was applied in this phase which only keeps entries which have valid bpm readings. In this case, the acceptable range is between 60 and 120 bpm.

Usually, the NN intervals are measured in milliseconds, however, the intervals used in this project were in seconds, since the pyHRV documentation states that the hrv function can receive values as input in both seconds and milliseconds. Once calculated, the NN intervals are sent to the HRV module from pyHRV which returns the organized HRV features. The formula used to derive the RR (NN) intervals, from the bpm, according to a similar work in the area (Dalmeida and Masala, 2021), was the following:

$$RR = 60/bpm \quad (6)$$

The module responsible for the extraction of the HRV features receives a list as argument, which corresponds to the NN intervals of a person. To begin this process, the frequency bands are chosen according to the intervals defined by the Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology (Electrophysiology, 1996). The mentioned bands are shown in the code extract below.

```
fbands = {'ulf': (0.0, 0.003),
          'vlf': (0.0033, 0.04),
          'lf': (0.04, 0.15),
          'hf': (0.15, 0.4)}
```

Code 2 – Frequency bands definition

The time domain features are extracted by using the time domain module of the pyHRV library, as shown in the following code.

```
resultsSDNN = td.sdnn(list)
resultsRMSSD = td.rmssd(list)
resultspNN50 = td.nn50(list)
resultspNN25 = td.nnXX(list,threshold=25)
resultspNN20 = td.nn20(list)
resultsAVNN = td.nni_parameters(list)

resultsSDNN_RMSSD = resultsSDNN['sdnn']/resultsRMSSD['rmssd']
resultsSDNN = resultsSDNN['sdnn']
resultsRMSSD = resultsRMSSD['rmssd']
resultspNN50 = resultspNN50['pnn50']
resultspNN25 = resultspNN25['pnn25']
resultspNN20 = resultspNN20['pnn20']
resultsAVNN = resultsAVNN['nni_mean']
```

Code 3 – Time domain feature extraction

After the time domain metrics are retrieved, there's also the need to extract the frequency domain ones. To that end, the frequency domain module of the same library is used. This step is demonstrated in the following code extract.

```
frequencyDomainResults = fd.welch_psd(list,show=True,fbands=fbands)

hfLfRatio = frequencyDomainResults['fft_ratio']

hfNormalized = frequencyDomainResults['fft_norm'][0]
lfNormalized = frequencyDomainResults['fft_norm'][1]

lfRelative = frequencyDomainResults['fft_rel'][2]
hfRelative = frequencyDomainResults['fft_rel'][3]

lfPeak = frequencyDomainResults['fft_peak'][2]
hfPeak = frequencyDomainResults['fft_peak'][3]

totalPower = frequencyDomainResults['fft_total']

ulfAbsolute = frequencyDomainResults['fft_abs'][0]
vlfAbsolute = frequencyDomainResults['fft_abs'][1]
lfAbsolute = frequencyDomainResults['fft_abs'][2]
hfAbsolute = frequencyDomainResults['fft_abs'][3]
```

Code 4 – Frequency domain feature extraction

Once the features are calculated, they are inserted in a data frame and returned, as described in the following code.

```
datasetWithMetrics = pd.DataFrame({'AVNN': resultsAVNN,'SDNN':
resultsSDNN,'RMSSD': resultsRMSSD, 'PNN50': resultspNN50, 'PNN20':
resultspNN20, 'PNN25': resultspNN25,'Hf/Lf': hfLfRatio, 'Hf Normalized':
hfNormalized,'Lf Normalized': lfNormalized,'Total Power': totalPower, 'ULF
Absolute': ulfAbsolute,
'VLF Absolute': vlfAbsolute, 'LF Absolute': lfAbsolute, 'HF Absolute':
hfAbsolute, 'LF Relative': lfRelative,'HF Relative': hfRelative,'LF Peak':
lfPeak,'HF Peak': hfPeak}, index=[0])
return datasetWithMetrics
```

Code 5 – Return of the dataset with HRV features

After the HRV features are calculated, the face data and PERCLOS of the participants are also retrieved. For each file, corresponding to 5-minute readings, the mean of each facial expressions and PERCLOS values are calculated and saved.

Lastly, the treated video application and questionnaire data are loaded and properly merged in a data frame, in order to correlate the physiological data with the stress labels obtained from the questionnaires. The merging process is implemented as demonstrated in the following code extract.

```
completeData =
pd.merge(finalDf,completeFormDf.sort_values("timeStamp"),on=['id','timeStam
p'],how="outer")
```

Code 6 – Code relative to the merging process between the questionnaire and physiological data

Initially, the data was collected from 22 participants. However, after all the data preprocessing, the amount of usable data was heavily decreased. This way, the final version of the dataset only contains data from 19 different participants

5.3 Data Labeling

After the data from all sources is merged, it is noticeable that there are no rows with the stress label filled. This is due to the fact that each person only fills a maximum of six stress-related questionnaires a day. This way, the exact timestamp of the forms and the timestamps of the PPG readings will not be coincident. Thus, it is not possible to directly merge all the data through the indicated timestamps. To fix this issue, an algorithm was developed to transform the data frame and fill the labels with an approximate value.

The labeling algorithm will fill the stress column of each user with the nearest available entry in the data for that id, based on the timestamp present. It also takes in consideration the part of the day, being either before or after lunch. The program considers any entry after 1 pm to be after lunch and verifies each line to fill the column “halfOfday”, where 0 and 1 refer to before and after lunch respectively. The developed code for this process is demonstrated in the following code extract.

```
date =
closestDate(completeDataMatchingDates[(completeDataMatchingDates["date"]==r
ow['date']) & (completeDataMatchingDates["id"]==row["id"]) &
(completeDataMatchingDates["halfOfDay"]==row["halfOfDay"])]["timeStamp"],ro
w['timeStamp'])

stress =
completeData[completeData['timeStamp']==date]['responseStress'].values[0]

completeData.at[index,'responseStress'] = stress

def closestDate(dates, dateToSearch):
    return min(dates, key=lambda x: abs(x - dateToSearch))
```

Code 7 – Relevant code extract of the stress labeling algorithm

The code mentioned in the previous snippet works by calling the “closestDate” function inside a recursive loop. In this loop, the data frame is iterated using the “iterrows” function available on the pandas module, as exemplified in Code 8.

```
for index,row in completeData.iterrows():
```

Code 8 – Code relative to the iteration of a dataframe

By calling the “closestDate” function inside the loop, it is possible to obtain, for every row that is missing a stress label, an approximate value of stress for the corresponding person, in the same day and part of day. This function receives two arguments: the first one consists in a pandas series containing all of the available dates that possess a valid stress label; the next contains the timestamp of the row which needs to be filled with a stress value. To properly implement this algorithm, the “closestDate” method is called by sending a timestamp series only containing the already filtered entries that match the previous mentioned criteria (date, id, half of the day). The method then returns the nearest timestamp available, given the matching criteria. Based on the returned timestamp, the corresponding stress value is obtained and is then inserted in the corresponding index. Additionally, an optional filter was implemented which only allows entries within certain minutes/hours of the timestamp of the rows missing the stress label.

In order for the labels to be applied to a binary classification context, the stress values need to be converted. For this study, given the data obtained, it was decided that the label is considered to be “stress” when the value is at least 3, in a scale from 0 to 10, which translates into a more cautious classification that aims to minimize false negatives.

5.4 Model Creation and Evaluation

To evaluate the developed machine learning models, a cross validation was implemented using the stratified K-fold. In all experiments, the value for k was always 10, since it provides good results with a moderate load on the processing unit.

The data used in the experiment is moderately balanced, which excludes the need of oversampling techniques.

In the performed experiments, the HRV variables used are demonstrated in the following table:

Table 11 – HRV features used in the experiments

AVNN	Average of NN intervals
SDNN	Standard deviation of the NN intervals
RMSSD	Root mean square of successive differences between normal heartbeats
SDNN_RMSSD	Ratio of SDNN to RMSSD (Nkurikiyeyezu, Yokokubo and Lopez, 2020)
PNN50	Percentage of adjacent NN intervals that differ from each other by more than 50 milliseconds
PNN25	Percentage of adjacent NN intervals that differ from each other by more than 25 milliseconds
PNN20	Percentage of adjacent NN intervals that differ from each other by more than 20 milliseconds
HF/LF	HF/LF ratio
HF Normalized	HF Normalized Power
LF Normalized	LF Normalized Power
Total Power	Total Power of all frequency bands (ms^2)
ULF Absolute	ULF Absolute Power (ms^2)
VLF Absolute	VLF Absolute Power (ms^2)
LF Absolute	LF Absolute Power (ms^2)
HF Absolute	HF Absolute Power (ms^2)
LF Relative	LF Relative Power (%)
HF Relative	HF Relative Power (%)
LF Peak	LF Peak Frequency (Hz)
HF Peak	HF Peak Frequency (Hz)

Additionally, some answers to the questionnaires will also be used. The information used is as follows:

Table 12 – Attributes used from the initial questionnaire

Age	Age of the Participants
Gender	Gender of the Participants
Weight	Weight of the Participants
Height	Height of the Participants
Role	Role of the participants (senior manager, intermediate manager, employee)
MaritalStatus	Marital Status (single, married, divorced, widowed, separated)
Studies	Years of formal study attended (less than 9th grade, between 9th and 12th grade, higher education)
WakeUpHours	Preferred hours to wake up (5.00-6:30 AM, 6:30-7:45 AM, 7:45-9:45 AM, 9:45-11:00 AM, 11:00-12:00 AM)
FirstHalfHour	Wellbeing during the first half hour after waking up (Very tired, somewhat tired, somewhat invigorated, very invigorated)
SleepingHours	Hours at which the participants feel tired and need to sleep (8:00-9:00 PM, 9:00-10:15 PM, 10:15-10:45 PM, 12:45 PM-2:00 AM, 2:00-3:00 AM)
BestDisposition	Hours at which the participants usually feel at their best (5:00-8:00 AM, 8:00-10:00 AM, 10:00 AM – 5:00 PM, 5:00-10:00 PM, 10:00 PM – 5:00 AM)
TypeOfSchedule	Type of person, according to the participants' schedule (definitely a morning person, more of a morning person than a night owl, more of a night owl than a morning person, definitely a night owl)
Smoke	Average number of cigarettes smoked per day
Coffee	Average number of coffees consumed per day
Alcohol	Drinking habits (not drinking, occasionally, once a week, once a day, more than once a day at meal times only, more than once a day at meal times and other times)
AverageSleepingHours	Total hours of sleep per day, including naps
SleepQuality	Sleep quality in the last week (from 0 to 10)

Lastly, the facial expressions used in the experiments are anger, fear, disgust, happiness, sadness, surprise and neutral.

The eye blink and pupil diameter data had poor quality and provided poor correlation with the participants' stress levels. For that reason, these two features were removed from the experiments, only keeping the facial expressions, HRV, PERCLOS and the data from the initial form.

The SDANN is computed using segments of 5 minutes. However, in the available dataset, there were a great number of intervals that were under 5 minutes of length, which resulted in several missing values for this attribute. With this, it was decided that the SDANN would be removed from the experiment.

The Recursive Feature Elimination (RFE) was used as the feature selection algorithm, provided by the scikit learn library for python. This method takes an estimator as argument, to use in the ranking of the features, in this case, the decision tree was used. Also, the method receives the n number of desired best features to return from the RFE.

Then, an algorithm was developed to perform a cross validation on all the combinations of best features, ranging from 5 to 15 features, and algorithms. The chosen range for the number of best features was decided based on straightforward testing which indicates that, with less than 5 features, the generated models will underperform. Additionally, no improvements were noticeable when using more than 15 features. So, to maximize the performance of the models and to reduce the load on the processing units, the defined range for the best features is between 5 and 15. In total 44 models were generated, which consisted in 4 algorithms versus the best 5 to 15 features, from which only the best three will be chosen for analysis.

Finally, a csv file is generated with a report of the data. For this experiment, the algorithms used were: Random Forest, AdaBoost, XGBoost and Gaussian Naïve Bayes. This cross validator was applied on different modifications of the original dataset.

5.4.1 First Experiment

The first change on the dataset consisted in a filter, with the value of thirty minutes, applied to the label filling algorithm. This means that each missing stress label will be filled with the label of the closest entry of the dataset for that ID. However, if there is no valid entry within thirty minutes from the missing entry, it will not be filled. Moreover, this version of the dataset only contains HRV and facial expressions data. The data distribution obtained in this experiment is shown in the following image.

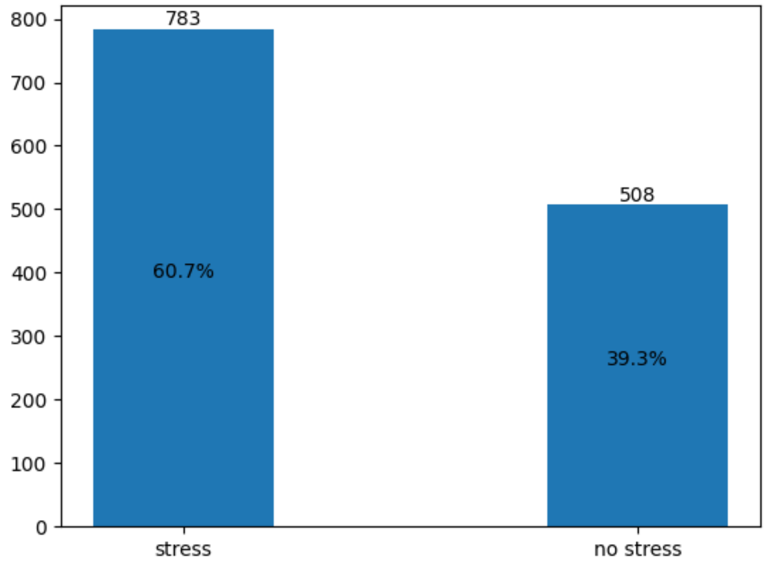


Figure 23 – Label distribution for the first experiment

With the implementation of the thirty-minute filter on the label filling algorithm it is possible to obtain data with improved quality. However, the improvement in quality, in this case, also translates into reduced data for the experiment.

The best results obtained in this experiment, using this dataset, are demonstrated in the following table.

Table 13 – Best results of the first experiment

Features	Algorithm	Accuracy	Precision	Recall	Balanced Accuracy	F1 Score
AVNN, SDNN, VLF Absolute, LF Absolute, LF Relative, Angry, Happy, Sad, Surprise, Neutral	XGBoost	75.68%	**75.79%**	88.13%	**72.31%**	81.47%
AVNN, SDNN, PNN20, Total Power, VLF Absolute, LF Absolute, LF Relative, HF Relative, Angry Fear, Happy, Sad, Surprise, Neutral	Random Forest	**75.91%**	75.14%	**90.42%**	71.98%	**82.02%**
AVNN, SDNN, VLF Absolute, LF Relative, Angry, Happy, Sad, Surprise, Neutral	Random Forest	75.60%	75.50%	88.77%	72.03%	81.54%

By analyzing the table above, it is noticeable that the best results always require a collection of data from the two available physiological signals in the experiments (HRV and Facial expressions). These features are selected by the RFE method, based on their correlation with the target variable (stress). It is also possible to observe that the majority of the attributes selected are common to the three best results.

The results demonstrated in Table 13 are very alike. However, the second combination manages to achieve slightly better results overall.

5.4.2 Second Experiment

In the next experiment, the dataset used to train the machine learning models is going to be slightly changed. This time, instead of the thirty minute maximum time difference used in the stress label filling algorithm, a maximum interval of one hour is going to be applied.

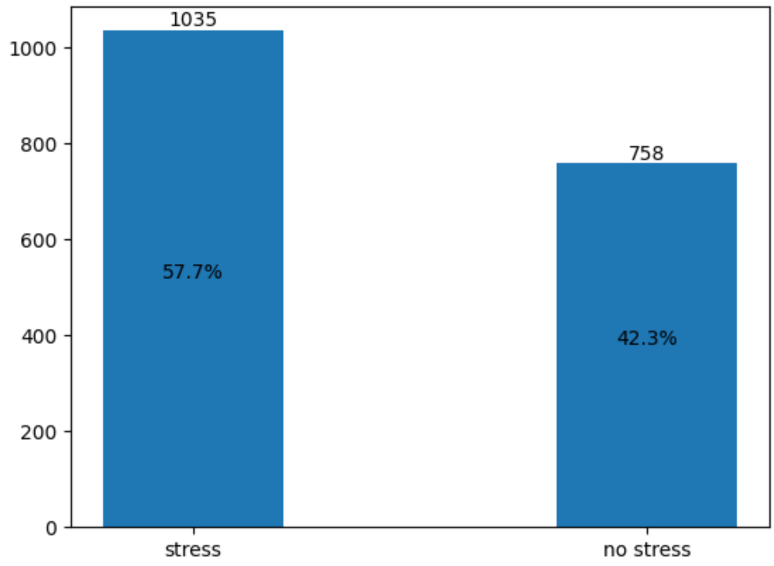


Figure 24 – Label distribution for the second experiment

By analyzing the dataset produced with the previously mentioned changes it is possible to observe that there was an increase in both positive and negative labels. Moreover, the balancing between both labels is better, when comparing to the label distribution in Figure 23. Since the maximum interval for the label filling algorithm is now doubled, it is predictable that, despite the increase of entries, the data is less accurate than before.

The results obtained with this change are demonstrated in Table 14.

Table 14 – Best results of the second experiment

Features	Algorithm	Accuracy	Precision	Recall	Balanced Accuracy	F1 Score
AVNN, SDNN, HF Absolute, Angry, Fear, Happy, Sad, Surprise	Random Forest	**74.51%**	**73.46%**	**87.64%**	**72.11%**	**79.88%**
AVNN, SDNN, HF Absolute, HF Relative, Angry, Fear, Happy, Sad, Surprise	Random Forest	74.01%	73.34%	86.57%	71.71%	79.38%
AVNN, SDNN, ULF Absolute, HF Absolute, HF Relative, Angry, Fear, Happy, Sad, Surprise	Random Forest	73.95%	72.96%	87.34%	71.50%	79.46%

The three best results proved to be very similar once again, using the same algorithm and sharing almost all the selected features. The first configuration is undoubtedly the best out of the three, while the other two are identical, since the values have almost unsignificant variance in all the metrics.

By comparing Table 13 and Table 14 it is possible to conclude that, by applying a maximum interval of thirty minutes to the stress labeling algorithm instead of 1 hour, the results have improved, despite having a slight decrease in data size. So, in the following experiments, this interval will be maintained.

5.4.3 Third Experiment

Following the previous results, another experiment was performed using the same settings and features of the dataset from the first experiment, while also adding the PERCLOS values of the subjects.

The results obtained in this process are demonstrated in the following table.

Table 15 – Best results of the third experiment

Features	Algorithm	Accuracy	Precision	Recall	Balanced Accuracy	F1 Score
AVNN, SDNN, Angry, Fear, Disgust, Surprise, PERCLOS	AdaBoost	**77.77%**	**78.35%**	87.75	**75.07%**	82.71%
AVNN, SDNN, Angry, Fear, Disgust, Surprise, PERCLOS	Random Forest	77.69%	77.78%	**88.77%**	74.69%	82.81%
AVNN, SDNN, Angry, Fear, Disgust, Sad, Surprise, Neutral, PERCLOS	AdaBoost	**77.77%**	77.83%	**88.77%**	74.79%	**82.86%**

The best results of this iteration, presented in Table 15 are, once again, identical. This time, the AdaBoost is the best performing algorithms, managing to obtain the highest value for accuracy, precision, and balanced accuracy in the first configuration, and the best accuracy, recall and F1 Score in the other one. All the results obtained in this iteration have reduced discrepancy among them. While all the configurations of the models managed to obtain impressive results, there's not a necessarily better model. In this scenario, the recall metric can be more important than the accuracy since it provides better feedback about the prediction power of the model in terms of classifying the positive label correctly, which is the stress. For this reason, the last AdaBoost configuration can be considered the best for this case.

By analyzing all the results, it is possible to conclude that the PERCLOS attribute is strongly correlated with the stress label, since the best results were always obtained from configurations where the RFE method chose the PERCLOS as an important feature. Moreover, the results obtained using PERCLOS were better than the all the results obtained in the previous experiments. The best result obtained in this experiment had an increase of approximately 1.86% in accuracy, 2.69% in precision, 2.81% in balanced accuracy and 0.84% in F1 Score, while only decreasing 1.65% recall, in comparison to the best performing model obtained in the previous experiments.

Since the PERCLOS value is increasing the classifiers' prediction power, this value is going to be present in the next experiments of this project.

5.4.4 Fourth Experiment

The following experiment consists in repeating the previous iteration, while adding the participants' data retrieved from the initial form. These answers are related to their age, gender, weight, company role, sleeping schedule, alcohol/smoking habits, and so on. The results obtained with this data are demonstrated in the following table.

Table 16 – Best results of the fourth experiment

Features	Algorithm	Accuracy	Precision	Recall	Balanced Accuracy	F1 Score
AVNN, Angry, PERCLOS, Weight, Type of Schedule	Random Forest	83.65%	**83.53%**	91.69%	**81.49%**	87.25%
AVNN, RMSSD, Angry, Fear, PERCLOS, Weight, Type of Schedule	Random Forest	**83.81%**	81.95%	**94.63%**	80.89%	**87.69%**
AVNN, RMSSD, Angry, Fear, PERCLOS, Weight, Type of Schedule	AdaBoost	83.03%	82.81%	91.44%	80.76%	86.77%

The results of this last experiment are undoubtedly the best of all the iterations performed. By adding the initial form data, the classifiers managed to increase all the metrics significantly. Although all configurations presented in Table 16 are good, the one that performs better for this scenario is the second configuration of features using the Random Forest classifier. This is the chosen configuration due to the slight increase of the f1 score and accuracy, as well as the fairly better recall, which is particularly important for the stress detection application.

With this experiment, it is possible to draw the conclusion that some participant characteristics, such as weight and schedule type, are strongly correlated with their levels of stress, as the RFE method consistently selected those characteristics to predict the outcomes and produced results that were superior to those of earlier iterations. Moreover, when combining these characteristics with the HRV features, facial expressions and PERCLOS, the obtained models prove to be great stress classifiers, achieving 83.81% accuracy with 94.63% recall.

5.4.5 Hyperparameter Tuning

After the best models have been obtained, there is still a margin of improvement for these results. To that end, the hyperparameters of the best model (Random Forest) will be tuned to verify that possibility.

To assess this, the GridSearchCV tool of the scikit-learn library is going to be used. This method performs an exhaustive search over specified parameter values for an estimator (scikit-learn, 2007).

The parameter values used for this project were based on a recent article which focuses on parameter optimization for the random forest classifier (Meinert, 2019). The parameter grid used is demonstrated in the following table:

Table 17 – Parameter grid for the Random Forest Classifier

n_estimators	100, 300, 500, 800, 1200
max_depth	5, 8, 15, 25, 30
min_samples_split	2, 5, 10, 15, 100
min_samples_leaf	1, 2, 5, 10

The GridSearchCV method takes, as argument, the metric which will be used to rank the hyperparameters. So, this method was executed once using each metric as the scoring value for the GridSearchCV.

Table 18 – Results of the GridSearchCV

n_estimators	max_depth	min_samples _split	min_samples _leaf	Accuracy	Precision	Recall	Balanced Accuracy	F1 Score
800	25	2	1	**84.04%**	82.58%	94.12%	81.33%	87.82%
1200	30	2	1	**84.04%**	**82.81%**	93.61%	**81.46%**	87.75%
1200	5	2	1	79.94%	76.19%	**97.96%**	75.08%	85.63%
800	25	5	1	**84.04%**	82.10%	94.89%	81.12%	**87.88%**

By analyzing Table 18, it is possible to observe that the results are slightly better than the best model obtained previously. Given the stress detection application where the desired model will be deployed, the best hyperparameters obtained are n_estimators=800, max_depth=25, min_samples_split=5 and min_samples_leaf=1. This combination of parameters was chosen due to its combination of high recall (94.89%), F1 Score (87.88%) and accuracy (84.04%). In comparison, to the best results of the fourth experiment, the accuracy increased by 0.23%, the precision increased by 0.15%, this recall value had a 0.26% increase, the balanced accuracy had an improvement of 0.23% and the F1 Score had an improvement of 0.19%.

Given the high prediction capabilities demonstrated by the developed model, the stress prediction classifier is going to be deployed in a REST API, in order to be available to provide its stress predictions to other Mad@Work-related projects.

Lastly, the tuned random forest classifier will also be implemented into a stress detection application.

5.5 Stress Detection Application

After the best machine learning model has been achieved, it will be inserted into a web application, in order to have a practical way of testing the prediction results of real cases. To this end, a web data application was developed using the Streamlit library. It consists in a Python library that simplifies the process of creating data/machine learning web application (Streamlit, 2022).

This application works by reading a file that contains the necessary data for the stress prediction. After the file is loaded, the application reads the data and sends it to the HRV feature extraction module. This module calculates the necessary features from the HRV data presented in the file and returns the arranged data in an acceptable format (e.g., list, array) to the machine learning model, to predict the results. The HRV feature extraction is done using the pyHRV library (Gomes, Margaritoff and Silva, 2019).

To properly implement the machine learning model in the web app, there's a need to serialize the model. To that end, the pickle module, present in Python, was used to convert the model into a savable file that can be loaded from anywhere, at any given time.

The following code demonstrates the process of model serialization using pickle.

```
filename = 'finalized_model.sav'
pickle.dump(model, open(filename, 'wb'))
```

Code 9 – Serialization of a machine learning model

After the model is successfully serialized, it can be loaded from other programs. In this case, the developed web app accesses the saved model to make predictions, based on the data inserted. The following snippet contains the code for the loading of a serialized model and prediction making.

```
filename = 'finalized_model.sav'
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.predict([[AVNN, RMSSD, angry, fear, perclos, weight,
schedule]])
```

Code 10 – Loading of a serialized machine learning model

The code relative to the stress application is divided in two more parts. The first part contains all the page elements displayed when the app starts. The snippet for this is present in Code 11.

```
st.title('Stress Prediction Application')
st.header('HRV Features')
file = st.file_uploader("Upload HRV file", type="csv",
accept_multiple_files=False, key=None, help=None, on_change=None,
args=None, kwargs=None, disabled=False)
button = st.button(label="Predict")
if button:
predict()
```

Code 11 – Code extract relative to the stress application interface

The other part consists in the machine learning component, used to predict the result. This subsequently englobes the HRV feature extraction procedures. First, the file needs to be loaded with a proper handling of the errors. This process is demonstrated in the following code snippet.

```
def loadFile():
        if file is not None:
              dataset = pd.read_csv(file,sep=";")
```

Code 12 - File loading method

After the file is loaded, the data is pre-processed and the mean of the facial expressions and PERCLOS are retrieved, as explained in chapter 5.2. Then, once the required HRV metrics are calculated, the predict method will classify the stress based on the developed model and the obtained features. The result is then shown on the web application through the st.header function. The procedure was implemented as demonstrated in the next code.

```
def predict():
        AVNN,RMSSD,angry,fear,perclos,weight,schedule=loadFile()
        result =
        loaded_model.predict([[AVNN,RMSSD,angry,fear,perclos,weight,schedul
        e]])
        if str(result[0])=="1":
              prediction = "stress"
        else:
              prediction="no stress"

        st.header("Result:"+" "+prediction)
```

Code 13 - Prediction method implementation

After the code is properly implemented, the web application is fully functional and ready to predict real life cases of stress. To interact with this application, the user will be presented with screens that contain several elements.

The first screen of the application consists in a simple page with a drag and drop box and two questions for the user. This program will read the inserted csv file and the users' answers to predict their stress. The file loading process, as well as the filling of the form, are exemplified in Figure 25, Figure 26 and Figure 27.

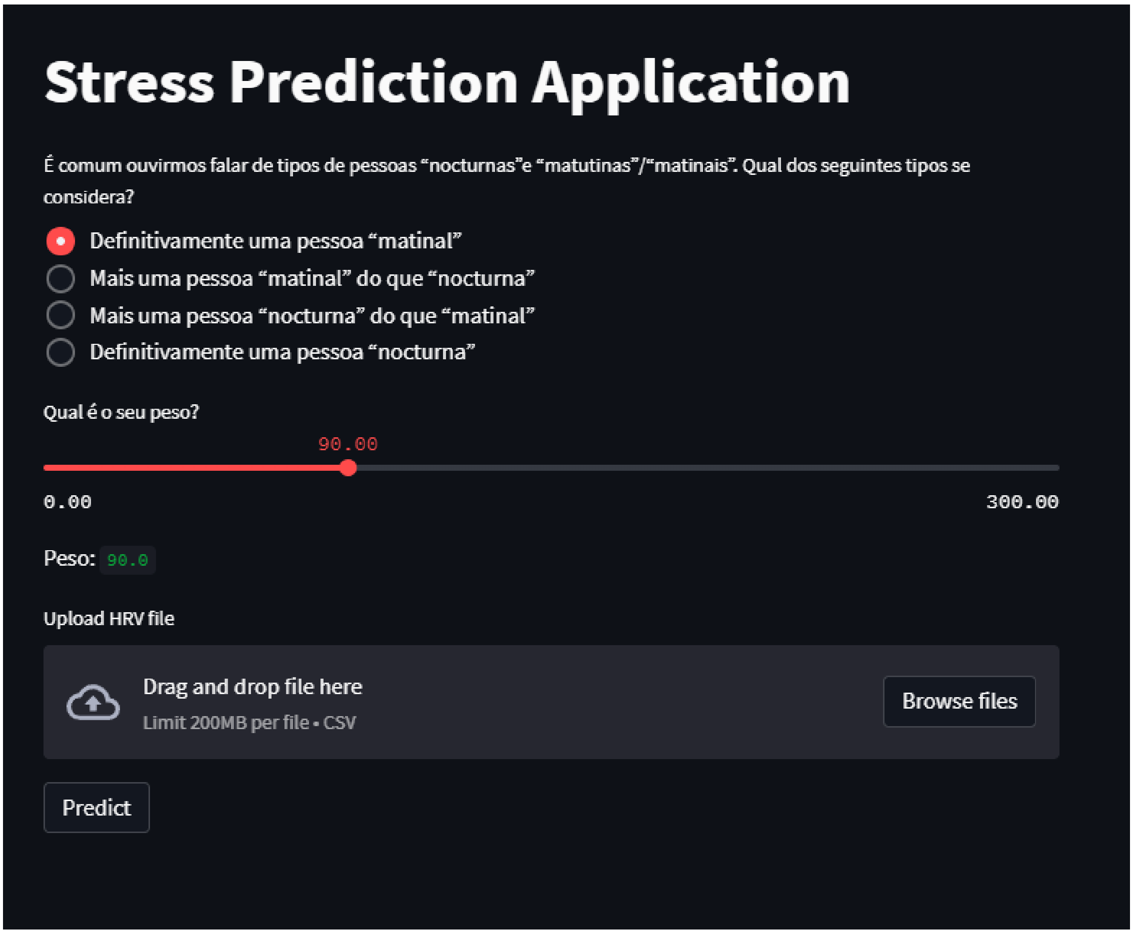
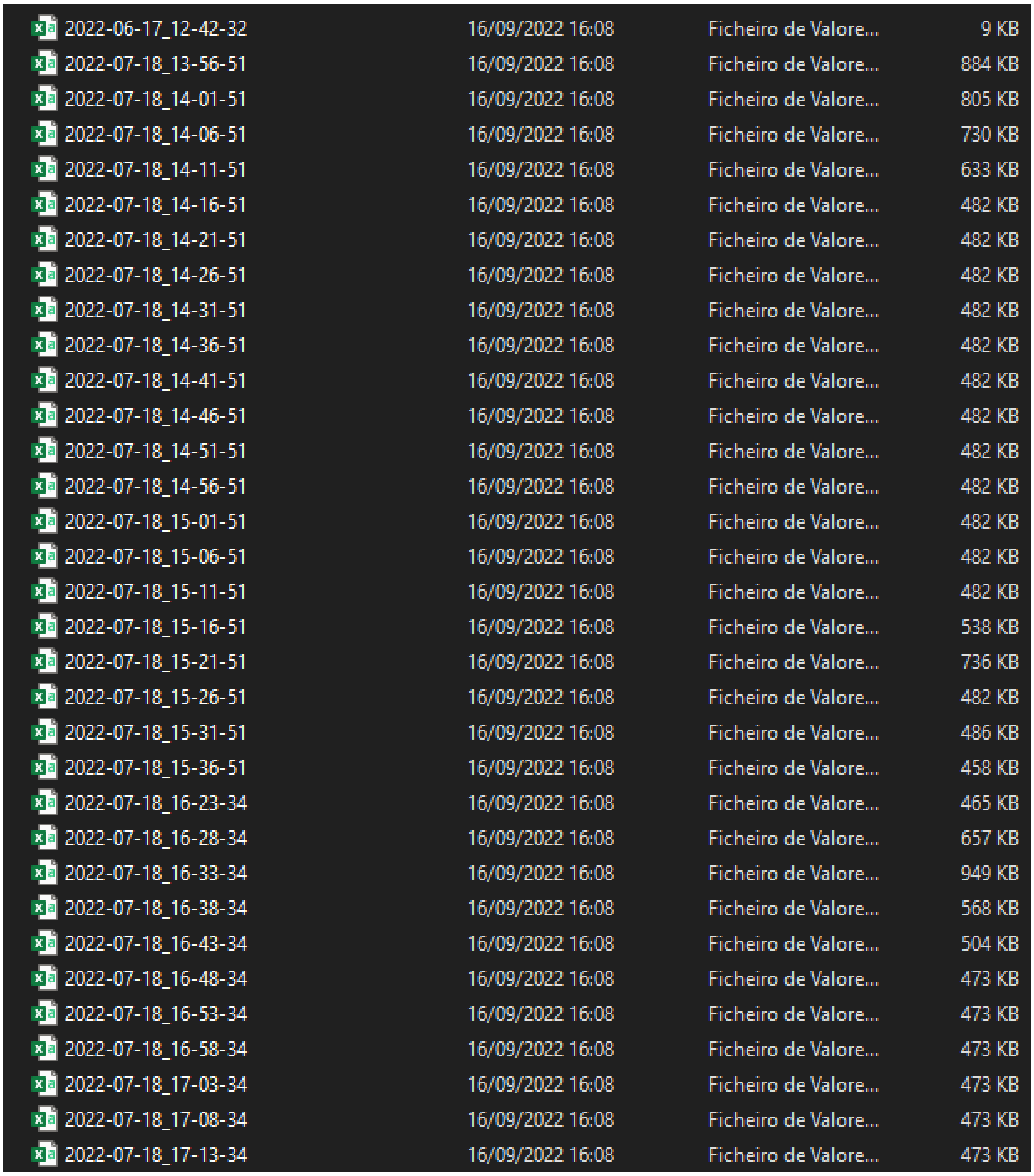


Figure 25 – Stress Prediction Application



2022-06-17_12-42-32	16/09/2022 16:08	Ficheiro de Valore...	9 KB
2022-07-18_13-56-51	16/09/2022 16:08	Ficheiro de Valore...	884 KB
2022-07-18_14-01-51	16/09/2022 16:08	Ficheiro de Valore...	805 KB
2022-07-18_14-06-51	16/09/2022 16:08	Ficheiro de Valore...	730 KB
2022-07-18_14-11-51	16/09/2022 16:08	Ficheiro de Valore...	633 KB
2022-07-18_14-16-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-21-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-26-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-31-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-36-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-41-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-46-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-51-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_14-56-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_15-01-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_15-06-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_15-11-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_15-16-51	16/09/2022 16:08	Ficheiro de Valore...	538 KB
2022-07-18_15-21-51	16/09/2022 16:08	Ficheiro de Valore...	736 KB
2022-07-18_15-26-51	16/09/2022 16:08	Ficheiro de Valore...	482 KB
2022-07-18_15-31-51	16/09/2022 16:08	Ficheiro de Valore...	486 KB
2022-07-18_15-36-51	16/09/2022 16:08	Ficheiro de Valore...	458 KB
2022-07-18_16-23-34	16/09/2022 16:08	Ficheiro de Valore...	465 KB
2022-07-18_16-28-34	16/09/2022 16:08	Ficheiro de Valore...	657 KB
2022-07-18_16-33-34	16/09/2022 16:08	Ficheiro de Valore...	949 KB
2022-07-18_16-38-34	16/09/2022 16:08	Ficheiro de Valore...	568 KB
2022-07-18_16-43-34	16/09/2022 16:08	Ficheiro de Valore...	504 KB
2022-07-18_16-48-34	16/09/2022 16:08	Ficheiro de Valore...	473 KB
2022-07-18_16-53-34	16/09/2022 16:08	Ficheiro de Valore...	473 KB
2022-07-18_16-58-34	16/09/2022 16:08	Ficheiro de Valore...	473 KB
2022-07-18_17-03-34	16/09/2022 16:08	Ficheiro de Valore...	473 KB
2022-07-18_17-08-34	16/09/2022 16:08	Ficheiro de Valore...	473 KB
2022-07-18_17-13-34	16/09/2022 16:08	Ficheiro de Valore...	473 KB

Figure 26 – Example files with physiological data

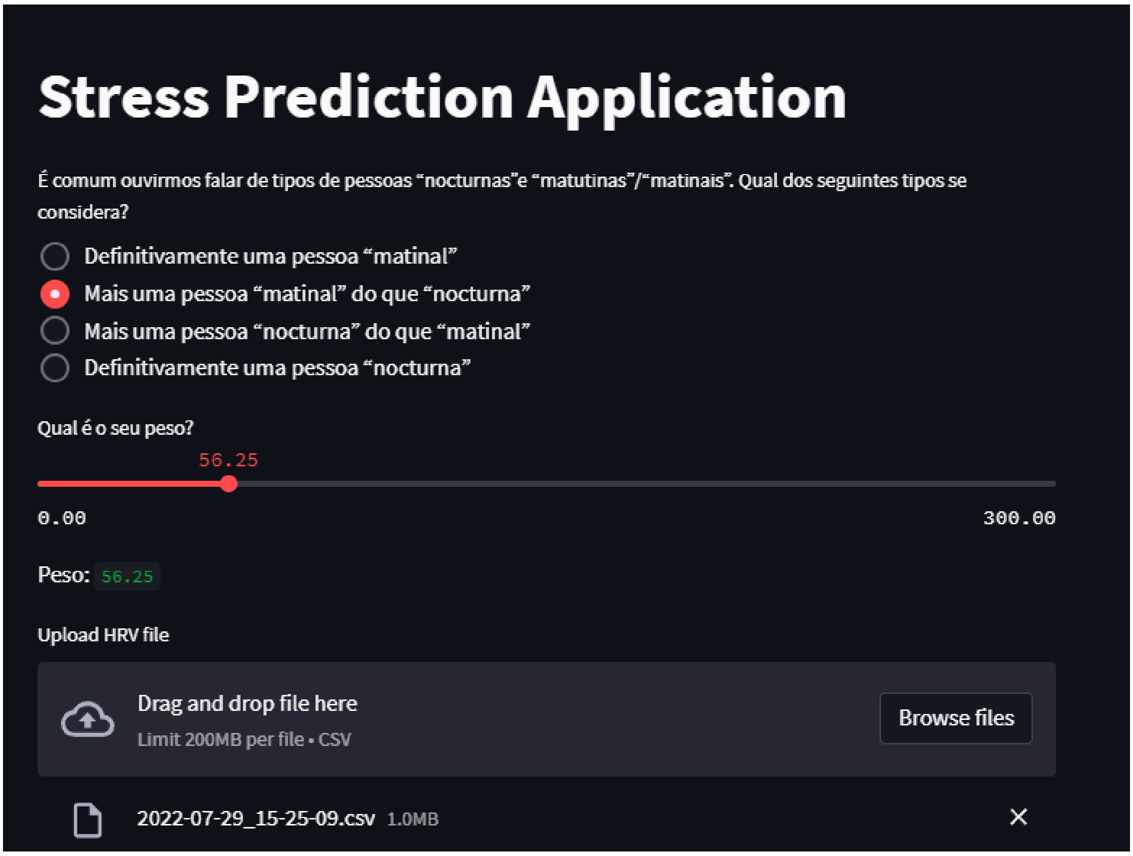


Figure 27 – Filled app form with uploaded file

After the file is loaded, the user can press the "predict" button to classify stress, based on the input data. In this case, the data used consists in a combination of some facial expressions (angry and fear), PERCLOS, HRV data (AVNN and RMSSD), the weight and type of schedule of the user. A stress prediction example is demonstrated in Figure 28.

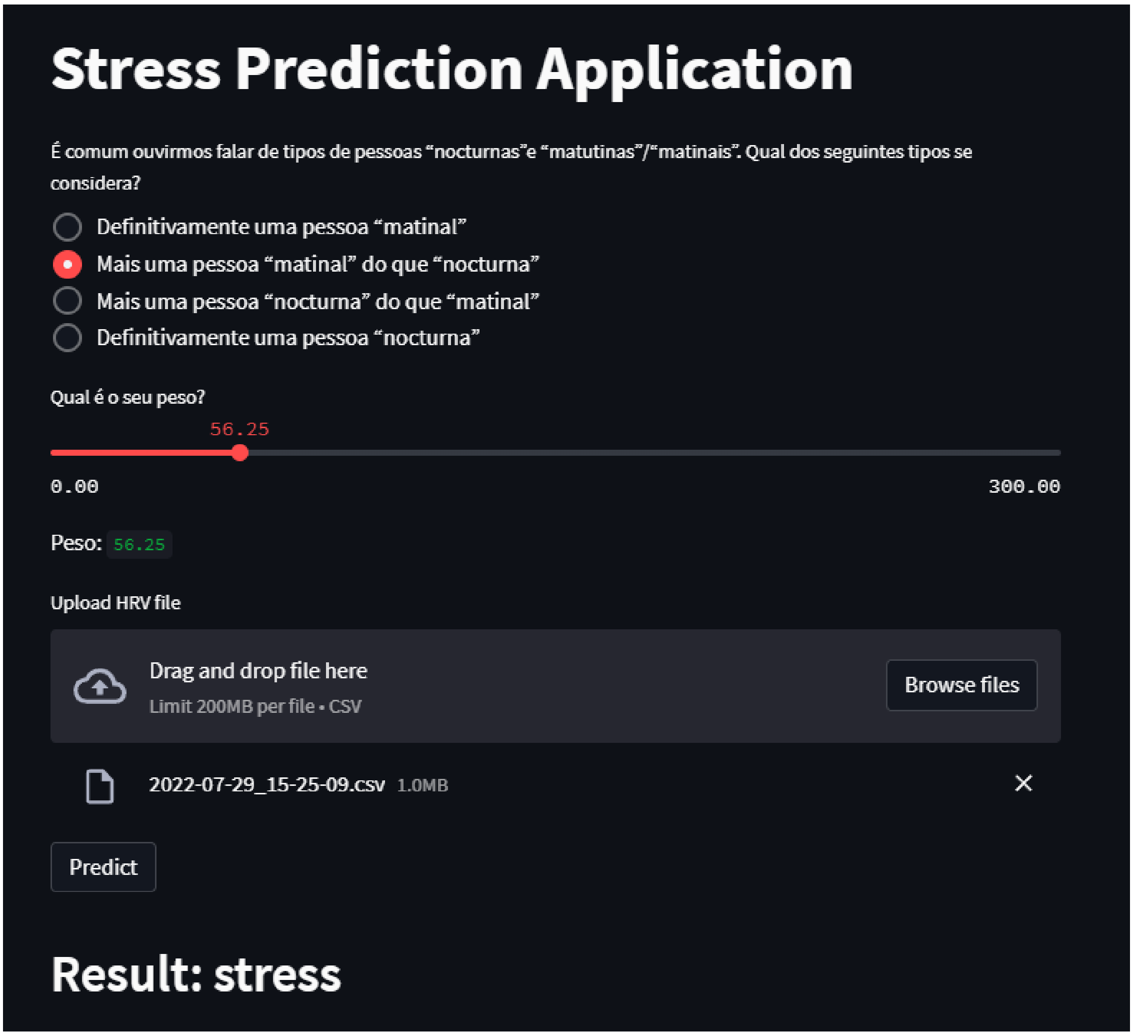


Figure 28 – Output example of the web app

After the stress detection application was fully implemented, it was used to test real life cases of stress in workers. For this test, 25 recently obtained readings were used to evaluate the performance of this system. The results obtained in this procedure are demonstrated through the confusion matrix below:

Table 19 – Confusion matrix for the stress detection application test

	Stress (Predicted)	Not stress (Predicted)
Stress (Ground-truth)	15	0
Not stress (Ground-truth)	6	4

The metrics calculated through the previous results consist in an accuracy of 76%, precision of 71.4%, recall of 100%, balanced accuracy of 70% and 83.3% F1 Score.

Given this information, it is noticeable that the model didn’t classify any ground-truth stress cases as “no stress”, which is indicated by the recall percentage. However, the model showed some difficulties in differentiating ground-truth non-stress from stress cases. Out of the 10 total non-stress cases, 6 were misclassified as stress. Although not as impactful a misclassifications of actual stress cases, it is still impacting the overall performance of the model.

Even though the results are fairly good, there is still room for improvement, which is due to the reduced quantity of good-quality data available.

6 Conclusion

The purpose of this dissertation is to develop a multimodal stress detection system using machine learning models. These models were trained using different data such as HRV, facial expressions, PERCLOS and demographic data.

For this project, the dataset used was entirely built from scratch, using the data obtained by an external video-based application, as well as a questionnaire desktop application which was specifically developed for this purpose. The created dataset is particularly interesting due to the fact that the physiological data was collected in a non-invasive way, by using a camera. Moreover, the data was retrieved from participants in an uncontrolled environment, which was at their workplaces. Since the conditions of the participants are not being supervised, there's the need of a more complex labelling process for the data, which is obtained through the self-perceived stress of the participants during their worktime, through the questionnaire application. The objective of this generated dataset is to be used in the development of machine learning models in order to create a powerful stress classifier.

To evaluate the solution, 4 experiments were performed, which consisted in alterations on the dataset, as well as an increment of features for the models. The results of these experiments mainly show that combining some HRV features, facial expressions, and questionnaire data alongside with the PERCLOS provides the best performing stress classifier. With a performed hyperparameter tuning on the best model obtained (forth experiment), it was possible to obtain a model that achieves a result of 84.04% accuracy, with 94.86% recall and 87.88% F1 Score. This model was a Random Forest Classifier which uses as input the AVNN, RMSSD, angry, fear, PERCLOS, weight and type of schedule.

With these results, it is concluded that it is possible to build a stress detection system through multimodal data, mainly retrieved from the users' web camera. And so, as the final part of the project, the obtained model is inserted into a web application which works alongside the PPG video application to form a system capable of detecting stress in users through multimodal data.

Bibliography

Abe, K. et al. (2013) ‘Automatic Classification of Eye Blink Types Using a Frame-Splitting Method’, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8019 LNAI(PART 1), pp. 117–124. Available at: https://doi.org/10.1007/978-3-642-39360-0_13.

American Psychological Association (2020) facial expression – APA Dictionary of Psychology. Available at: https://dictionary.apa.org/facial-expression (Accessed: 13 January 2022).

Bilchick, K.C. and Berger, R.D. (2006) ‘Heart rate variability’, Journal of cardiovascular electrophysiology, 17(6), p. 691.

Blount, W.P. (1927) ‘STUDIES OF THE MOVEMENTS OF THE EYELIDS OF ANIMALS: BLINKING’, Quarterly Journal of Experimental Physiology, 18(2), pp. 111–125. Available at: https://doi.org/10.1113/expphysiol.1927.sp000426.

Borza, J.S. (2011) ‘FAST Diagrams : The Foundation for Creating Effective Function Models’, in. trizcon 2011.

Brownlee, J. (2020) Repeated k-Fold Cross-Validation for Model Evaluation in Python. Available at: https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/ (Accessed: 16 February 2022).

Bülbül, H.I. and Ünsal, Ö. (2011) ‘Comparison of classification techniques used in machine learning as applied on vocational guidance data’, in Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011, pp. 298–301. Available at: https://doi.org/10.1109/ICMLA.2011.49.

Can, Y.S. et al. (2019) ‘Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study’, Sensors, 19(8). Available at: https://doi.org/10.3390/s19081849.

Champseix, R., Ribiere, L. and le Couedic, C. (2021) ‘A Python Package for Heart Rate Variability Analysis and Signal Preprocessing’, Journal of Open Research Software, 9. Available at: https://doi.org/10.5334/jors.305.

Chen, L., Zhang, X. and Wang, H. (2015) ‘An Obstructive Sleep Apnea Detection Approach Using Kernel Density Classification Based on Single-Lead Electrocardiogram’, Journal of medical systems, 39, p. 222. Available at: https://doi.org/10.1007/s10916-015-0222-6.

Chen, W. et al. (2019) ‘A Model for Basic Emotions Using Observations of Behavior in Drosophila’. Available at: https://doi.org/10.3389/fpsyg.2019.00781.

Cohen, S., Kamarck, T. and Mermelstein, R. (1994) ‘Perceived stress scale’, Measuring stress: A guide for health and social scientists, 10(2), pp. 1–2.

Crosswell, A.D. and Lockwood, K.G. (2020) ‘Best practices for stress measurement: How to measure psychological stress in health research’, Health Psychology Open, 7(2), p. 205510292093307. Available at: https://doi.org/10.1177/2055102920933072.

Dalmeida, K.M. and Masala, G.L. (2021) ‘Hrv features as viable physiological markers for stress detection using wearable devices’, Sensors, 21(8). Available at: https://doi.org/10.3390/s21082873.

David F. Dinges and Grace, R. (1998) PERCLOS, a valid psychophysiological measure of alertness as assessed by psychomotor vigilance. Edited by U.States.F.H.Administration.O. of M.C.R. and Standards. Washington, DC (400 Seventh St., SW, Room 3107, Washington 20590): Federal Highway Administration, Office of Motor Carriers, Office of Motor Carrier Research and Standards (Techbrief.). Available at: https://doi.org/10.21949/1502740.

Dewulf, K. (2013) ‘Sustainable Product Innovation: The Importance of the Front- End Stage in the Innovation Process’, in Advances in Industrial Design Engineering. InTech. Available at: https://doi.org/10.5772/52461.

Ekman, P. (1992) ‘An argument for basic emotions’, Cognition and Emotion, 6(3–4), pp. 169–200. Available at: https://doi.org/10.1080/02699939208411068.

Electrophysiology, T.F. of the E.S. of C. the N.A.S. of P. (1996) ‘Heart Rate Variability’, Circulation, 93(5), pp. 1043–1065. Available at: https://doi.org/10.1161/01.CIR.93.5.1043.

European Agency for Safety and Health at Work. (2010) OSH in figures : work-related musculoskeletal disorders in the EU : facts and figures. EUR-OP.

Eurostat (2021) Self-reported work-related health problems and risk factors - key statistics - Statistics Explained. Available at: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Self-reported_work-related_health_problems_and_risk_factors_-_key_statistics (Accessed: 4 February 2022).

Ganster, D.C. and Rosen, C.C. (2013) ‘Work Stress and Employee Health: A Multidisciplinary Review’, Journal of Management, 39(5), pp. 1085–1122. Available at: https://doi.org/10.1177/0149206313475815.

Gao, H., Yüce, A. and Thiran, J.-P. (2014) ‘Detecting emotional stress from facial expressions for driving safety’, in 2014 IEEE International Conference on Image Processing (ICIP), pp. 5961–5965. Available at: https://doi.org/10.1109/ICIP.2014.7026203.

van Gent, P. et al. (2019a) ‘Analysing Noisy Driver Physiology Real-Time Using Off-the-Shelf Sensors: Heart Rate Analysis Software from the Taking the Fast Lane Project’, Journal of Open Research Software, 7(1), p. 32. Available at: https://doi.org/10.5334/jors.241.

van Gent, P. et al. (2019b) ‘HeartPy: A novel heart rate algorithm for the analysis of noisy signals’, Transportation Research Part F: Traffic Psychology and Behaviour, 66, pp. 368–378. Available at: https://doi.org/10.1016/j.trf.2019.09.015.

Gigalixir (2022) Elixir’s Platform as a Service. Available at: https://www.gigalixir.com/ (Accessed: 20 May 2022).

Gomes, P., Margaritoff, P. and Silva, H. (2019) ‘pyHRV: Development and evaluation of an open-source python toolbox for heart rate variability (HRV)’, in Proc. Int’l Conf. on Electrical, Electronic and Computing Engineering (IcETRAN), pp. 822–828.

Haak, M. et al. (2009) ‘Detecting stress using eye blinks and brain activity from EEG signals’, Proceeding of the 1st driver car interaction and interface (DCII 2008), pp. 35–60.

van den Haak, P. et al. (2010) ‘Stress Assessment of Car-Drivers Using EEG-Analysis’, in Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD

Students in Computing on International Conference on Computer Systems and Technologies. New York, NY, USA: Association for Computing Machinery (CompSysTech ’10), pp. 473–477. Available at: https://doi.org/10.1145/1839379.1839463.

Harley, J.M. (2016) ‘Chapter 5 - Measuring Emotions: A Survey of Cutting Edge Methodologies Used in Computer-Based Learning Environment Research’, in S.Y. Tettegah and M. Gartmeier (eds) Emotions, Technology, Design, and Learning. San Diego: Academic Press, pp. 89–114. Available at: https://doi.org/10.1016/B978-0-12-801856-9.00005-0.

Harvard Health Publishing Staff (2021) Heart rate variability: How it might indicate well-being - Harvard Health. Available at: https://www.health.harvard.edu/blog/heart-rate-variability-new-way-track-well-2017112212789 (Accessed: 12 January 2022).

Henderson, R.R., Bradley, M.M. and Lang, P.J. (2018) ‘Emotional imagery and pupil diameter’, Psychophysiology, 55(6), p. e13050. Available at: https://doi.org/10.1111/psyp.13050.

Huang, C.-L., Chen, M.-C. and Wang, C.-J. (2007) ‘Credit Scoring with a Data Mining Approach Based on Support Vector Machines’, Expert Syst. Appl., 33(4), pp. 847–856. Available at: https://doi.org/10.1016/j.eswa.2006.07.007.

Hughes, G.D. and Chafin, D.C. (1996) ‘Turning new product development into a continuous learning process’, Journal of Product Innovation Management, 13(2), pp. 89–104. Available at: https://doi.org/10.1016/0737-6782(95)00112-3.

IBM Cloud Education (2020) What is Supervised Learning? | IBM. Available at: https://www.ibm.com/cloud/learn/supervised-learning (Accessed: 12 January 2022).

IBM Cloud Education (2021) What is a REST API? | IBM. Available at: https://www.ibm.com/cloud/learn/rest-apis (Accessed: 18 February 2022).

Irwin, D.E. and Thomas, L.E. (2010) ‘Eyeblinks and cognition.’, in Tutorials in visual cognition. New York, NY, US: Psychology Press (Macquarie monographs in cognitive science.), pp. 121–141.

JONES, J.R. (2005) ‘Artificial organs’, in L.L. Hench and J.R. Jones (eds) Biomaterials, Artificial Organs and Tissue Engineering. Elsevier (Woodhead Publishing Series in Biomaterials), pp. 142–152. Available at: https://doi.org/10.1533/9781845690861.3.142.

Junaedi, S. and Akbar, H. (2018) ‘Driver Drowsiness Detection Based on Face Feature and PERCLOS’, Journal of Physics: Conference Series, 1090(1), p. 012037. Available at: https://doi.org/10.1088/1742-6596/1090/1/012037.

Keltner, D. et al. (2019) ‘What Basic Emotion Theory Really Says for the Twenty-First Century Study of Emotion’, Journal of Nonverbal Behavior, 43, pp. 195–201. Available at: https://doi.org/10.1007/s10919-019-00298-y.

Kim, H.-G. et al. (2018) ‘Stress and Heart Rate Variability: A Meta-Analysis and Review of the Literature’, Psychiatry investigation. 2018/02/28, 15(3), pp. 235–245. Available at: https://doi.org/10.30773/pi.2017.08.17.

Koen, P. et al. (2001) ‘Providing Clarity and A Common Language to the "Fuzzy Front End’, Research-Technology Management, 44(2), pp. 46–55. Available at: https://doi.org/10.1080/08956308.2001.11671418.

Kowalska, M. and Wróbel, M. (2017) ‘Basic Emotions’, in Encyclopedia of Personality and Individual Differences. Cham: Springer International Publishing, pp. 1–6. Available at: https://doi.org/10.1007/978-3-319-28099-8_495-1.

Lerner, J.S. et al. (2007) ‘Facial Expressions of Emotion Reveal Neuroendocrine and Cardiovascular Stress Responses’, Biological Psychiatry, 61(2), pp. 253–260. Available at: https://doi.org/10.1016/j.biopsych.2006.08.016.

Li, G. and Chung, W.-Y. (2014) ‘Estimation of Eye Closure Degree Using EEG Sensors and Its Application in Driver Drowsiness Detection’, Sensors, 14, pp. 17491–17515. Available at: https://doi.org/10.3390/s140917491.

Mad@Work (2021) Mad@Work – Project for researching and developing solutions for stress in knowledge work. Available at: https://www.madatwork.eu/ (Accessed: 6 January 2022).

Mansfield, T. (2019) Value Proposition Canvas explained: how to match your services to customer needs. Available at: https://interaction.net.au/articles/value-proposition-canvas-explained/ (Accessed: 28 January 2022).

McDuff, D., Gontarek, S. and Picard, R. (2014) ‘Remote measurement of cognitive stress via heart rate variability’, in 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2957–2960. Available at: https://doi.org/10.1109/EMBC.2014.6944243.

Meinert, R. (2019) Optimizing Hyperparameters in Random Forest Classification | by Reilly Meinert | Towards Data Science, Optimizing Hyperparameters in Random Forest Classification. Available at: https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6 (Accessed: 26 September 2022).

Mental Health Foundation (2021) Stress. Available at: https://www.mentalhealth.org.uk/a-to-z/s/stress (Accessed: 8 February 2022).

Moody, G. (2002) RR Interval Time Series Modeling - The PhysioNet Computing in Cardiology Challenge 2002 v1.0.0. Available at: https://physionet.org/content/challenge-2002/1.0.0/ (Accessed: 12 January 2022).

Moody, G. (2016) RR Intervals, Heart Rate, and HRV Howto. Available at: https://archive.physionet.org/tutorials/hrv/ (Accessed: 10 February 2022).

el Naqa, I. and Murphy, M.J. (2015) ‘What Is Machine Learning?’, in I. el Naqa, R. Li, and M.J. Murphy (eds) Machine Learning in Radiation Oncology: Theory and Applications. Cham: Springer International Publishing, pp. 3–11. Available at: https://doi.org/10.1007/978-3-319-18305-3_1.

Nkurikiyeyezu, K., Yokokubo, A. and Lopez, G. (2020) ‘Effect of Person-specific Biometrics in Improving Generic Stress Predictive Models’, Sensors and Materials, 32(2), pp. 703–722. Available at: https://doi.org/10.18494/SAM.2020.2650.

Novaković, J.Dj. et al. (2017) ‘Evaluation of Classification Models in Machine Learning’, Theory and Applications of Mathematics & Computer Science, 7(1), p. Pages: 39 – 46. Available at: https://uav.ro/applications/se/journal/index.php/TAMCS/article/view/158 (Accessed: 9 October 2022).

Osterwalder, A. (2004) The business model ontology a proposition in a design science approach. Available at: https://doi.org/10.22005/BCU.15985.

Osterwalder, A. and Pigneur, Y. (2010) Business model generation: a handbook for visionaries, game changers, and challengers. John Wiley & Sons.

Othman, N. and Romli, F.I. (2016) ‘Mental workload evaluation of pilots using pupil dilation’, International Review of Aerospace Engineering, 9(3), pp. 80–84. Available at: https://doi.org/10.15866/irease.v9i3.9541.

Park, J., Kim, J. and Kim, S. (2018) ‘Prediction of Daily Mental Stress Levels Using a Wearable Photoplethysmography Sensor’, in TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 1899–1902. Available at: https://doi.org/10.1109/TENCON.2018.8650109.

Patro, R. (2021) Cross-Validation: K Fold vs Monte Carlo | by Rebecca Patro | Towards Data Science. Available at: https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b (Accessed: 17 February 2022).

Pedrotti, M. et al. (2014) ‘Automatic Stress Classification With Pupil Diameter Analysis’, International Journal of Human-Computer Interaction, 30(3), pp. 220–236. Available at: https://doi.org/10.1080/10447318.2013.848320.

Peshori, K.R. et al. (2001) ‘Aging of the trigeminal blink system’, Experimental Brain Research, 136(3), pp. 351–363. Available at: https://doi.org/10.1007/s002210000585.

Petkar, H. et al. (2009) ‘A Pilot Study to Assess Designer’s Mental Stress Using Eye Gaze System and Electroencephalogram’, in Volume 2: 29th Computers and Information in Engineering Conference, Parts A and B. ASMEDC, pp. 899–909. Available at: https://doi.org/10.1115/DETC2009-86542.

Qastharin, A. (2015) ‘Business Model Canvas for Social Enterprise’, in. The 7th Indonesia International Conference on Innovation, Entrepreneurship, and Small Business (IICIES 2015).

Red Hat (2020) What is a REST API? Available at: https://www.redhat.com/en/topics/api/what-is-a-rest-api (Accessed: 18 February 2022).

Rigó, M. et al. (2021) ‘Work stress on rise? Comparative analysis of trends in work stressors using the European working conditions survey’, International Archives of Occupational and Environmental Health, 94(3), pp. 459–474. Available at: https://doi.org/10.1007/s00420-020-01593-8.

Saaty, T.L. (1988) ‘What is the Analytic Hierarchy Process?’, in Mathematical Models for Decision Support. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 109–121. Available at: https://doi.org/10.1007/978-3-642-83555-1_5.

Saaty, T.L. (2008) ‘Decision making with the analytic hierarchy process’, International Journal of Services Sciences, 1(1), pp. 83–98.

scikit-learn (2007) sklearn.model_selection.GridSearchCV — scikit-learn 1.1.2 documentation. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (Accessed: 19 September 2022).

Sen, P.C., Hajra, M. and Ghosh, M. (2020) ‘Supervised Classification Algorithms in Machine Learning: A Survey and Review’, Advances in Intelligent Systems and Computing, 937, pp. 99–111. Available at: https://doi.org/10.1007/978-981-13-7403-6_11.

Shaffer, F. and Ginsberg, J.P. (2017) ‘An Overview of Heart Rate Variability Metrics and Norms’, Frontiers in Public Health. Frontiers Media S.A. Available at: https://doi.org/10.3389/fpubh.2017.00258.

Sioni, R. and Chittaro, L. (2015) ‘Stress Detection Using Physiological Sensors’, Computer, 48(10), pp. 26–33. Available at: https://doi.org/10.1109/MC.2015.316.

Sirohi, P., Agarwal, A. and Tyagi, S. (2017) 'A comprehensive study on security attacks on SSL/TLS protocol', in Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016. Institute of Electrical and Electronics Engineers Inc., pp. 893–898. Available at: https://doi.org/10.1109/NGCT.2016.7877537.

Streamlit (2022) Streamlit documentation. Available at: https://docs.streamlit.io/ (Accessed: 26 April 2022).

Torres-Salomao, L.A., Mahfouf, M. and El-Samahy, E. (2015) 'Pupil diameter size marker for incremental mental stress detection', in 2015 17th International Conference on E-Health Networking, Application and Services, HealthCom 2015. Institute of Electrical and Electronics Engineers Inc., pp. 286–291. Available at: https://doi.org/10.1109/HealthCom.2015.7454513.

Trutschel, U. et al. (2011) 'PERCLOS: An Alertness Measure of the Past', in Proceedings of the 6th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design : driving assessment 2011. Iowa City, Iowa: University of Iowa, pp. 172–179. Available at: https://doi.org/10.17077/drivingassessment.1394.

VanBruggen, M.D. et al. (2011) 'The Relationship Between Serum and Salivary Cortisol Levels in Response to Different Intensities of Exercise', International Journal of Sports Physiology and Performance, 6(3), pp. 396–407. Available at: https://doi.org/10.1123/IJSPP.6.3.396.

Winn, B. et al. (1994) 'Factors affecting light-adapted pupil size in normal human subjects.', Investigative Ophthalmology & Visual Science, 35(3), pp. 1132–1137.

Wirth, R. and Hipp, J. (2000) 'CRISP-DM: Towards a Standard Process Model for Data Mining', in Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1). London, UK: Springer-Verlag.

World Health Organization (2020) Occupational health: Stress at the workplace. Available at: https://www.who.int/news-room/questions-and-answers/item/ccupational-health-stress-at-the-workplace (Accessed: 5 January 2022).

World Health Organization/Europe (2017) WHO/Europe | Events - Mental health at the workplace – World Mental Health Day 2017. Available at: https://www.euro.who.int/en/media-centre/events/events/2017/10/mental-health-at-the-workplace-world-mental-health-day-2017 (Accessed: 6 January 2022).