UNISINOS

Programa de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

William da Rosa Fröhlich

ATHENA I :
An Architecture for Real-Time Monitoring of Physiological Signals supported by Artificial Intelligence

São Leopoldo, 2022

William da Rosa Fröhlich

ATHENA I: An Architecture for Real-Time Monitoring of Physiological Signals supported by Artificial Intelligence

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Advisor:
Prof. PhD. Sandro José Rigo

Co-advisor:
Prof. PhD. Marta Rosecler Bez

São Leopoldo
2022

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Fröhlich, William da Rosa

ATHENA I: an architecture for real-time monitoring of physiological signals supported by artificial intelligence / William da Rosa Fröhlich — 2022.

103 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2022.

“Advisor: Prof. PhD. Sandro José Rigo, Unidade Acadêmica de Pesquisa e Pós-Graduação”.

1. Artificial Intelligence. 2. Multi-signal Acquisition. 3. Physiological Signals. 4. Stress Detection. 5. Wearables Sensors. I. Título.

CDU 004

Bibliotecária responsável: Silvana Dornelles Studzinski — CRB 10/2524

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001

ABSTRACT

Wearable sensors may obtain reliable physiological signals to diagnose diseases and detect changes. Wearables can measure signs such as electrocardiogram, heart rate, electroencephalogram, electromyogram, or galvanic skin response. All these signals have intrinsic characteristics in a normal state and change if associated with illness. The literature presents the Machine Learning Approaches and Deep Learning Models as alternatives to pattern detection in physiological signals. The state-of-the-art in this area indicates a trend to use wearables for continuous monitoring of patients, whether in a hospital or home environment, as it is a portable and non-invasive option. In addition, many studies point out the low cost of wearable sensors as another advantage compared to traditional hospital medical equipment. Other studies highlight the possibility of supporting automatic diseases detection, especially chronic diseases, using artificial intelligence in physiological signals. Based on the review carried out, it is possible to conclude that there are still new development opportunities. The studied papers do not address at the same time aspects of lower cost, greater flexibility, wide use of Machine Learning resources, and communication of results. This work's main objective is to develop an architecture for multisignal acquisition with wearable sensors for continuous monitoring and stress detection. The architecture comprises wearable sensors and single-board computers—the wearable sensor for data processing and single-board computer to communicate the results to other platforms. The differentials of this architecture consist of the integration of resources for multi-signal acquisition for continuous monitoring of patients with a low implementation cost, flexibility, and ease of use. We developed a prototype in a modular way, and we tested each module of the architecture. These tests aimed to guarantee the independence of the components, carefully evaluating the stability and plausibility of the data. We also carried out two practical stress-inducing experiments. The first composed a proprietary dataset to generate a Machine Learning model, and the second allowed full architecture assessment, focusing on real-time detection. The training and classification results of the Machine Learning model showed promising results, with accuracy above 98.72% for binary classification and 92.72% for classification with three classes. When analyzing the real-time classification, we obtained an accuracy of 69.00% for participants in the first round of experiments. The architecture presented excellent communication and operation stability. During the experiments, the architecture performed short and long acquisitions efficiently. The acquired data showed promising results, with plausible and justifiable values within the context of the experiment performed. The classification results obtained when testing the model with participants who were in training, the results were relatively high.

Keywords: Artificial Intelligence. Multi-signal Acquisition. Physiological Signals. Stress Detection. Wearables Sensors.

RESUMO

Sensores vestíveis (*wearable sensors*) vêm sendo utilizados para obtenção confiável de sinais fisiológicos, com o intuito de apoiar o diagnóstico de doenças. Os *wearable sensors* podem medir sinais diversos, tais como eletrocardiograma, variação da frequência cardíaca, eletroencefalograma, eletromiograma ou resposta galvânica da pele. Todos esses sinais possuem suas próprias características em estado normal e mudam quando associadas a alguma doença, ou sob efeito de estresse. Abordagens de Aprendizado de Máquina (*Machine Learning*) ou Modelos de Aprendizagem Profunda são opções para o uso destes sinais na detecção e predição de padrões, atuando como apoio no diagnóstico. O estado da arte nesta área indica uma tendência no uso de *wearable sensors* para monitoramento contínuo de pacientes, seja em ambiente hospitalar ou doméstico, por ser uma opção portátil e não invasiva. Estudos apontam o baixo custo dos sensores vestíveis como uma vantagem, quando comparados com equipamentos médicos hospitalares tradicionais. Outros estudos destacam a possibilidade de apoio na detecção automática de doenças, em especial doenças crônicas, utilizando inteligência artificial em sinais fisiológicos. Com base na revisão realizada, é possível concluir que há oportunidades de desenvolvimento ainda não exploradas, pois não foram encontrados trabalhos que contemplem ao mesmo tempo os aspectos de menor custo, maior flexibilidade e uso amplo de recursos de Aprendizado de Máquina para detecção automática. Neste contexto, o principal objetivo desse trabalho é o desenvolvimento de uma arquitetura para aquisição de multisinais com *wearable sensors*, monitoramento contínuo e detecção de estresse. A arquitetura é composta por sensores vestíveis, para aquisição do sinal, e computadores de placa única, para processamento dos dados e comunicação de resultados para outras plataformas. Os diferenciais desta arquitetura consistem na integração de recursos para aquisição multisinais para monitoramento contínuo de pacientes com um baixo custo de implementação, flexibilidade e facilidade de uso. Um protótipo foi desenvolvido de forma modular e cada módulo da arquitetura foi testado individualmente, para garantir a independência dos componentes, avaliando criteriosamente a estabilidade e a plausibilidade dos dados. Além dos experimentos de testes, também foram realizados dois experimentos práticos de indução de estresse. O primeiro para composição de um *dataset* proprietário da pesquisa para geração de um modelo de Aprendizado de Máquina. O segundo para avaliação completa da arquitetura, com foco na detecção em tempo real. Os resultados de treinamento e classificação do modelo de *Machine Learning* apresentaram bons resultados, com acurácia acima de 98,72% para classificação binária e 92,72% para classificação com três classes. Ao analisarmos a classificação em tempo real, ocorreu uma queda para 52,33% de acurácia com todos novos participantes e participantes que estiveram no primeiro experimento e 69,00% para apenas os participantes que estiveram na primeira rodada de experimentos. A arquitetura apresentou ótima estabilidade de comunicação e funcionamento, realizando aquisições curtas e longas de forma eficiente, sem perdas de dados nem travamentos. Os dados do *dataset* também apresentaram bons resultados, com valores plausíveis e justificáveis dentro do contexto do experimento realizado. Os resultados de predição obtidos ao testar o modelo com os participantes que estiveram no treinamento, os resultados foram bastante elevados. Porém apresentaram queda significativa ao tentar classificar participantes novos, como nos experimentos em tempo real.

Palavras-chave: Aquisição Multisinais. Detecção de Estresse. Inteligência Artificial. Sensores Vestíveis. Sinais Fisiológicos.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Bi-Dimensional
3D	Three-Dimensional
etc.	Et cetera
GB	Gigabit
GHz	Gigahertz
Hz	Hertz
kHz	Kilohertz
MB	Megabyte
min.	Minutes
sec.	Seconds
TEMP	Temperature
txt	Text File

LIST OF ACRONYMS

ABNT	Associação Brasileira de Normas Técnicas
ACC	Accelerometry
AI	Artificial Intelligence
ANN	Artificial Neural Networks
API	Application Programming Interface
ARM	Advanced RISC Machines
BMI	Body Mass Index
BP	Blood Pressure
CAAE	Certificado de Apresentação de Apreciação Ética
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CGM	Continuous Glucose Monitoring
CNN	Convolutional Neural Network
CVS	Comma-separated Values
DL	Deep Learning
DT	Decision Tree
ECOC	Error-correcting Output Code
ECG	Electrocardiogram
EDA	Electrodermal Activity
EDR	Electrodermal Response
EEG	Electroencephalogram
EMR	Electronic Medical Records
EOG	Electrooculography
FCM	Fundamental Modeling Concepts
FIR	Finite Impulse Response
FN	False Negative
FP	False Positive
FAPERGS	Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSR	Galvanic Skin Response
HDMI	High-Definition Multimedia Interface
HPA	Hypothalamic-Pituitary-Adrenal
HR	Heart Rate

HRV	Heart Rate Variability
IoT	Internet of Things
IP	Internet Protocol
kNN	k Nearest Neighbor
LAN	Local Area Network
LDA	Linear Discriminant Analysis
LED	Light Emitting Diode
LR	Linear Regression
LSTM	Long-Short-Term-Memory
LUX	Light Sensor
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Erro
MAV	Mean Absolute Value
MAC	Media Access Control
MIST	Montreal Imaging Stress Task
ML	Machine Learning
MLP	Multilayer Perceptron
MNF	Mean Frequency
MSE	Mean Squared Error
NFC	Near Field Communication
NLP	Natural Language Processing
PNS	Parasympathetic Nervous System
PPG	Photoplethysmogram
RAM	Random Access Memory
ReLu	Rectified Linear Unit
RF	Random Forest
RFID	Radio-Frequency IDentification
RMSE	Root Mean Squared Error
RMS	Root Mean Square
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic Curve
RSP	Respiration
SAM	Sympathetic-Adrenal-Medullary
SBC	Single-board Computer

SC	Skin Conductance
SCR	Skin Conductance Response
sEMG	Surface Electromyographic
SNS	Sympathetic Nervous System
SoC	System On Chip
SPAM	Sending and Posting Advertisement in Mass
SPI	Serial Peripheral Interface
SVM	Support Vector Machine
TAM	Technical ArchitectureModeling
TCP	Transmission Control Protocol
TD	Time Domain
TEMP	Temperature
TFD	Time-Frequency Domain
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
UML	Unified Modeling Languag
USB	Universal Serial Bus

CONTENTS

1 INTRODUCTION

Stress has had a direct impact on people's lives in recent years, as reported by Farias et al. (2011) to Can, Arnrich and Ersoya (2019). Studies about the stress occurrence in health professionals demonstrated that many people suffer its effects. The most common physical symptoms of stress are headache, feeling tired, leg pain, and tachycardia. In addition, many professionals reported difficulty differentiating emotional and physical stress. Not dealing with stress may turn it into a problem if the person is periodically under stress for long periods. The authors explain that stress may cause issues development in many parts of our body, such as in the stomach, heart, lymphatic system, immune system, decreasing the levels of endorphin and serotonin, consequently causing discouragement.

The aspects listed are just a tiny part of the problems arising from the lack of health care. Gibbs et al. (2015) argue that sedentary behavior increases the risk of the occurrence of health diseases, and the level of physical inactivity may be related to the early development of other conditions, such as diabetes or high blood pressure. The continuous monitoring of physiological signs can help prevent health issues by using pattern detection resources to identify possible stress contexts. Wearable sensors are a great low-cost alternative for patient monitoring in the hospital context, in contrast to commercial medical equipment that has a high cost, which is often a problem for the public sector (NISWAR et al., 2019). Elmalaki et al. (2021) explains that the wearable sensors, due to their cost, are good alternatives for wide use in the context of public health, thus allowing a large-scale use, serving more people with less financial resources.

Furthermore, wearable sensors are an excellent option for performing medical monitoring simplified outside the hospital environment. Wearable devices are good non-intrusive options for monitoring patients. For this reason, their use has increased in the health care field in recent years (KIM; KO, 2021). Besides, Dunn, Runge and Snyder (2018) argues that the wearable sensors provide opportunities to improve the quality of the treatments due to the easy patient follow-up, regardless of the patients' location, such as in the hospital, clinic, ambulatory, or home. On the other hand, obtaining valuable data is a complicated issue due to the large volume of information. Despite these advantages, it is important to highlight what Kim and Ko (2021) explains, that there are still many legal issues related to the use of wearable sensors, mainly regarding telemedicine.

Kim and Ko (2021) highlight that several works presented good results with these devices applied in ambulatory monitoring. The authors explain that there are already medical devices with non-intrusive characteristics. For example, the Holter monitors continuous ECG recordings with excellent signals. However, these devices cannot get a wide variety of signs, such as EMG, EDA, or EEG. Gao, Brooks and Klonoff (2018) argue that the wearable sensors allow continuous monitoring of various signals, be in the hospital, work, or house. Besides, it is an alternative non-intrusive, no restricting the movement, and non-intrusive, without insertion. The precision is currently improved, and the communication is more stable to transfer the data to the device that stores the signs. After the acquisition, the signal quality can be even improved by filtering the data and reducing some eventual noise or correcting distortions (PONCIANO et al., 2020).

An architecture that performs physiological signals acquisition can be part of a larger ecosystem by including many other issues about architectures and applications. For example, the architecture can mix computing, edge computing, or even if the application would run in the cloud or locally (HEIDEL; HAGIST, 2020). Concerning the application, there are many alternatives from signaling and visualization. Another option is to configure the architecture to send a message if anything is detected. Collaborating with the aspects previously indicated, Dunn, Runge and Snyder (2018) evaluated twenty-seven different wearable sensors. The authors also describe the importance of the wearable sensors applied to health, highlighting how a medical evolution enabled monitoring patients outside the hospital environment. Dunn, Runge and Snyder (2018) point out the lack of standardization since each device is limited to its platform, hindering the possibility of application and development.

Heidel and Hagist (2020) points out the benefits and risks of using wearable sensors in a healthcare system. According to the authors, Germany is the first world country to use wearable sensors on a large scale. The main benefits are easy patient follow-up; the patient can carry out his acquisition with minimal guidance; simplified sharing of the data obtained; no location limits. On the other hand, there is a concern regarding the privacy of the acquired data. Another issue is the possibility of losing relevant data for the data transmission. As highlighted, we see a research gap to be fulfilled in this context since medical equipment has a very high cost for the health system. We may combine this information with the fact that many devices do not allow continuous patient monitoring out of hospital beds. Finally, we believe it is essential that the system performs real-time anomaly detection based on defined data.

1.1 Research question and objectives

Based on the context previously mentioned, this research aims to develop an integrated architecture for multi-signals acquisition to detect patterns based on artificial intelligence. The research question for this work is:

- How to detect stress through a low-cost architecture for the physiological multi-signals acquisition based on wearable sensors?

To answer the research question, the main objective of this work was the development of an architecture to integrate wearable sensors and single-board computers to identify in real-time the patterns of interest.

Thus, to respond to the general objective, we formulate the following specific objectives:

a) Define requirements for an architecture composed of single-board computer and wearable sensors, acquiring, processing, and filtering physiological signals;

b) Explore artificial intelligence options to detect patterns based on physiological signals;

c) Analyze the architecture through practical application of data acquisition;

d) Train artificial intelligence models and tests for real-time stress detection.

1.2 Methodology

This work starts with a systematic literature review to identify the state-of-the-art connected wearable sensors and artificial intelligence applied to health. Some issues were listed, such as: What can be the differentials developed? What are the possible challenges encountered throughout the project? What concepts are necessary for the development of the proposed architecture? This revision determined the scope and established the goals to develop the architecture.

We adopted an exploratory methodology in this work that aims to describe details of an applied scenario for the architecture use and demonstrate possibilities for the development of a solution through experimentation, as described Koche (2011). Regarding the methodological process, this work has as characteristic an experimental approach (GERHARDT; SILVEIRA, 2009), aiming to create prototype hardware and software to perform the detection of physiological signals through artificial intelligence.

Due to the focus on data, this work follows a data-driven approach, intending to use artificial intelligence (BLIKSTEIN, 2011). The prototype has four blocks, focusing on allowing flexibility for the architecture. The developed architecture was named ATHENA I, an acronym for Architecture for Healthcare Reinforced by Artificial Intelligence. Aiming to collaborate with the area, the division also seeks to permit other researchers to explore the advances of this work.

This research counted on the support of the post-graduate program in psychology at the University of Vale do Rio do Sinos (UNISINOS) to validate the results and decisions related to the health field. A psychology research group with volunteers conducted all experiments that we have performed to validate the ATHENA I. This process has received the approval of the scientific ethics committee to carry out. Due to the context of the COVID-19 pandemic, the experiments followed all health protocols for the safety of the participants.

The research methodology for the development of this architecture follows the following steps (WAZLAWICK, 2014): I) bibliographic review of physiological signals, the internet of things context, as well as concepts of artificial intelligence; II) systematic review of the state-of-the-art; III) development of the architecture prototype; IV) acquisition of the dataset, through experiments; V) development of the artificial intelligence patterns detection model; VI) experiments to evaluate the results; VII) description and documentation.

1.3 Scientific Contributions

This Master Thesis investigates an architecture to detect stress patterns using physiological signs. We expect to improve the current stress detection possibilities with this architecture, providing a low cost and flexible support. In this way, we detailed the main contributions of this work below.

- Mapping of the state of the art approaches for physiological pattern detection, with a literature review, conducted considering both computing and healthcare areas;
- New architecture to detect stress patterns using wearable sensors and single-board computers. Based on the research opportunities identified in the literature, we proposed new architecture. The new architecture divided into blocks operates from signal acquisition, processing, data storage, model training, and real-time classification;
- Acquisition of a dataset based on the induction of stress in clinical participants (they have psychological monitoring) and non-clinical participants (they do not have monitoring);
- Validation system process developed to test each of the blocks individually and tests with the entire architecture;
- Machine learning model training focused on stress detection;
- Experiments using real-time detection in stress-inducing experiments;
- Comparative analysis between the results of clinical and non-clinical patients highlighting the impact of the allostatic load.

1.4 Text structure

This research is structured in seven other chapters besides this one (Chapter 1). Starting with the explanation of the main concepts used in the present work, in the chapter 2, which is divided into three sections. These three sections cover the topics Physiological Aspects (2.1), Internet of Things (2.2) and Artificial Intelligence (2.3). In the chapter 3, we present a systematic review of the literature, which identifies the most important articles in the research area (3.1), in addition to highlighting the most significant contributions of each work (3.2).

Next (in chapter 4), based on the theoretical aspects evaluated and with the bibliographic review carried out, we describe the proposed architecture, which seeks to solve the problems identified in this work, in addition to answering the question of research. This chapter has two sections, one to present an overview of the model (4.1), to describe it at a conceptual and design level, and then present the components used (4.2). Next, in Chapter 5, the stages of development of the prototype are presented, having as reference the proposed architecture. We carried out the development in stages, according to the architecture division. Each section presents a development stage, starting with the Acquisition block (5.2), where the architecture's operating cycle begins, passing through Processing (5.3), Database (5.4) and Pattern Classification (5.5).

After the implementation of the prototype, we present the experiments carried out in Chapter 6. Which we divided into six stages, as the communication stability (6.1), the processing of signals in real-time (6.2), and the third (6.3) covering storage data. The following section (6.4) describes the practical experiment carried out to compose the dataset. We carried out this experiment with participants in partnership with the Graduate Program in Psychology. The last two sections are for classification; the section (6.5) presents the process of the Offline classification experiment. In the section (6.6), we present how we performed the stress induction experiment to perform the detection in real-time.

The chapter 7 aims to gather all the data obtained and perform an in-depth analysis of the data obtained (7.1), architecture stability, the volume of data generated (7.2), as well as evaluating the signals obtained (7.3). The second part of the results chapter (7) presents the results related to the training of the ML model (7.4), in the 7.5 section, we present the results of stress induction experiments to perform detection in real-time. The last section (7.6) presents the results obtained in training to update the Machine Learning model, using data from both ML experiments. Finally, the chapter 8 summarizes all the conclusions obtained with the present work so far. The topics approached are an overview of the architecture, the contributions of the work (8.1), and suggestions for next works and next steps in the development of the architecture (8.2).

2 THEORETICAL BACKGROUND

This section presents and details the basic concepts used in this work. We detail the principal physiological signs, describing each signal and explaining their particularities. In the sequence, we present some concepts of the internet of things and some devices valuable for this context. Lastly, we talk about artificial intelligence. The topics are machine learning techniques, deep learning models, and concepts.

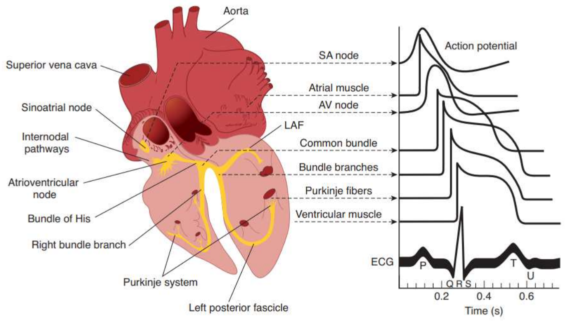
2.1 Physiological Aspects

Physiological signs are relevant ways to check a person's health status, indicating the presence or absence of disease. Many papers address the topic of physiological signals as a means of automatic disease detection by intelligent systems, as can be seen in the works carried out by the authors Ali et al. (2020), Li and Liu (2020) and Murat et al. (2020). We may obtain the analyzed signals through sophisticated and expensive medical equipment. An alternative is to use simpler devices with a low cost, such as wearable sensors. Among the most used signs are the Electrocardiogram, Heart Rate, Blood Volume Pulse, Electroencephalogram, Electromyogram, Electrodermal Activity or Galvanic Skin Response, Respiration, Temperature, Blood Pressure, among others. In this section, we describe the main signs used for this purpose.

2.1.1 Electrocardiogram

Electrocardiogram (ECG) is a graphical representation used to identify the heart's electrical activities. The ECG uses electrodes to obtain these signals. The information obtained based on the ECG signal is heart rate (HR) and heart rate variability (HRV), also known as the R-R interval. The electrocardiogram trace formed by the electrical signals that occur in the heart, composed of the points P, Q, R, S, T, and U, as shown in Figure 1. The P wave results from atrial depolarization. The waves that form the QRS complex correspond to ventricular depolarization. The T wave represents the repolarization of the ventricle (REIS et al., 2013).

Figure 1 – Electrical Activity of the Heart

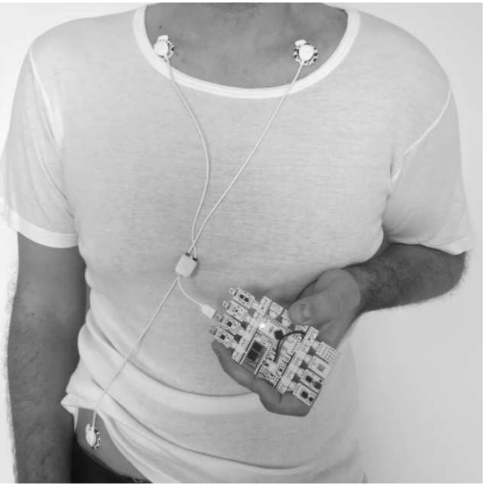


Source: Barrett et al. (2010)

According to Feldman and Goldwasser (2004), HR is the number of heartbeats in a minute. The usual heart rate has values between 60 bpm and 100 bpm. The decrease in HR, called bradycardia, may be observed in conditioned athletes, after periods of meditation or under the effect of medication. The increase in heart rate, called tachycardia, is more frequently observed after exercise, anxiety, and after ingestion of alcohol, caffeine, or nicotine. Heart rate variability (R-R interval) is the time interval between the occurrence of each QRS complex. This interval is typical when its intervals are regular and constant. If there is any irregularity in the signs, it is necessary to investigate (FELDMAN; GOLDWASSER, 2004).

The ECG is one of the tests most commonly used in the studies analyzed, proving accurate in identifying diseases, moods or stress. A standard electrocardiogram consists of twelve recording leads. The points are six frontal, three bipolar (D1, D2, D3), and other unipolar (aVR, aVL, and aVF). Six from the horizontal electrical plane, all of which are unipolar (V1, V2, V3, V4, V5, and V6) (FELDMAN; GOLDWASSER, 2004). For easy exams, as well as in the wearable sensors, we used only the three frontal bipolar electrodes, as shown in Figure 2.

Figure 2 – ECG Electrode Placement



Source: PLUX – Wireless Biosignals, S.A. (2020a)

2.1.2 Blood Volume Pulse

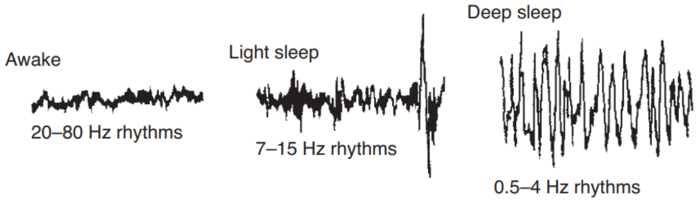
Blood Volume Pulse (BVP) signals are widely used for heart rhythm monitoring and provide simple detection of arrhythmia usually. The method used for this measure is the photoplethysmogram (PPG), which positions a sensor at the fingertip, and it has extensive use in smartwatches. The PPG comprises two components, an infrared light-emitting source and a photodetector. It measures the pulse changes in blood volume in the arteries and capillaries that correspond to changes in the heart rate and blood flow (KIM; KO, 2021). According to Kim and Ko (2021), BVP has applicability in situations where we desire to obtain heart rate variability, but instead of using ECG, use the PPG due to the practicality of measuring. The main reason for this is that it allows greater freedom of movement, but it does not have such high accuracy.

2.1.3 Electroencephalogram

Electroencephalogram (EEG) analyzes brain electrical activity and wave characteristics such as frequency, amplitude, and shape. Usually, this method is non-intrusive and performed through electrodes. Due to the sensitivity of the signals, it is essential to apply the electrodes in the exact location, under penalty of invalidity of the acquired data (KWON; SHIN; KIM, 2018). As stated by Schomer and Silva (2017), EEG measures voltage fluctuations resulting from ionic current within neurons in the brain. Diagnostic applications usually focus on the spectral content of the EEG, that is, on the type of neural oscillations. The typical values of the brain signals observed are between 1 and 20 Hz.

Barrett et al. (2010) highlights that the standard signal obtained through the EEG (4) has a very regular wave with a frequency of 8 to 13 Hz and amplitude of 50-100 μV. However, we can distinguish four types of mental waves according to the individual's state. The first is the beta wave, classified by 14 Hz to 30 Hz, verified in a normal state. The second is the Alpha waves, characterized by 8 Hz to 13 Hz, occurring in relaxed mental states and sleep. Theta waves have wavings of 4-7 per second observed in meditation and deep relaxation periods. Finally, the Delta wave varies from 0.5 Hz to 3 Hz, occurring only in a deep sleep (SCHOMER; SILVA, 2017).

Figure 3 – EEG Typical Signal



Source: Adapted from Barrett et al. (2010)

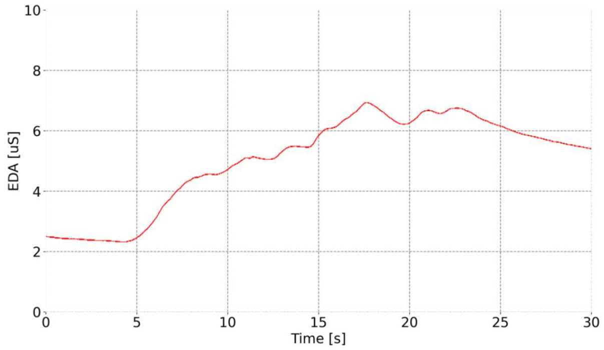
2.1.4 Electromyogram

Electromyogram (EMG) signals to measure the electrical activity of muscles. The muscle movement occurs due to bio-electric signals sent by our brain to the muscle fibers. The electromyogram uses electrodes to measure the amplitude of these signals (PLUX – WIRELESS BIOSIGNALS, S.A., 2020b). We may perform the EMG in an intrusive or non-intrusive way. If performed non-intrusive, it is usually called surface electromyographic (sEMG). According to Merletti et al. (2009), the signals obtained through EMG are essential for several areas. The EMG is helpful in rehabilitation medicine, ergonomics, sports medicine, physiotherapy, and neurophysiology.

2.1.5 Electrodermal Activity

Electrodermal Activity (EDA) is related to the peripheral response arising from the activation of the sympathetic nervous system. This signal is an involuntary response controlled by the central nervous system, acting directly on the activation of the skin's sweat glands, which are responsible for the production of the sweat due to emotional stimulation (DAWSON; SCHELL; FILION, 2007). The EDA is also called skin conductance (SC), galvanic skin response (GSR), electrodermal response (EDR), or skin conductance response (SCR). In the case of signs obtained through wearable sensors PLUX – WIRELESS BIOSIGNALS, S.A., 2020c, the data obtained have Siemens as the unit of measurement, in order of magnitude of microsiemens (µS), as may be seen in the 4.

Figure 4 – EDA Typical Signal - Unfiltered



Source: PLUX – Wireless Biosignals, S.A. (2020c)

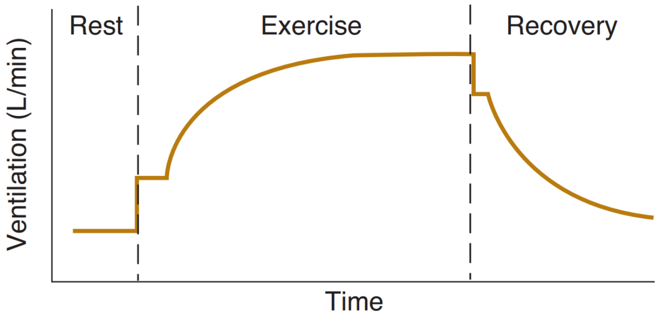
Sharma, Kacker and Sharma (2016) explains that the galvanic resistance of the skin is an accessible and quite sensitive parameter of the nervous system. It reflects the emotional state of the individual. The response appears to increase the electrical conduction of the skin. This increase results from the decrease of the intrinsic resistance of the skin. The most appropriate place to check these signals would be the palms of the hands or the soles of the feet, which may identify approximately two seconds after the stimulation.

2.1.6 Respiratory Rate

Respiratory rate, or simply respiration (RSP), consists of the gas exchange of oxygen (O_2) and carbon dioxide (CO_2) of the organism with the environment. Breathing patterns vary according to age. We expect 18 to 25 breaths per minute for people around six years old. By 10, we usually get values between 17 and 23 breaths per minute. Finally, in adulthood, rage is generally found between 15 and 18 breaths per minute (BARRETT et al., 2010).

As stated by Barrett et al. (2010), the ventilation usually increases immediately after starting a physical activity in the first moments (Figure 5). A few minutes after, the increase becomes slower. However, the process of recovering has similar behavior, with the difference that ventilation has decreased. The wearable sensors generally use piezoelectric sensors to capture breath signals by checking the displacement variation when inhaling and exhaling (PLUX – WIRELESS BIOSIGNALS, S.A., 2020d).

Figure 5 – Representation of respiratory changes



Source: Barrett et al. (2010)

2.1.7 Stress

Stress is a physical and mental response resulting from our responsibilities for daily activities, and it is hard to try to avoid it. It is often difficult to identify the state of the stress, and it is necessary to know the common symptoms. The symptoms include chronic pain, insomnia, trouble sleeping, digestive problems, loss of appetite, difficulty concentrating, and tiredness. So that stress does not become a chronic problem, we must control it. Following some guidelines that may help, such as maintaining a good diet, sleeping enough hours of sleep, about 7 to 8 hours a night, accomplishment regular physical exercise, as well as a decrease in alcohol and coffee intake (LEGG, 2020).

According to Cohen, Janicki-Deverts and Miller (2007), stress has two main divisions of how the body reacts. The first is acute stress, which has a rapid response triggered by the sympathetic-adrenal-medullary axis (SAM). The second is the response to chronic stress following the hypothalamic-pituitary-adrenal (HPA) axis, which has a slow response. These actions result from our autonomic nervous system. We may divide the autonomic nervous system into the sympathetic nervous system (SNS) and parasympathetic nervous system (PNS).

Stress is often associated with other problems, further aggravating the problem. “Stress and anxiety often go hand in hand. Stress comes from the demands placed on our brain and body. Anxiety is when we feel high levels of worry, unease, or fear. (LEGG, 2020).”. Thus, we concluded that stress is undoubtedly a problem that we must monitor and treat, directly affecting the person’s quality of life. Cohen, Janicki-Deverts and Miller (2007) explains that chronic stress frequently causes other diseases, such as cardiovascular disease, depression, increased blood pressure, and decreased immunity.

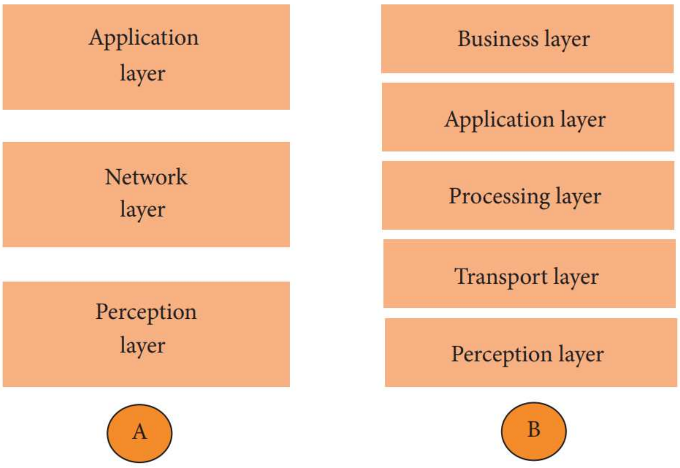
Recent studies have analyzed the effect and impact of allostatic load on the behavior of physiological signs of people who have or do not have some psychological support. As Corrigan et al. (2021) explains, the allostatic load is the metabolic energy composed by biological measurements. Allostatic load captures the dysregulation of various physiological systems as a result of chronic exposure to stress (GUIDI et al., 2021). An overload of allostatic load can generate several problems in various organs. As a consequence, cellular degeneration, hypersensitivity, hypertension, anxiety, and depression (CORRIGAN et al., 2021).

2.2 Internet of Things

2.2.1 Architecture

Di Martino et al. (2018) argues that the Internet of Things (IoT) has a wide variety of technologies and application areas. This application results in a great diversity of solutions with a flexible architecture. The application may be specific, as applied to health, safety, and industry, or generic, resulting in different architectures for each case. Sethi and Sarangi (2017) defines two architectures possibilities for IoT models (Figure 6), a model with three layers and another more specific, formed by five layers.

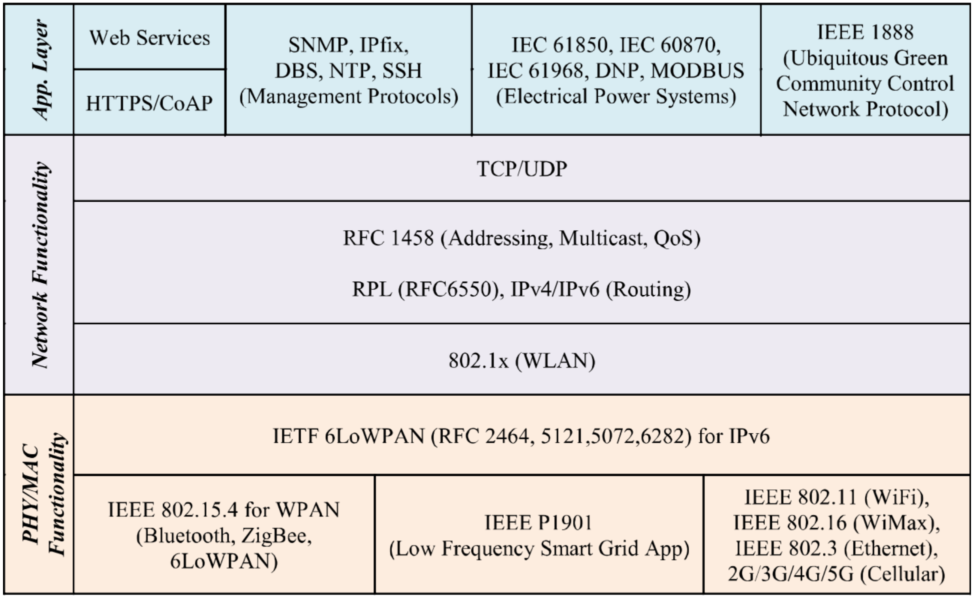
Figure 6 – IoT Layers



Source: Sethi and Sarangi (2017)

In model A, the first layer, called the perception layer, is the physical layer, which has sensors to detect and acquire information about the environment. The second layer, called the network layer, is when communication between devices happens. The devices may be intelligent, standard network equipment, or servers, local or on the cloud in this layer. In the case of the last layer of model A, the application layer is responsible for the services that are available to the user (SETHI; SARANGI, 2017). It is possible to check the details of protocols present in each layer of the model through Figure 7.

Figure 7 – IoT Protocol Layers



App. Layer	Web Services HTTPS/CoAP	SNMP, IPfix, DBS, NTP, SSH (Management Protocols)	IEC 61850, IEC 60870, IEC 61968, DNP, MODBUS (Electrical Power Systems)	IEEE 1888 (Ubiquitous Green Community Control Network Protocol)
Network Functionality	TCP/UDP			
	RFC 1458 (Addressing, Multicast, QoS) RPL (RFC6550), IPv4/IPv6 (Routing)			
	802.1x (WLAN)			
PHY/MAC Functionality	IETF 6LoWPAN (RFC 2464, 5121,5072,6282) for IPv6			
	IEEE 802.15.4 for WPAN (Bluetooth, ZigBee, 6LoWPAN)	IEEE P1901 (Low Frequency Smart Grid App)	IEEE 802.11 (WiFi), IEEE 802.16 (WiMax), IEEE 802.3 (Ethernet), 2G/3G/4G/5G (Cellular)	

Source: Nour et al. (2019)

Related to model B, the Network layer has become two other layers. Another alteration in this model is the addition of another layer after the Application layer. The transport layer is responsible for transferring the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, LAN, Bluetooth, RFID, or NFC. The Processing layer, also known as the middleware layer, stores, analyzes and processes the data from the transport layer. At least, the business layer manages the whole IoT system (SETHI; SARANGI, 2017).

Nour et al. (2019) explains that Smart Healthcare is one of the most relevant applications of the IoT. Smart healthcare uses devices to measure the vital signal of the patients, it is possible to facilitate the diagnosis of the diseases through remote exams or health monitoring. In this context, many technologies are used, such as sensors, actuators, wireless networking devices, medical applications, and wearable sensors. Medical sensors may acquire data from physiological signals. Personal mobile devices are also reliable options in this context, as they have motion and location sensors, such as an accelerometer (ACC) and GPS.

According to Costa et al. (2018) the IoT area has one sub-area, called the Internet of Health Things (IoHT), in which the focus is on the patient monitoring. The authors explain that one of the essential features of this area is machine learning techniques to correlate the data, transforming it into useful information.

2.2.2 Single-board computer

Jovanovic et al. (2014) define the single-board computer (SBC) as a general-purpose computer built on a single board with a microprocessor, memories, communication protocols, digital and analog ports. Due to their flexibility, these devices are often used in automation, industrial and residential systems. The single-board computer has had its capabilities increasing in recent years, allowing these devices to take on more responsibility in IoT networks. An appreciable characteristic of this type of device is the possibility to support an operating system, which allows great flexibility of applications.

In the case of conventional computers, the dominant operating system is Windows. However, the presence of the Linux operating system in embedded systems is much more prevalent. Linux is an open-source operating system characterized by high reliability and easy development JOVANOVIC et al., 2014. It is possible to compare some SBCs in Table 1, it contains four different models from two different manufacturers. We compare them concerning their main characteristics, such as model, processor architecture, the processor's model, number of cores, and frequency of operation of the processor.

Besides the operating system supported by the board, as well as the amount of RAM available. In addition to this more performance-oriented information, we can also see in the Table 1 data about the physical interfaces. Some examples of the physical interfaces listed are the Ethernet interface, the presence of Wi-Fi or not, Bluetooth, which is the model of the video output, and the number of General Purpose Input/Output (GPIO).

Table 1 – Comparision of SBCs

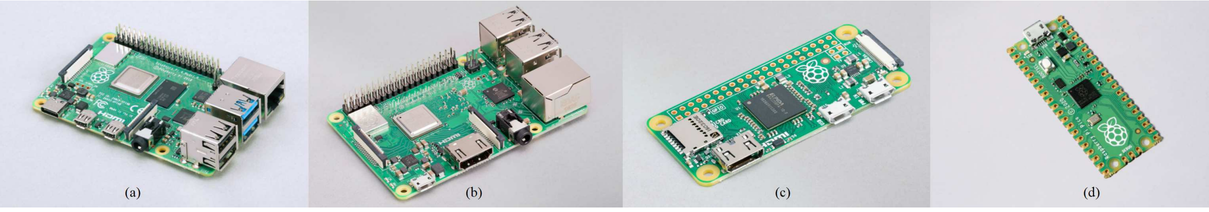
	Raspberry Pi	**Raspberry Pi**	**BeagleBone**	**BeagleBone**
Model	3 Model B+	4 Model B	Black	AI
Architecture	Cortex-A53	Cortex-A72	Cortex-A8	Cortex-A18
Processor	BCM2837B0	BCM2711	AM335x	AM5729
Number of cores 4	4	1	2	
CPU Frequency	1.4 GHz	1.5 GHz	1 GHz	1 GHz
Supported OS	Linux / Windows	Linux / Windows	Linux / Android	Linux
RAM	1 GB	2/4/8 GB	512 MB	1 GB
Ethernet	Gigabit	Gigabit	Megabit	Gigabit
Wi-Fi	Yes	Yes	No	Yes
Bluetooth	Yes	Yes	No	Yes
Video Output	HDMI	micro-HDMI	HDMI	micro-HDMI
GPIO	40	40	2x46	2x46

Source: Adapted from Raspberry Pi Foundation (2021) and Texas Instruments (2020)

Raspberry Pi stands out due to its low cost combined with good processing capacity, as seen in Table 1. This platform is the one adopted for this project. The Raspberry Foundation, from the United Kingdom, is responsible for the device development, with an ARM architecture processor based on the System on Chip (SoC). In addition, the Raspberry Pi has the possibility of Bluetooth and serial communication via SPI (RASPBERRY PI FOUNDATION, 2021).

Raspberry has a wide variety of boards available, as shown in Figure 8. Figure 8 (a) shows the Raspberry Pi 4 Model B, the board with the highest processing power developed so far, with models with different amounts of RAM, USB 3.0, and the possibility to connect up to two HDMI monitors. In Figure 8 (b), we can see the Raspberry Pi 3 Model B +, which has many similarities with the model presented previously (Figure 8 (a)), but with the main differences the processor and the available RAM.

Figure 8 – Raspberry models



Source: Adapted from Raspberry Pi Foundation (2021)

Figure 8 (c) shows the Raspberry Pi Zero W, a model focused on low cost, reduced processing, and also with a smaller size. This model features a processor with only a 1 GHz core and 512 MB of RAM, looking very similar to the BeagleBoard models. It has GPIOs available, as well as Bluetooth and Wi-Fi connection. Finally, we have the Raspberry Pi Pico, exemplified in Figure 8 (d). Unlike the previous ones, this card does not aim to actuate as a computer but rather as a microprocessor. However, it has a dual-core ARM processor with only 264 KB of internal RAM.

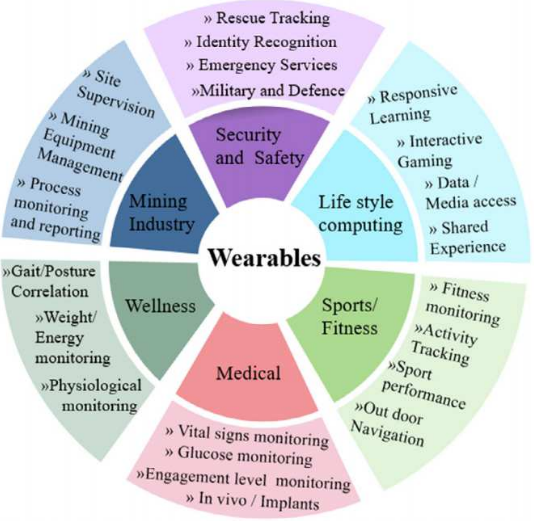
2.2.3 Wearable Sensors

Nasiri (2019) defines wearable sensors as devices with the ability to be integrated directly over the body as a way to monitor the individual's health status. Due to the nanosensors integrated into devices, these devices provide an excellent non-intrusive approach. These sensors usually provide information related to physiological signals, which allows monitoring in several areas, not only in health, as can be seen in Figure 9.

Jovanov et al. (2005) argues that wearable sensors, when integrated with solutions with artificial intelligence, allow automatic detections of the user's health status. They also allow the sending of alerts to health professionals based on previously defined physiological parameters. Wearable sensors can assist in monitoring chronic diseases or even preventing them. Wilson and Laing (2018) defend the use of wearable sensors in rehabilitation and postoperative situations. Wilson and Laing (2018) explains that for good use of wearable sensors, several other devices must be close to the system, such as actuators, controllers, microcontrollers, source of energy, acquisition software, and storage location.

There are several types of wearable sensors available nowadays, each aimed at a kind of signal, focusing on different areas, such as for personal use, we can mention the Apple Watch (APPLE WATCH, 2021). It has good accuracy for measuring data such as oxygenation in the blood and electrocardiogram. The measure happens through green, red, and infrared LEDs.

Figure 9 – Wearable Sensors Application



Source: Sreenilayam et al. (2020)

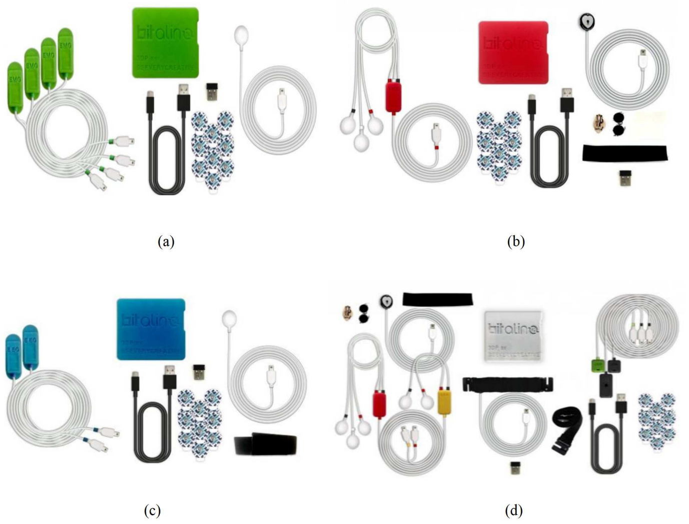
These LEDs shine light into the blood vessels of the pulse, and the photodiodes measure the amount of reflected light. Concerning wearable sensors for sports use, we can mention the company Polar (POLAR, 2021), which develops commercial products aimed at physiological monitoring in physical activities, such as watches with heart monitors and integrated GPS or straps to monitor the variation of heart rate.

The company PLUX (PLUX WIRELESS BIOSIGNALS, 2021), developer of the wearable sensor, already mentioned, BITalino has a wide variety of models. The models have a wide variability of objectives, such as MuscleBIT (Figure 10 (a)), aimed at monitoring the electrical signals of muscles. The HeartBIT (Figure 10 (b)) to monitor the cardiovascular system through sensors such as ECG and PPG. The NeuroBIT (Figure 10 (c)) aimed at monitoring EEG signals. Finally, the PsychoBIT (Figure 10 (d)) is a set of sensors aimed at monitoring a set of signals influenced by the nervous system, influenced by psychological effects, such as stress.

In addition to these kits presented, there are also options with only the controller and unit sensors, which allows for more personalized development when desired. The model that we use in this work is PsychoBIT (Figure 10 (d)) due to the wide variety of sensors, which can be identified both for detecting mood changes, detecting stress, and, also, cardiac monitoring (PLUX WIRELESS BIOSIGNALS, 2021). The communication of this wearable sensor, like the vast majority, uses Bluetooth to perform a data transfer. Another essential feature is the possibility of using the battery as a power source. Among the available sensors, we have:

- Electrocardiography (ECG) Sensor
- Electrodermal Activity (EDA) Sensor
- Respiration (PZT) Sensor
- Light (LUX) Sensor

Figure 10 – BITalino Models



(a) (b)

(c) (d)

Source: Adapted from PLUX Wireless Biosignals (2021)

- PulseSensor (PPG)
- Pushbutton (BTN) Sensor

2.3 Artificial Intelligence

In this section, we make a short description of artificial intelligence. Initially, we talked about Machine Learning and, next, talking about Deep Learning. Artificial Intelligence (AI) is analogous to human intelligence but performed by software. Kaplan and Haenlein (2019) defines artificial intelligence as a system capable of learning based on data provided, making a relationship between information to identify patterns. Among the possible data sources that AI can use are IoT systems.

2.3.1 Machine Learning

Machine learning (ML) is one of the sub-areas of AI, which aims to develop models to identify, classify, predict data, and perform systems control. Supervised learning is a possible approach to intelligent systems based on data provided by the user. In this case, the system tries to infer if they can group the data by general parameters. Another possibility is unsupervised learning, which performs the classification based on the data characteristics. Machine learning, although very useful, is a tool that we cannot apply in the same way to all problems HAYKIN, 2009.

According to Heaton (2015), ML has great capacity for generalization. Generalization consists of classifying a dataset not yet seen previously correctly. If the model has low generalization capacity, it usually occurs due to overfitting in the training of the data, which results from low classification rates for new instances. There is a wide variety of algorithms capable of detecting patterns. We can mention Linear Regressions (LR), Decision Tree (DT), Random Forest (RF) Algorithm, Naive Bayes Algorithm, K-Nearest Neighbors (kNN) Algorithm, Support Vector Machine (SVM) Algorithm, and Artificial neural networks (ANN).

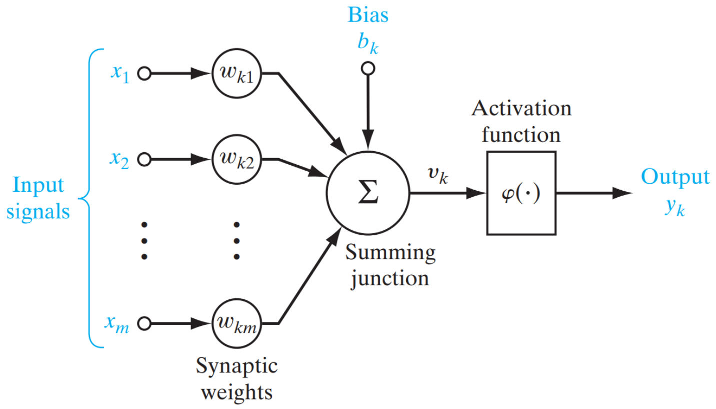
Linear regression draws a straight line from a relation in a scatter plot. The data is displayed graphically, and the algorithm tries to draw a line that separates the data. This line summarizes a relationship between the data of two variables and can also make predictions. This type of algorithm has a recommendation for datasets that have any constant growth/de-growth trend (HAYKIN, 2009). DT is a supervised non-parametric Machine Learning method that makes decisions walking from the root node to the leaf node. The purpose of a decision tree is to decompose all available data and then tries group with their peers about the defined metric (BISHOP, 2006).

Random Forest Algorithm is a supervised learning algorithm that creates decision trees randomly, thus forming a forest. This method uses each tree in the choice of the final result. The RF adds the elements randomly, and instead of looking for the best feature to perform the separation of the nodes, the algorithm searches for the best feature (RUSSELL; NORVIG, 2009). The Naive Bayes algorithm is a probabilistic classifier based on the Bayes Theorem. Among the possibilities of applications is the classification of an e-mail as SPAM or Non-SPAM and identifying a subject based on its content, such as natural language processing (NLP) and medical diagnostics. The method is usually applied when the attributes that describe instances are conditionally independent (BISHOP, 2006).

kNN is a non-parametric algorithm, also classified as a lazy algorithm because it does not require training data to generate the model. All data from the dataset are used in the test, making training fast and slow testing and validation. kNN has a variable called K, defined as the main parameter to be selected (BISHOP, 2006). SVM is a supervised learning algorithm focusing on datasets training and classification. This method also can actuate on regressions, most indicated for binary sorting. SVM operates by seeking a separation line, called a hyperplane, between the class data to maximize the distance between the nearby points and minimize the error. SVM has four types of functions, also called the kernel, which is linear, quadratic, polynomial, and radial (HAYKIN, 2009).

ANNs are ML models based on the activity of biological neurons. Analog to the brain human, learning through training and weight adjustment. The main example of an ANN network is the Multilayer Perceptron (MLP), an architecture with greater use in Artificial Neural Networks and usually applied to supervised learning, widely used in standard classification and prediction of values. The network has an input, hidden layers that perform processing and output (HAYKIN, 2009). The artificial neurons, as can be seen in Figure 11, have several inputs, being able to suffer external stimulus, varying from x_1 to x_m, or stimulation of other neurons. Inputs are weighted, modified during the learning process (w_{k1}, w_{k2}, ... w_{km}). All inputs are added to the next neuron and sent to an activation function (Haykin (2009)).

Figure 11 – Model of a Neuron



Source: Adapted from Haykin (2009)

As stated by (Haykin2009), the activation functions limit the possible range for the amplitude of the output signal (y_k) to a finite value. Typically this range of the neuron output is normalized from $[0.1]$ to $[-1.1]$. Among the most used activation functions, we can mention the Step Function, or Heaviside, which allows the neuron output to be only positive or negative, indicated for binary classifications. The Sigmoid function with increasing characteristics behaves between linear and non-linear. The Rectified Linear Unit (ReLu) function is currently one of the most used activation functions. Convolutional Neural Networks (CNN) is the main application of the ReLu, because of an efficiency close to more advanced functions. Finally, we have Softmax, which transforms each output to values in the range between 0 and 1, and the sum of all outputs does not exceed 1, widely used for multi-class classification BISHOP, 2006.

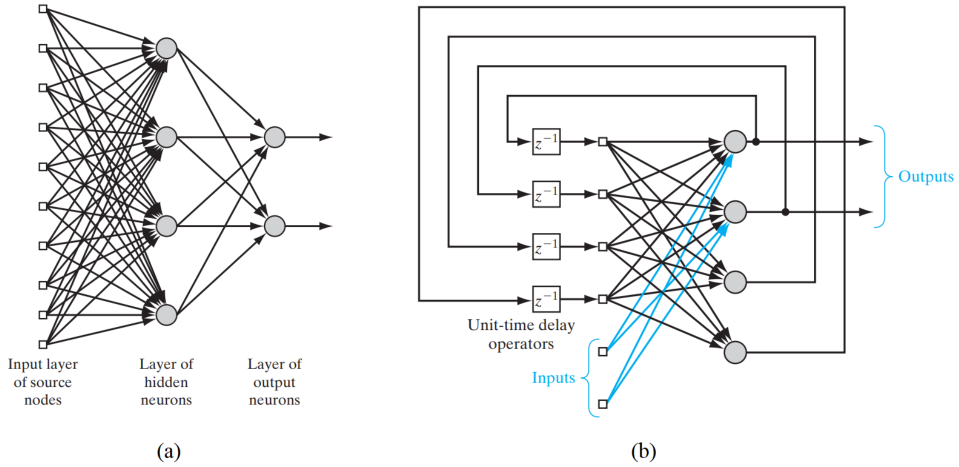
2.3.2 Deep Learning

The traditional ML techniques have a negative aspect, the need to extract features. The feature extraction requires a wide dominance of both the algorithms and the data. The scientists seeking to solve this problem developed the Deep Learning (DL) method, an AI approach in which computers learn to understand the world and identify patterns. This learning happens through models represented graphically through multiple layers of processing BISHOP, 2006.

DL models have an automatic feature extraction through a cascade of several layers. These layers have non-linear processing units for the extraction and feature transformation. As an ANN, this technique applies a series of layers that actuate similarly to a neuron. The neurons perform the processing of a small part of the total information. There are two main classes of deep neural networks. The first we can say is the feedforward (Figure 12 (a)), which has as a characteristic of decision-making only based on the information received in the previous layer. The second type is the recurring network (Figure 12 (b)), whose main difference is decision making based on decisions previously taken HAYKIN, 2009.

When it comes to models of approach to DL models, we can mention three main ones. The

Figure 12 – Types of Neural Network



Source: Adapted from Haykin (2009)

main models are Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). CNN is a class of artificial neural networks, classified as feedforward, which has as a characteristic the decision making based only on the information received and passes its decision to the neurons of the next layer. Processing and analysis of digital images is the main application. Convolutional networks follow biological processes, such as the connectivity between neurons in the organization of the visual cortex. The convolution is part of the CNN recognition process, which has three stages: the convolution of each input layer; the application of a non-linear activation function; subsampling, called pooling (BISHOP, 2006).

RNN is a neural network that detects sequences in data streams, mapping output to an input, depending on the output of the previous layers. In other words, we can say that the RNN makes decisions based not only on the data received but also on decisions taken previously. As such, recurrent networks have two sources of input, the present and the recent past, which combine to determine how they respond to new data. Due to this characteristic, we can say that an RNN has memory (HAYKIN, 2009). LSTM is an RNN architecture that has the feature of storing values in intervals. We generally use this network in scenarios where we desire to classify, process, and predict time series. One of the main appreciable differences in this model is that it may store data at much higher intervals when compared with an RNN. It seeks to solve a common problem in RNNs, in which the vanishing gradient occurs, which consists of the loss of gradient information in networks with many layers (HEATON, 2015).

Some functions are valuable when dealing with DL, such as pooling, which is a network evolution process. The pooling groups the features in a higher-level layer, generated from lower layer patterns. Another relevant function is the Dropout, a function widely used to reduce network overfitting. It is necessary due to the large number of neurons in the hidden layers (HEATON, 2015).

2.3.3 Evaluation

According to (Haykin2009), an efficient evaluation of the trained algorithm or model is essential to certificate the efficiency of the classifiers. We can carry out the evaluation process through quality metrics or a confusion matrix. After training, we need a method to assess how many instances were classified correctly and incorrectly in each class of the problem.

Firstly, we must mention the accuracy (Eq. 2.1), which is the most used method to evaluate. The accuracy evaluates the correctness rate concerning the number of correctness in the model. Corresponds to the number of correct answers that the model obtained based on the number of tested instances. Precision (Eq. 2.2) is the rate of samples labeled as "True" which are effective in the "True" class. Recall (Eq. 2.3), also known as sensitivity, is the measure that corresponds to the proportion of positive examples that are correctly classified. F1 score (Eq. 2.4) consists of performing the harmonic average between precision and recall. The main characteristic of this metric is the fact that if the precision, or the recall, is zero or very close to that, the F1 score is shallow (HEATON, 2015).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

$$F_1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.4}$$

The confusion matrix has two dimensions, the true classes, and the predicted classes. For a confusion matrix with two classes, the matrix has its choices structured to classify whether the event occurred (positive class) or not (negative class). Thus, when a sample is positive and receives the classification as positive, it is called True Positive (TP). Moreover, if the negative sample is classified correctly, we have True Negative (TN). Conversely, if the negative example has a positive classification, it is called False Positive (FP). On the other hand, if the positive sample received a negative classification, we say that we have a False Negative (FN). Another relevant metric is the Receiver Operating Characteristic (ROC) Curve, an interesting metric for tasks with disproportionate classes, measuring the area under a curve formed by the graph between the rate of positive examples, which are positive, and the rate of false positives (HEATON, 2015).

Haykin (2009) present three other relevant metrics the Mean Squared Error (MSE) (Eq. 2.5) is one of the most used, responsible for calculating the average of the squared model errors. Hence, the more minor differences have less impact than the more prominent differences.

Derived from the MSE, it is possible to calculate the Root Mean Squared Error (RMSE) (Eq. 2.6), which consists of a measure that calculates the errors between observed values (real) and predictions (hypotheses). Mean Absolute Error (MAE) (Eq. 2.7) calculates weights by squaring all values, assigning the same weight to all differences. Finally, we have the Mean Absolute Percentage Error (MAPE) (Eq. 2.8) calculates the average percentage of the absolute deviation between forecasts and reality.

$$MSE = \frac{1}{n}\sum_{i=1}^{\mathrm{n}}(Y_i - \hat{Y_i})^2 \tag{2.5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{\mathrm{n}}(Y_i - \hat{Y_i})^2} \tag{2.6}$$

$$MAE = \frac{1}{n}\sum_{j=1}^{\mathrm{n}}|Y_i - \hat{Y_i}| \tag{2.7}$$

$$MAPE = \frac{100\%}{n}\sum_{j=1}^{\mathrm{n}}|\frac{Y_i - \hat{Y_i}}{Y_i}| \tag{2.8}$$

In the next chapter, chapter 3, we will present the state of the art of related works to this work. First, we list the main points of the work, highlighting methodologies and results obtained. Subsequently, we performed a critical evaluation of the results to verify the current research gap and the strengths of the current papers.

3 RELATED WORK

This chapter discusses the studied papers related to this research. The reviewed works seek to guide us concerning the advantages and disadvantages of wearable sensors in health monitoring and pattern identification. We intended to verify which ones are most used nowadays and their applications. Another relevant topic in our research is identifying the most appropriate approaches for developing a pattern prediction model for different physiological signals. We seek to design a flexible signal acquisition architecture that accepts different wearable sensors with distinct signs.

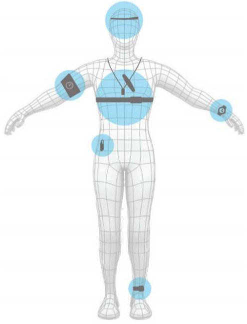
Initially, we do a brief description of the works. In the sequence, we analyze these papers, highlighting the principal contributions and seeking opportunities. To select the most relevant papers on the subject of this dissertation, we carried out exploratory research in scientific works. Some studies analyzed do not meet both requirements. However, they have relevance in at least one of the two topics of the work. Whether the importance of the current artificial intelligence approach or wearable sensors for physiological signals acquisition. We have surveyed the CAPES website, Google Scholar, Elsevier, IEEE Explore, and ACM. For each repository, we used the following strings: "wearable"; "machine learning AND physiological signs"; "deep learning AND physiological signs"; "wearable AND artificial intelligence"; "stress detection"; "stress monitoring". We have filtered the papers from 2017 to 2021. After the selection outlined above, we have analyzed 20 papers.

3.1 Related Work Analysis

This section relates to some recent works relative to wearable sensors and artificial neural networks applied to physiological signs. We have reviewed several papers with this theme to check the state-of-the-art related to the application of Artificial Intelligence in signals obtained through wearable sensors. According to Sreenilayam et al. (2020) and Rienzo and Mukkamala (2021), in reason of the current increasing demand for wearable sensors to monitoring people's health, these devices are more and more accurate in their measures. The authors also highlight that wearable sensors are an excellent option for monitoring and preventing diseases at a low cost. In addition to having a wide variety of applications, they are being able to acquire data such as cardiovascular, ECG, EMG, EEG, EDA, accelerometer (ACC), a global positioning system (GPS), temperature sensors (TEMP), metabolic, gastrointestinal monitoring, sleep, neurology, movement disorders, mental health, maternal, and pulmonary health (DUNN; RUNGE; SNYDER, 2018). However, during the acquisition of the signs, the wearable sensors can measure noise with the raw signs. Consequently, the literature recommends analyzing the signal and applying some filters to eliminate these noises with various pre-processing techniques (MURAT et al., 2020).

As stated by Jacobsen et al. (2021), in the hospital environment, it is essential for efficient medical evaluation signals such as HR, BP, oxygen saturation, and temperature (Figure 13). Wearable sensors are a relevant way to get reliable signals at a low cost. The authors proposed a review of the current papers to prove and discover the real situation of wearable sensors applied in this context. After the revision of the works, Jacobsen et al. (2021) claim that wearable sensors have great potential in the medical care area and already have many successful cases. Nevertheless, they emphasize that it is essential to choose adequate wearable sensors according to the context. Mishra et al. (2020) describe a work that relates to the use of wearable sensors to detect COVID-19 cases. According to the authors, wearable sensors are valuable devices for detecting COVID-19 patients. The authors have used smartwatches for data acquisition like Fitbits and Apple watches. The process happened with about 5,000 participants with 32 infected cases. The authors observed an elevation in resting heart rates and increased heart rates relative to the number of steps. The study showed that at last, 63% of the COVID-19 cases could identify before the first symptoms.

Figure 13 – Example of Wearable Sensors Placement



Source: Adapted from Elenko, Underwood and Zohar (2015)

Kim and Ko (2021) explains that wearable sensors are an alternative to monitoring arrhythmia based on sensors like ECG or PPG. The authors also emphasize that an approach combining both sensors has been showing excellent results. The researchers combined two wearable sensors, the ATP100 for the ECG and the AliveCor for the PPG. However, Kim and Ko (2021) warns that there are still many legal issues surrounding telemedicine and wearable sensors. Han et al. (2017) carried out work intending to detect stress. For it, they applied a stress induction protocol, the Montreal Imaging Stress Task (MIST). The authors acquired the signals using unspecified wearable sensors. The experiment had the participation of 39, which measured the RSP and ECG data. The best result was achieved using a combination of Random Forest and Support Vector Machine, obtaining an accuracy of 84% for the stress classification with three classes and 94% for binary classification.

In a complete study carried out by Schmidt et al. (2018), they collected data through a wearable sensor to detect stress. The authors used RespiBAN to acquire data in this study, such as ECG, EDA, EMG, RSP, and TEMP. After the acquisition, the authors consolidated the data into a dataset. The protocol performed for data acquisition had four main stages. Being the baseline, in which the participant sat neutrally. The second is called amusement, in which the participant watches some fun videos. The next was the stress stage, in which the participant had some stressful tasks to perform. The protocol followed was the TSST. Finally, they instructed the participant to calm down through meditation. The authors have used Machine Learning algorithms to detect patterns. The techniques that Schmidt et al. (2018) used was DT, RF, and kNN. They got for binary classification 87.00%, 92.28%, and 74.20%, respectively. The results for classification with three classes were between 63.56% and 79.57%.

Al-shargie et al. (2017) presented a paper in which he uses the wearable sensor Discovery 24E to obtain EEG signals to perform stress detection. The author explains that it was necessary to apply a Butterworth filter to reduce noise. The main reason is that the EEG signal is sensitive. The authors applied machine learning algorithms to perform stress detection, applying the Support Vector Machine algorithms with error-correcting output code (ECOC) to classify them in three levels. Thus, the authors obtained satisfactory results of 94.79% of accuracy. Kwon, Shin and Kim (2018) developed a Deep Learning model, more specifically a Two-Dimensional CNN (2D), based on data from the DEAP dataset. They obtained an effectiveness of 73.4% through the EEG and GSR analysis. Nevertheless, the data was processed using a wavelet transform before applying the DL model.

Based on the arguments before listed connected wearable sensors, Murat et al. (2020) explains that Deep Learning techniques have become an important alternative to classify signals. For example, the electrocardiogram (ECG) is an exam that measures a person's heart rate (HR). They have analyzed five different classes of signals with more than 100,000 beats using the MIT-BIH arrhythmia database (MOODY; MARK, 2015). They are classified using as Normal Beats (NB), Atrial Premature Beats (APB), Left Bundle Branch Block (LBBB), Right Bundle Branch Block (RBBB), and Premature Ventricular Contraction (PVC). During the experiments was used the, Scikit-Learn and Keras with TensorFlow. Both are essential Python libraries focused on Machine Learning (ML) algorithms. Firstly was made the pre-processing of the raw data. In sequence, they determined the patterns using ML techniques. After the pre-processing, the authors separated the data into three groups. The first group is the "Training Data", with 80% of signals. The remaining signals had 10% as validation and 10% as testing. The point used as reference was the R peak, with 99 samples before and 160 after the R peak. The authors (MURAT et al., 2020) argue that the results obtained in the study are satisfactory and solve a relevant problem in the field, which obtained 99.26% of accuracy.

Mozos et al. (2017) used three Machine Learning algorithms to identify stress in young people. They acquired the signals through pairs of sensors and followed the gold standard protocol for stress induction, the Trier Social Stress Test (TSST). During the experiments carried out with all participants, the accuracy obtained was above 90%. However, when performing the training experiments removing half of the participants and using this group for tests, the result obtained dropped to about 60% (MOZOS et al., 2017). Ahuja and Banga (2019) carried out an experiment in which they monitored young people during exam preparation and while using the internet. This work aimed to detect mental stress. The experiments had 206 students' data, and for the AI experiments, the authors used Linear Regression (LR), Naïve Bayes (NB), RF, and SVM algorithms. Ahuja and Banga (2019) used the 10-Fold Cross-Validation to check for possible overfitting. The best values obtained were with Support Vector Machine, with 85.71% accuracy.

Bringas et al. (2020) propose a study to identify the Alzheimer's stage in patients using Deep Learning (DL) models through the data obtained using accelerometers. The authors conducted the study with 35 participants with Alzheimer's Disease (AD) in different stages. Based on the data collected, it was possible to identify the level of mobility for each patient and estimate the disease stage. They developed the model using a Convolution Neural Network (CNN), a class of ANN focused on two-dimensional sets, and it does not require any pre-processing. The study had an accuracy of approximately 90%, confirming the hypothesis raised by the researchers. Although, Murat et al. (2020) points out that the models using only CNN have the problem of designing a structure compatible with several datasets. Yang et al. (2021) developed a motion monitoring system to acquire and detect the lower limb motions. The prototype has inertial measurement units (IMUs) and flexible membrane compression sensors as the base. A K-nearest neighbor was applied to detect the patterns. The results obtained were 99.96% for kNN and 99.57% for SVM.

As stated by Faust et al. (2018) that have reviewed 53 papers about Deep Learning methods applied to physiological signals, the research focused on EMG, EEG, ECG, and electrooculography (EOG). The author highlights the importance of the dataset's large volume of data on an excellent performance in training the model using Deep Learning. The result obtained using a combination between CNN and LSTM have presented a relevant improvement in terms of the results using just one of these methods (SALEHZADEH; CALITZ; GREYLING, 2020). Nevertheless, in contrast, Murat et al. (2020) explain that besides the results obtained using Convolutional Neural Network usually being better than other methods, this kind of Neural Networks demands a high computational cost and can be considered a disadvantage. Another disadvantage highlighted is that this method requires a large dataset for proper training. Moreover, a hybrid system as proposed for Salehzadeh, Calitz and Greyling (2020), also is detached by Murat et al. (2020), such as CNN-LSTM, tends to have better and more successful results. However, the problem generated is more extensive than before because LSTM models require more time and processing when compared to other algorithms. Besides that, this method needs more volume of data.

Wang and Guo (2020) argue that the use of DL in the context of stress detection shows better results, as CNN captures local dependency between multiple sensors, and the RNN could learn the material resources of the time-series signals. Li and Liu (2020) conducted a study comparing the results of a DL model and an ML algorithm using the same dataset (WESAD SCHMIDT et al., 2018). The experiments have shown significant accuracy improvement. The experiment analyzed EMG, ECG, GSR, HR, RSP, ACC, and BVP. The results for DL were 99.55%, besides

Linear Discriminant Analysis (LDA) was 76.50%, AdaBoost 75.21%, and Random Forest with 76.17%.

Ali et al. (2020) present a study developed that intends to predict heart disease efficiently to treat cardiac patients before a heart attack occurs. The authors (ALI et al., 2020) seek to resolve this problem using Deep Learning models with healthcare data about heart diseases. The intelligent healthcare system proposed to combine DL methods and feature fusion approaches. The feature fusion method combines sensors data and electronic medical records (EMR). In the sequence, they have eliminated all irrelevant or redundant information. The developed system is compared with conventional classifiers to evaluate the results. The proposed system had an accuracy of approximately 98.5%. Another fact highlighted by the authors is the accuracy obtained in the Deep Learning model. The accuracy obtained was highest compared to the results of the other methods. The authors compared the DL model with the Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, and Naive Bayes.

Tripathi et al. (2018) describe the model developed to recognize emotions using EEG signals in a group with 32 participants, using two different neural networks, the first being a Deep Neural Network (DNN), consisting of an Artificial Neural Network with multiple layers, and the second a CNN. The authors have applied four layers to the first model, with 5000 nodes in the first layer and 500 neurons in the second layer. The third layer added 1000 neurons and 2, or 3, nodes in the fourth layer. The second model has used two convolutional layers to convert the data into 2D image format, with Tan Hyperbolic in the first layer and MaxPooling in the second layer. The accuracy obtained in the model was between 81% and 73% for the two classes. Another important subject related to pattern detection is the feature extraction in physiological signs.

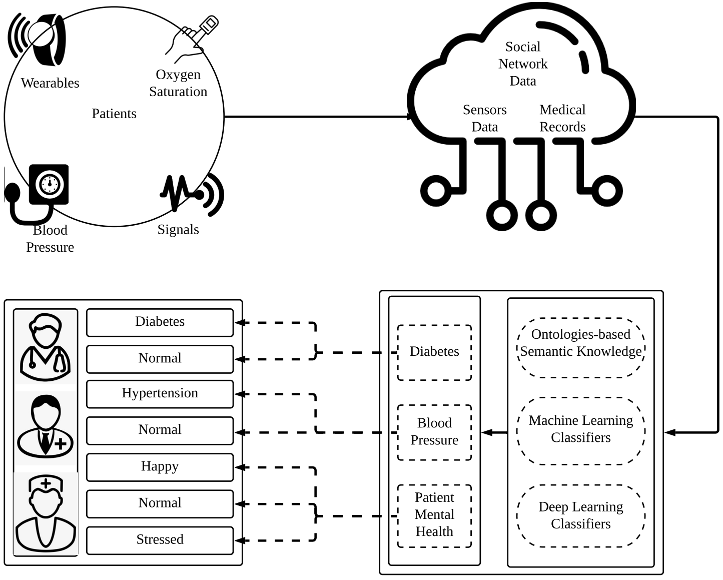
Nevertheless, despite this, it is a complicated theme because of the unique characteristics of each signal. According to Ferreira, Attrot and Sakuray (2011), the analysis for feature extraction needs to be done for each sign, taking into account the information present in each signal. For example, the ECG has, at last, four different features to extract, the R-R interval, P-P interval, Q-T interval, and the QRS complex. We can mention several feature extraction methods, such as derivates, digital filters, wavelet transform, neural networks, and frequency analysis. Furthermore, in adding, Faust et al. (2018) says that the researcher must evaluate various feature extraction algorithms for each kind of physiological sign. However, on the other hand, Santamaria-Granados et al. (2018) suggests in their work to use a CNN for automatic feature extraction.

Related to stress detection, a point that has been studied recently is the influence of allostatic load on the behavior of the signals of individuals. Corrigan et al. (2021) highlight the use of heart rate variability as a way to monitor stress and allostatic load. The allostatic load consists of metabolic energy composed of biological measures. It captures the dysregulation of multiple physiological systems as a result of chronic exposure to stress (GUIDI et al., 2021). HRV decreases in response to a stressful situation, whether physically or cognitively (CORRIGAN et al., 2021). They observed that HRV recovery is usually slow in response to higher magnitude stressors. They also conclude that people with allostatic load disorder usually have a lower standardization HRV response. The authors emphasize that further studies are needed to further the usefulness of HRV in assessing allostatic load.

The use of AI for stress detection has several limitations and uncertainties that we need to deepen. Research indicates that experiments in controlled laboratory environments have high precision for detecting stress (CAN; ARNRICH; ERSOYA, 2019). The stress level is often significantly different from the induced stress level outside of controlled environments. They also emphasize that wearable sensors are excellent means to carry out these experiments. The participants tend to accept easily non-intrusive and discreet devices. Based on the state-of-the-art, we can conclude that there is still a lot to be explored in stress detection.

Ali et al. (2021) proposed a work that consists of integration between wearable sensors and social networking for data collecting. The authors claim that this method is an efficient health-care monitoring, Ali et al. (2021) describe the scenario with a cloud environment and a big data engine. As can be seen in Figure 14, the big data stage can have a relation with data mining, ontologies, and DL. The system developed for the authors (ALI et al., 2021) can classify the patients' health status based on their physiological data such as BP, diabetes, social networking data, and drug review. The accuracy obtained through this method was around 79% and 89%, using only LSTM. Furthermore, between 90% to 94% when combined LSTM and ontologies. The authors used the following metrics accuracy, precision, recall, function measures, RMSE, and mean absolute error (MAE). The authors performed experiments with CNN (70%), MLP (80%), SVM (73%), Fuzzy Logic (83%), Logistic Regression (72%), Random Forest (70%), and K-Nearest Neighbors (KNN) (58%). The PhysioNet MIMIC-II dataset obtained an accuracy of 88% with this dataset applied to training.

Figure 14 – Example of an Integrated Architecture



Source: Adapted from Ali et al. (2021)

3.2 Evaluation and Research Opportunities

Based on the survey results, all the papers were related and compared. We assessed which are the most commonly used signals in the works. We also intended to verify how they obtained the data and the method used to generate the model.

Table 2 has columns that contain the information concerning the papers previously mentioned. The first column presents the author's name and the year of publication of the work. The second column shows the principal theme of the work. The third column refers to the signals analyzed by the paper. The fourth column presents the data source that the authors have used, which can be two different types. According to the application of the work, the data source may be a dataset or some wearable sensors. Finally, we have the fifth column, which refers to the method used for the classification. It can be ML algorithms or DL models.

Table 2: Summary of the related works

Paper	**Application**	**Sign / Sensor**	**Data Source**	**Method**
Ahuja and Banga (2019)	Stress Detection	ECG	Dataset	RF; SVM; RL; NB
Al-shargie et al. (2017)	Stress Detection	EEG	Wearable	SVM-ECOC
Ali et al. (2020)	Arrhythmia Detection	ECG; BVP	Dataset	DL
Ali et al. (2021)	Health Monitoring	ECG; BVP	Wearable	LSTM
Bringas et al. (2020)	Stage of Alzheimer's Disease	ACC	Wearable	CNN
Can, Arnrich and Ersoya (2019)	Stress Detection	ECG	Wearable	RF; SVM; MLP
Corrigan et al. (2021)	Stress Detection	HRV	Wearable	-
Dunn, Runge and Snyder (2018)	Health Monitoring	ECG; HR; EMG; EDA; TEMP; ACC; GPS	Wearable	-
Faust et al. (2018)	Arrhythmia Detection	ECG; EEG; EMG; EOG	Dataset	CNN; LSTM
Guidi et al. (2021)	Stress Detection	ECG	-	-

Han et al. (2017)	Stress Detection	ECG; RSP	Wearable	Random Forest-SVM
Jacobsen et al. (2021)	Health Monitoring	ECG; PPG; TEMP; ACC	Wearable	-
Kim and Ko (2021)	Arrhythmia Monitoring	ECG; PPG	Wearable	-
Kwon, Shin and Kim (2018)	Emotion Recognition	EEG; GSR	Dataset	CNN
Li and Liu (2020)	Stress Detection	ECG; HR; EMG; GSR; RSP; BVP; ACC	Dataset	MLP
Mishra et al. (2020)	COVID-19 Detection	ECG; HR	Wearable	-
Mozos et al. (2017)	Stress Detection	ECG; EDA; EMG	Wearable	SVM, MLP, RF
Murat et al. (2020)	ECG Recognition	ECG	Dataset	CNN-LSTM
Rienzo and Mukkamala (2021)	Health Monitoring	ECG; PPG; BP; ACC	Wearable	-
Salehzadeh, Calitz and Greyling (2020)	ECG Recognition	ECG	Wearable	CNN-LSTM
Santamaria-Granados et al. (2018)	Emotion Recognition	ECG; EEG; GSR	Wearable	CNN
Schmidt et al. (2018)	Stress Detection	ECG; EMG; EDA; TEMP	Wearble	DT; AB; LDA; KNN
Sreenilayam et al. (2020)	Health Monitoring	ECG; BP; TEMP; RSP	Wearable	-
Tripathi et al. (2018)	Emotion Recognition	EEG	Dataset	DNN; CNN
Wang and Guo (2020)	Stress Detection	ECG; EMG; HR; GSR; RSP	Wearable	AB; MLP

Source: Elaborated by the author.

A relevant aspect that we concluded about the analyzed papers is the possibility of acquiring data to develop our signal dataset, as made by (SCHMIDT et al., 2018 and SANTAMARIA-

GRANADOS et al., 2018). Other researchers like (ALI et al., 2020, FAUST et al., 2018, and KWON; SHIN; KIM, 2018, LI; LIU, 2020) have used these datasets in their applications. In other words, we can consider that the composition of a dataset represents a double contribution to the area. Not only contributing to the research but also providing a database helpful to other research. Moreover, it stayed explicit in the papers the great variety of signals that we may acquire with wearable sensors. Besides, many studies have used the same signals to detect different anomalies.

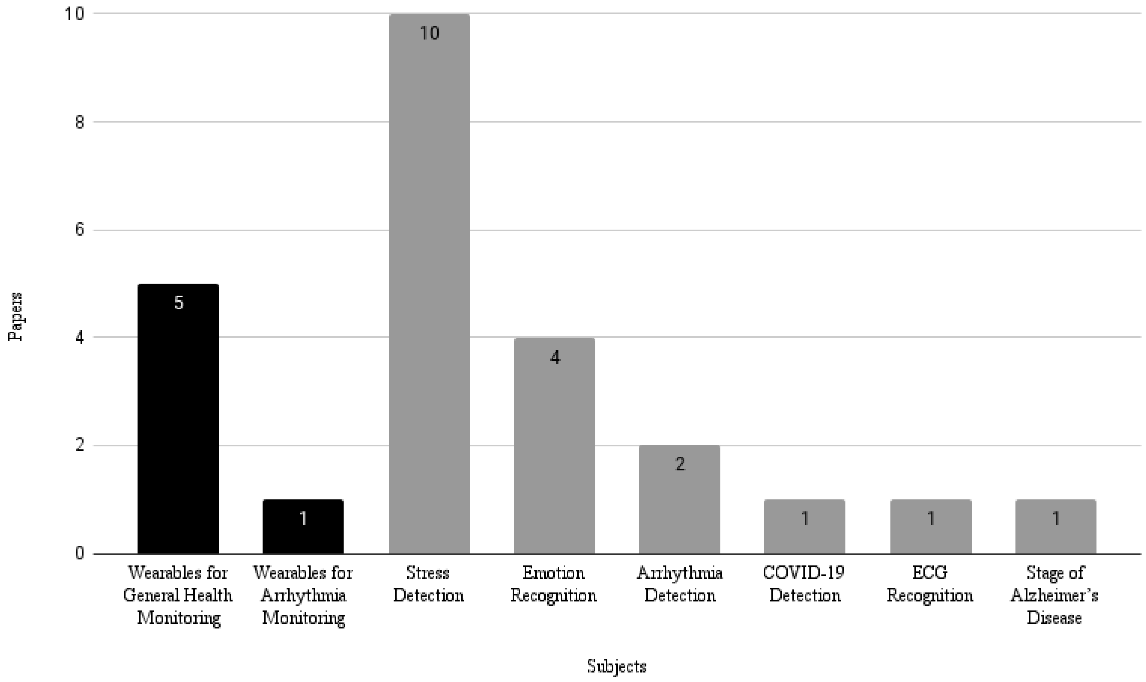
Among the topics covered by the works, we can mention arrhythmia detection, disease detection, movement detection, stress detection, and emotional state, a more subjective theme. Nevertheless, we verified a gap in current work on psychological state and the effect of stress in patients with few research and papers before 2013. Few studies analyze the effects of stress on the human body through wearable sensors, as the study carried out by Schmidt et al. (2018).

As shown in Table 2, we can separate the analyzed works into two major classes. We use the term classes to classify the kind of paper reviewed in this context. In other words, the papers have two different groups, as previously stated. We have six papers focused on health monitoring using wearable sensors in the first group. Five papers relate to general health monitoring, and one about arrhythmia monitoring. Fourteen works are in the second group focusing on the artificial intelligence applied to physiological signals patterns detection. There are five papers on Stress Detection and four papers on Emotion Recognition within these classifications. Among the other classifications analyzed were Arrhythmia Detection two papers and COVID-19 Detection, ECG Detection, and Stage of Alzheimer's Disease with just one work each.

We have used two different paper classes to represent different parts of the project. The first represents continuous patient monitoring without automatic detection or intervention. We describe the use of the term in the context of this work to clarify any doubts. We use this terminology to refer to the monitoring of the patient without pauses during long periods. It includes monitoring outside the hospital environment. The second group of papers refers to artificial intelligence techniques applied to physiological signals. We must highlight that separating the classes of papers during the state-of-the-art review was necessary. We needed to do this because we did not find any work with both themes (Figure 15).

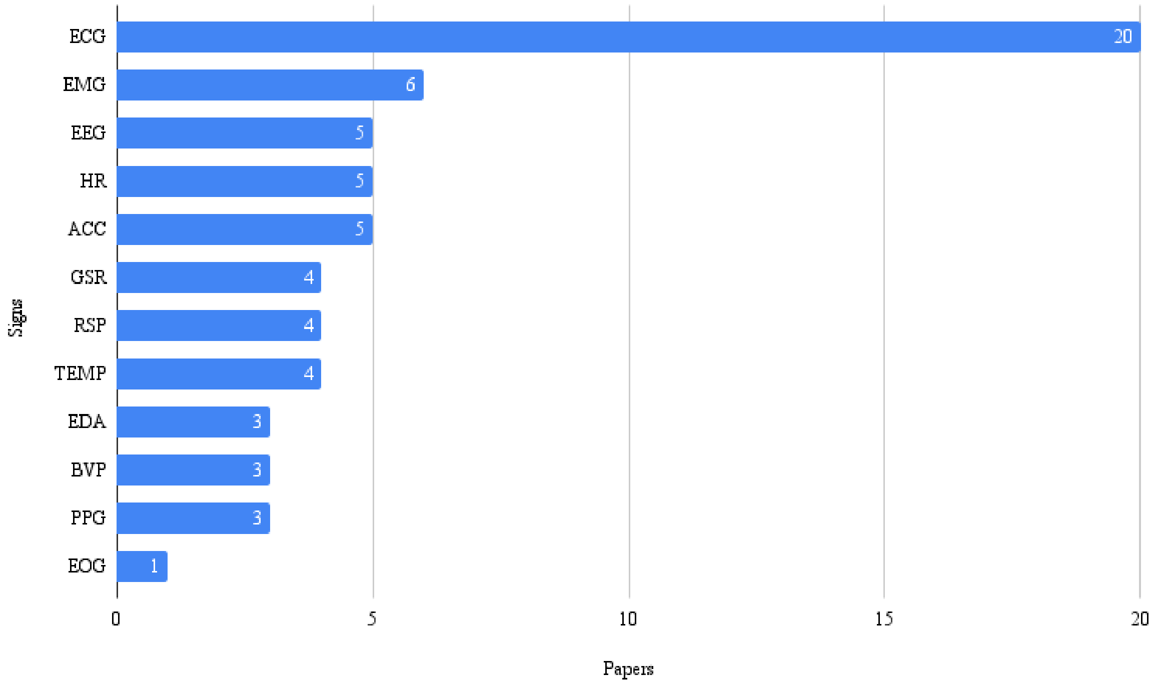
We found that the most used signals are the ECG, present in 80% of the papers, and the EMG in 24%. The use of EEG, HR, and ACC 25%. Other widely used (20%) signals are GSR, RSP, and TEMP (Figure 16). With the study of related works, we have verified that the most used method for predicting patterns in physiological signals is Machine Learning models, with 32% of all the reviewed papers. It is possible to check a tendency to use the Random Forest algorithm as an efficient classifier. These models have shown good results regardless of the signals. We also verified a great use of Deep Learning as an alternative. The authors indicate several ways to use Deep Learning networks but always focus on large datasets with a wide variety of participants. For this reason, we believe that the use of Machine Learning is present in most examples.

Figure 15 – Subjects Distribution in the Reviewed Papers



Source: Elaborated by the Author

Figure 16 – Signals Distribution in the Reviewed Papers



Source: Elaborated by the Author

Consonant with the papers reviewed, it is possible to observe that the study related to ML models in physiological signals shows promising results. The works indicate wearable sensors as a reliable method for physiological signal acquisition. Following this line and gathering both systems, we intend to develop an integrated architecture mixing wearable sensors and Machine Learning models. The architecture has a complete application, making the acquisition, processing, and, in the sequence, the diagnostic with the patterns predicted previously.

We could observe a gap in works that reconcile both topics analyzed during the state-of-the-art review. Many works address the continuous monitoring of patients, but within not flexible architectures, which do not allow the replacement of devices. On the other hand, we review many papers that approach signal detection using artificial intelligence. The research opportunity we have observed is stress detection using artificial intelligence through a flexible, multi-signal data acquisition architecture through continuous patient monitoring.

The papers analyzed demonstrated a research opportunity that integrates both themes, continuous monitoring of physiological signals with artificial intelligence for patterns detection in physiological signals. All works focus on data acquisition and storage or aim to extract patterns of data already acquired. Consequently, none of the analyzed papers have a real-time detection system applied to continuous monitoring. We used the terminology real-time to refer to the signal monitoring as close as possible to the moment when the architecture measures the data. Thus, the architecture analyzes the data obtained as soon as possible and not just at a specific time of day.

In the following chapter (chapter 4), we present the proposed architecture and its functioning. We present the architecture at the conceptual level and then the architecture at the design level to explain the objectives of the architecture, how it works and what the main features are.

4 ATHENA I

This chapter presents the architecture at the conceptual and design level. We also describe the initial choices of devices made for the prototype implementation. We designed the proposed architecture to use as few devices as possible. However, always consider the possibility of integrating more devices into the ecosystem, even distributing tasks across different devices. The architecture can operate autonomously, with only an acquisition device and a processing and registration device. However, we designed it to reconcile the approach in cloud computing. The following sections present an overview of architecture (4.1) at the conceptual level and level of design, as well as a brief explanation of the choices made in the project (4.2).

4.1 Overview

This work aims to develop an architecture that can be easily adapted to other applications, such as different wearable sensors, increasing/decreasing the number of signals and the number of channels treated. For example, suppose we desired to add a new wearable sensor to the acquisition ecosystem. In that case, we could distribute the applications to the data acquisition between Wearable sensors and Single-Board Computer. Considering that most wearable sensors currently in the market use Bluetooth for communication, it allows them to transmit their data to other devices wirelessly. For this reason, the communication between the wearable and the SBC takes place via Bluetooth communication. However, if we consider a network of SBCs, this communication could be over the network, centralizing the records and the detection into one other device.

The developed architecture aims to meet both the acquisition of physiological signals and the patterns detection in these signals. Another feature that this architecture allows, besides pattern detection, is the data acquisition and composition of a dataset to generate an artificial intelligence model that can detect anomalies in physiological signals. As seen, many works approach wearable sensors to monitor patients. As well, many research analyses use physiological signals to perform pattern detection. However, we have not found papers that reconcile both stages, acquisition and classification, all in real-time.

We divided the architecture into four main blocks, intending to distribute the functionalities among the blocks, thus allowing greater flexibility. We conceived the developed architecture following the guidelines of SAP AG (2007) regarding the notation Technical Architecture Modeling (TAM), defined as a standard to represent the definition of architectures. The TAM reconciles the Unified Modeling Language (UML)[1] standards, for design, and Fundamental Modeling Concepts (FCM)[2], at the conceptual level (KNOPFEL, 2007).

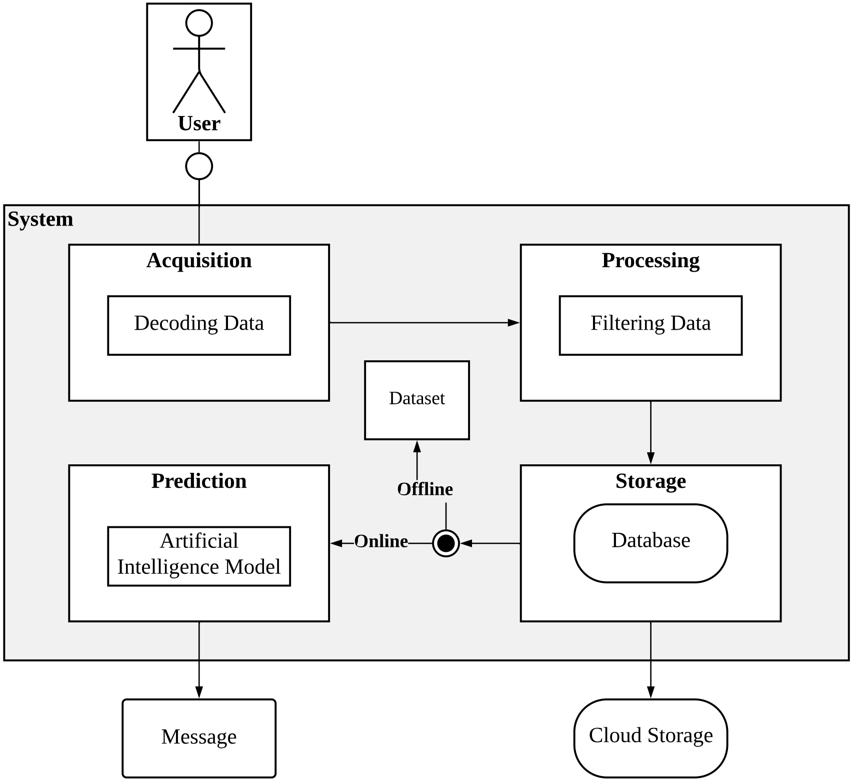
Firstly, we have conceived the architecture at a conceptual level, which follows the guide-

[1] https://www.uml.org/#UML2.0

[2] http://www.fmc-modeling.org

lines present in the FCM, separating the functions of the architecture into blocks. With a primary focus on flexibility, after defining the bases of the architecture, we planned the design level. In Figure 17, we can see the proposed architecture representation at the conceptual level. This description consists of four blocks and their integration flow.

Figure 17 – Architecture Conceptual Level

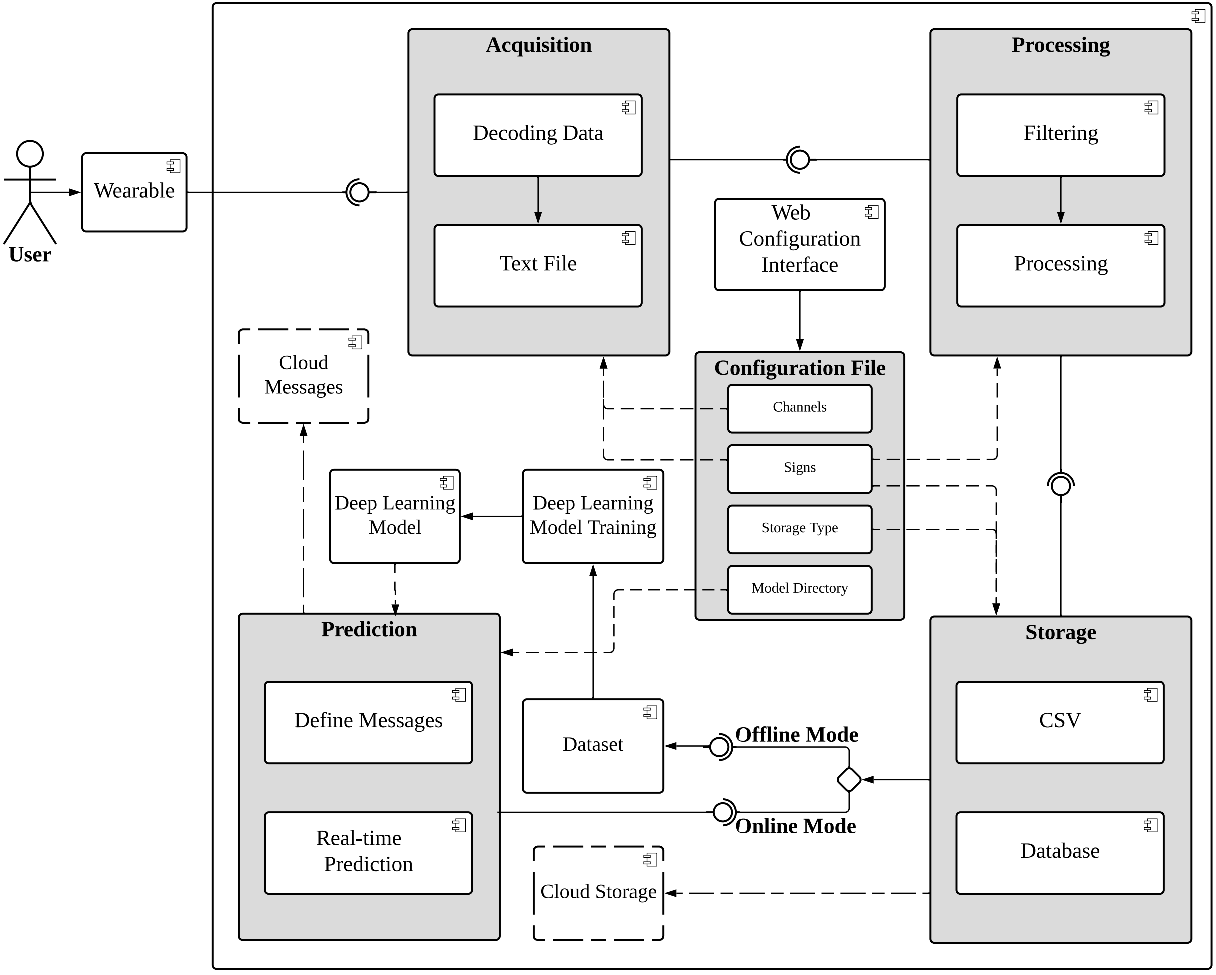


Source: Elaborated by the Author

The process starts with the user interacting with the architecture. In terms of the architecture, the first step is the acquisition block, responsible for performing the acquisition and decoding of the data acquired from the user. The next is the processing block, responsible for filtering and processing the signals. After this step, the architecture passes the data to the storage block, where the architecture stores the data in the database. Nevertheless, also there is the option of implementing a cloud storage option. At this point, there are two possible operations for the architecture. The first mode is called Offline Mode, in which the architecture only stores the data for the composition of a dataset. The second option is using the Online Mode, in which the architecture passes the data to the fourth block. The classification block performs the detection in real-time, based on a previously trained artificial intelligence model. If the architecture detects any signal change based on the classification, the architecture may send a message describing this situation.

Advancing in the designing of architecture, we move to the design level (Figure 18) according to UML guidelines, which describes the blocks in more detail and the components responsible for interacting with the architecture are presented more clearly. As demonstrated in architecture at the conceptual level, the architecture starts with the user. The first component of the architecture, and the only one with which the user has contact, is the wearable sensor, which measures the data. The wearable sensor is part of the acquisition block, the other part of this block is the decoding data. The acquisition block only performs operations related to the wearable sensor because if we change the wearable sensor for another device, it is necessary to adapt only this block. Because the decoding happens in the acquisition block, this process is specific for each wearable sensor.

Figure 18 – Architecture Design Level



Source: Elaborated by the Author

The wearable sensor measures the physiological signals and sends them via Bluetooth to the SBC. Bluetooth was chosen as the primary form of communication because most studied wearable sensors support Bluetooth communication. After the SBC receives the data, they decode it for signal extraction and store it in text files to communicate with the next block. The next block is the processing block, as previously indicated, performing the filtering and processing of data through specific libraries for this purpose. The processing step is essential for the architecture to remove noise and obtain the signal in the desired format. In the third block, the architecture store the data based on information defined by the user. The storage can occur in two ways. First, as soon as the architecture reads the text file, it stores the data without filtering or processing directly in comma-separated values (CSV) file. Then, the architecture store the processed data in a database.

The architecture has two possibilities of operation, the Offline Mode and the Online Mode. The first refers to the mode used for the composition of a dataset. This option is valuable if the user does not have a dataset or wants to use the architecture to compose the own dataset following some specific protocol. After the composition of the dataset, it is necessary to perform the model training to supply the fourth block. The second available operating mode is the Online Mode. The architecture must have the pattern detection model previously loaded for this model to work. In this way, the data stored by the storage block is sent to the classification block and analyzed; if the architecture considers the signal as altered, it sends a message, but if it evaluates the signal as regular, the architecture continues to operate normally. However, there is the possibility to configure a message in a normal state.

The architecture can send a message to another device via the socket or another device via the cloud. A new element presented in architecture at the design level is the configuration file, edited through a web interface. The principal function of the configuration file is to define the most basic parameters of the architecture. The possibilities are the operating mode, number of channels, signals, type of storage, and DL model file. This element's principal function is to allow for system flexibility. In the sequence, we make a brief description of the choices made for the implementation of the architecture prototype (chapter 5).

4.2 Specific View

In this section, we present an even closer view of architecture, describing the choices for the proposed architecture implementation. The proposed architecture consists of two principal devices. The first device is the wearable sensor BITalino (PLUX WIRELESS BIOSIGNALS, 2021), Model PsychoBIT, and the second device is the Raspberry Pi 4, Model B + (RASPBERRY PI FOUNDATION, 2021. The choice of BITalino as the wearable sensor used in the architecture is because it is an open-source wearable sensor with low-cost and easy adaptability. As an aspect of adaptability that we can mention, the company that develops this platform provides several kits. There are kits for development, expansion, or commercial use among the possibilities.

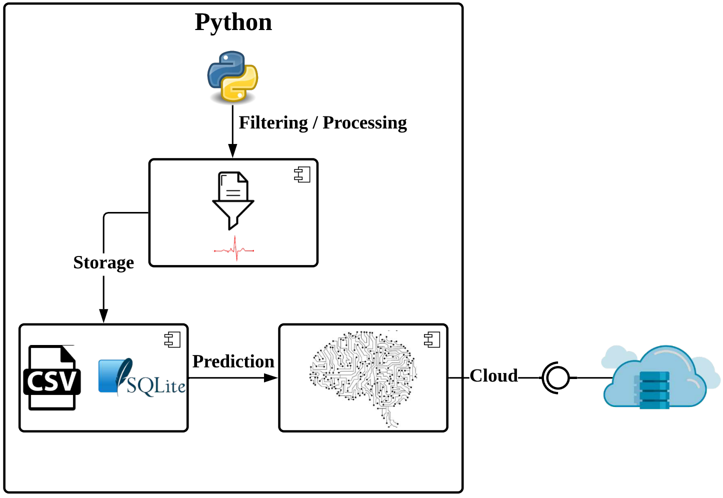
In addition to the existing flexibility, the company provides many sensors. As we chose the model PsychoBIT, which focused on measuring psychological effects, the architecture provides ECG, EDA, and RSP.

The choice of Raspberry occurred due to the processing power of the board. In addition to the availability of Bluetooth and Wi-Fi interfaces, requirements for implementation in a project that seeks flexibility. The processing power is essential considering the number of applications that the architecture executes internally within the SBC. However, if we expand the ecosystem and distribute the applications within a more significant architecture, we can reduce processing power in each board.

BITalino acts on the physiological signals acquisition arranged directly on the participant's body through electrodes. The architecture sends the data measured by BITalino to Raspberry via Bluetooth. When Raspberry receives the signals, Raspberry processes the data and stores it. The subsequent steps are filtering and detection.

Due to the characteristics of the Raspberry Pi, we have developed the codes in two different languages (Figure 19), the first in C++[3], due to the need to use Bluetooth for communication, since libraries with support for Bluetooth existing at Raspberry are focused on C++. The second programming language used was the Python 3[4], the reason for choosing this language is motivated by the libraries for data manipulation and artificial intelligence.

Figure 19 – Example of Raspberry Architecture Topology



Source: Elaborated by the Author

IAs seen previously, the programming languages used follow a cycle of operation, which starts with C++ performing communication with the Bluetooth interface. After decoding the data, Python is responsible for performing all other steps in sequence.

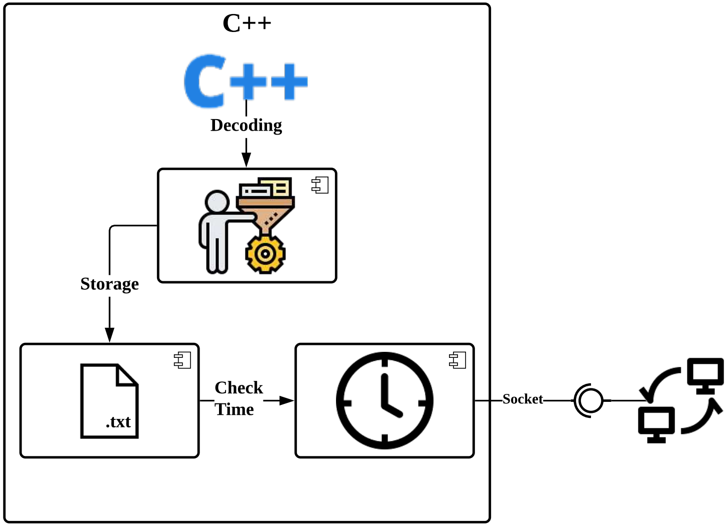
In Figure 20, the code presented in a simple schematic form belongs to C++, which has as fundamental functions the decoding of the data received via Bluetooth and then decoding them. The data after decoded needs to be stored so that the process is as dynamic as possible. To improve the dynamism of the architecture, we decided to record the data in text files. This storage in the text files happens with each cycle of execution of the code to avoid data loss. After each cycle of program execution, the code checks the time after receiving the data. Moreover, if a minute has passed, the architecture sends a notification via socket to the next block.

From the processing block, we have written in the code Python. We have chosen Python for the rest of the development because this language has the more accessible manipulating

[3] https://docs.microsoft.com/en-us/cpp/

[4] https://docs.python.org/3/

Figure 20 – Example of the C++ Code



Source: Elaborated by the Author

data as the main feature. It is necessary to carry out several steps such as data range selection and application of the filter, like low-pass, high-pass, band-pass filters, and even digital filters. Furthermore, Python has several libraries dedicated to filtering, data science, and artificial intelligence.

The code in Python has an operation cycle that includes the other three blocks, which in the short act of filtering and processing, which have support by the BioSPPY[5] (CARREIRAS et al., 2015) physiological signal processing library. Next, we have storage, carried out through databases and CSV files, with support from libraries such as Pandas[6] and SQLite3[7]. Finally, we have the classification block, which has the support of Keras[8], a library dedicated to artificial intelligence. The last function of Python within the architecture is to prepare the sending of the message. The architecture can send this message via socket to another device over a local network or cloud.

In the next chapter (Chapter 5, we detail the implementation of the prototype, following the architecture proposed in this chapter (Chapter 4). We describe the choices made and the difficulties encountered during the development. In addition, we relate the necessary changes for the architecture if compared with the previously idealized.

[5] https://github.com/PIA-Group/BioSPPy
[6] https://pandas.pydata.org
[7] https://www.sqlite.org/index.html
[8] https://keras.io

5 PROTOTYPE

This Chapter describes the prototype implementation process based on the architecture proposed in Chapter 4. As previously described by Wilson and Laing (2018), an integrated wearable sensor architecture must have a single-board computer, power source, acquisition software, and storage location. Hence, the proposal makes the architecture as flexible as possible, allowing the maximum of steps with the least number of devices. On the other hand, there is no intention to limit the architecture, allowing anyone to expand the architecture.

We have designed the architecture in blocks. We explain the decisions and explain what can change in the architecture. In the description, we describe according to the code duty cycle, first describing the acquisition block (section 5.2), then the section 5.3, the processing and filtering block, and the section 5.4, block that stores the data. Lastly the section 5.5, responsible for the patterns classification. The classification section describes the Deep Learning training model developed. However, we carry out the training model and the fourth block implementation as continuity of this work in the next steps.

5.1 Configuration

The configuration file is an element to be highlighted, which directly interferes with all blocks. The configuration file has the function of providing a simple way to configure the parameters of the architecture. Among the parameters that the file can configure are the physiological signals that the architecture process and the wearable sensor's channels for acquisition. They are being able also to define which wearable sensor we are using. In addition, we can also define information such as the MAC of the Bluetooth interface, the forms in which the architecture stores the data, and define the ML model provided to the architecture.

The ATHENA I has this configuration in a configuration file inside the main point of the architecture. In addition to the points listed, this configuration file defines which architecture blocks are active and if any are disabled. Another possible configuration is defining the storage servers used, whether internal or external to the collection device.

5.2 Acquisition Block

This block is responsible for establishing Bluetooth communication between the wearable and the SCB. This block also must decode the information received and send it to the next block.

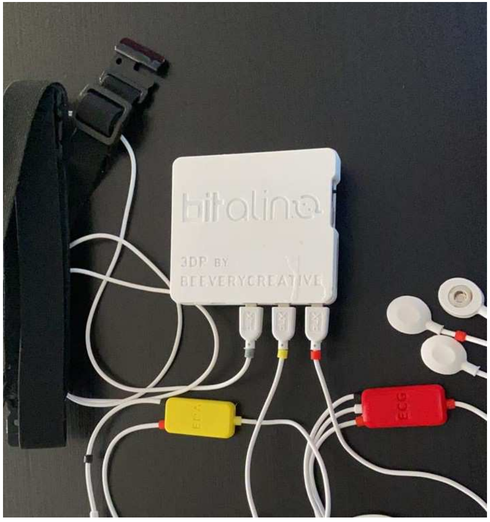
As previously described, this code is all written in C++ due to the need to use Bluetooth libraries, whose performance is better on the Raspberry Pi. With support from the BITalino library for C++, developed by Hugo Silva and Carlos Azevedo[3], made available by the manufacturer (PLUX WIRELESS BIOSIGNALS, 2021), it was possible to adapt the code that would

[3] https://github.com/BITalinoWorld/cpp-api

allow communication between SCB and BITalino through Bluetooth and to decode the received data. It is essential to note that, as previously informed, this architecture focuses on providing an easily adaptable architecture. For this reason, the main function of the code can be used almost entirely for other wearable sensors, being necessary only to change the libraries that perform the decoding.

Figure 21 shows the BITalino used in the prototyping of the architecture. The BITalino used in the architecture is the PsychoBIT model. The model used has sensors to measure signals such as electrocardiogram, electrodermal activity, and respiratory pattern. We used the signals from these sensors in the development of the prototype.

Figure 21 – Wearable BITalino PsychoBIT



Source: Elaborated by the Author

The developed code needs parameters for the program call, the participant's name and surname, and information that the architecture uses throughout the process to relate the data to the individual. At the beginning of the code, we define the most relevant parameters that can change according to the adjustment. Among the parameters is the selection of the signals the architecture acquires.

In this implementation, the architecture measures the ECG, EDA data, and RSP. In addition to being necessary to define the channels that the wearable sensor makes the acquisition, as an example of the configuration used $\{0, 1, 2\}$. Another necessary adjustment is to define the acquisition rate that we want. The acquisition rate is essential because it defines the number of data obtained per second of the acquisition. Therefore, the more data, the more precision.

Regarding the SBC used in the architecture prototype, we have the Raspberry Pi 4, chosen for its cost-effectiveness and processing power. Figure 22 shows the Raspberry used for the development of the prototype inside the case used for protection and cooling.

Finally, concerning the storage file, it is possible to define the directory in which the architecture stores the data. Furthermore, it is possible to change the IP address and port that the block sends to another about the socket. In other words, it is possible to define another device

Figure 22 – Single-Board Computer Raspberry Pi 4

Source: Elaborated by the Author

to acquire and perform the other blocks' tasks. We distributed blocks across other devices. All devices need to be on the same network. Then, through the configuration file, we define the addresses of each device on the network and its respective block.

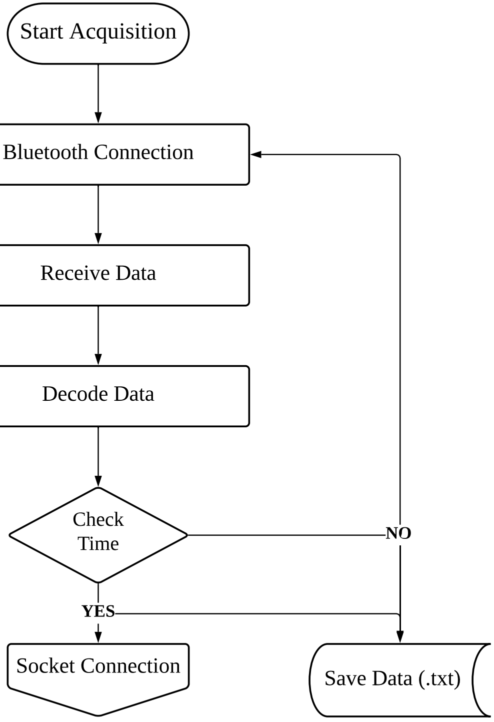
The same device acts for acquisition and processing as in the implemented architecture. We defined that the IP $127.0.0.1$ and port 18000. Before starting the acquisition, we must pair the wearable sensor with the Raspberry. For this, we can use the command $bluetoothctl$ to access the Bluetooth settings and then pair with the MAC address of the wearable sensor with the command $pair\ xx : xx : xx : xx : xx : xx$. However, if we do not know the MAC address, we can scan nearby devices to find BITalino, using the $scan\ on$ command.

Figure 23 illustrates the operation cycle of the acquisition block of the architecture. The process starts when anyone starts the acquisition through the web page that calls the acquisition code, passing the participant's name as a parameter. In the sequence, the program establishes communication between Raspberry with BITalino, using the MAC address of the Bluetooth card of BITalino to connect. Then, the parameters mentioned previously are defined, given the rate of signal acquisition and which channels acquire.

In the next step, the Raspberry receives the data in frames with a maximum of 8 bytes. The BITalino libraries support decoding the data correctly, as each channel has a different range of possible values. When decoding the data, the code needs to check the time and verify if a minute has passed since started the acquisition process. If it is still within the same minute, the program opens the text file with the participant's name, date, and time acquisition, stores the acquired data, and returns to the Bluetooth connection to wait for another frame. The cycle happens successively until one minute has elapsed since the reference minute. The reference minute receives the value when starting the acquisition or each time that minute pass, so the reference minute is updated.

With an acquisition rate of 1 kHz, this cycle occurs 60,000 times per minute. We chose to store the data in text files because the other option would be to send the data via socket. However, the process of sending the socket was slow because the socket was unable to respond

Figure 23 – Acquisition Flowchart



Source: Elaborated by the Author

in the necessary time, less than 0.001 seconds. Therefore, we decided to implement the socket connection to inform the participant's name, date, and reference minute; the next block must read the text file each time. In this way, the other application can check if the read file is the correct one. The files are divided by minute so that the volume of data acquired in this period can be processed by the second block and analyzed by the Machine Learning model present in the fourth block. During development, we found that a smaller volume of data may be insufficient for processing and analysis.

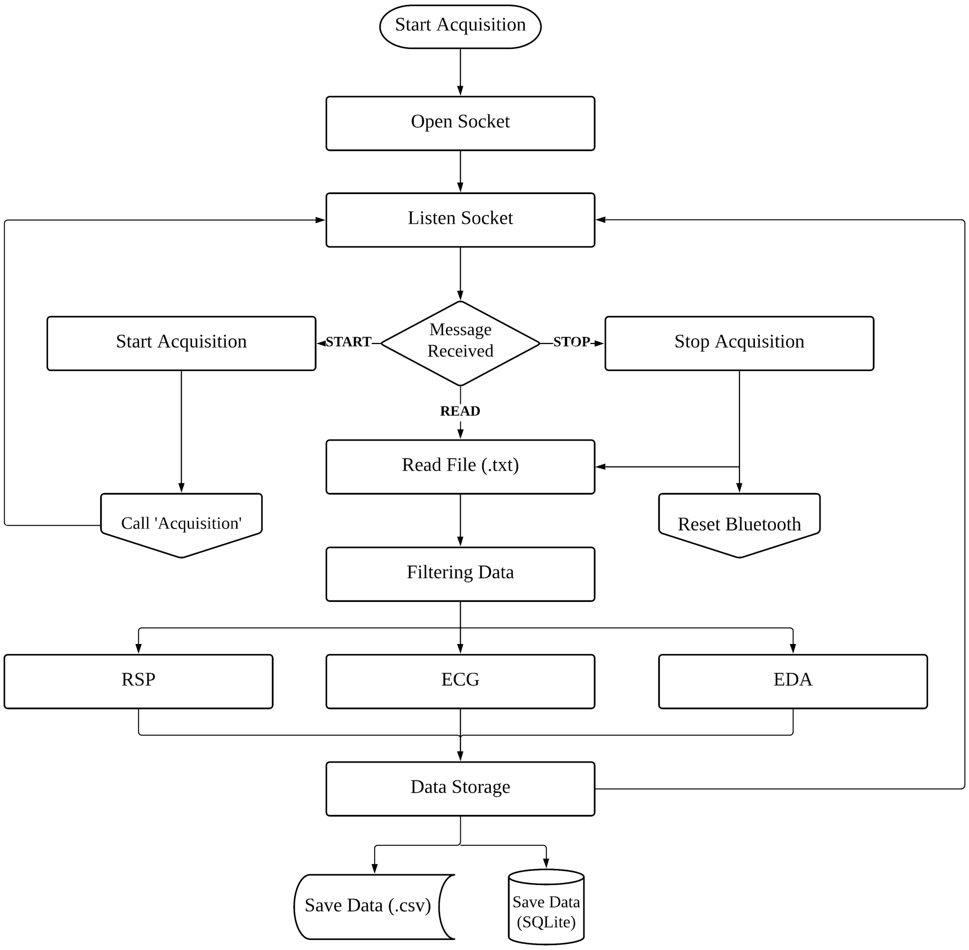
The architecture opens the communication socket and sends a UDP packet when it reaches the reference minute. Following the code, the reference value changes, and the file that stores the data also changes. This process occurs successively until the wearable sensor interrupts the communication. The acquisition program sends a packet notifying the processing block when interrupting the communication.

We added the minimum possible number of operations to make the architecture as efficient as possible. We avoid adding unnecessary functions in the main acquisition loop to ensure this. Among the operations performed by the architecture during the execution of the loop while the reference minute remains the same as the real minute are receiving the data, decoding it, getting date and time values from the architecture, and then checking the time. If the reference time has not changed, the data is saved to the text file and returned to the zero point of the loop. If the reference time has changed, the code has a few more operations, such as a socket connection check. In case of the socket is closed, it is opened again for sending the packet. In addition, a new text file is created with a new reference value and in the same way as before. The data is stored in the text file and returns to the zero points of the loop.

5.3 Processing Block

When passing to the blocks developed in Python, the blocks are chained more clearly, based on packets received via socket from different applications. The primary choice of using a socket for this application is the transparency allowed between the communication of different devices in a network. Whether or not the same device, so thinking in a context with more devices on a network, it is acceptable to assume that there are different devices with different applications that need to communicate (STEVENS; FENNER; RUDOFF, 2003). It works so that the socket is always open for listening, ready to receive any notification, following a working cycle passing to the blocks. In Figure 24 we can check the operation cycle of the code for Offline Mode, used to perform the acquisition of the dataset. We decided to represent all blocks in the same flowchart to make it easier to understand how one block interacts with another.

Figure 24 – Offline Mode Flowchart

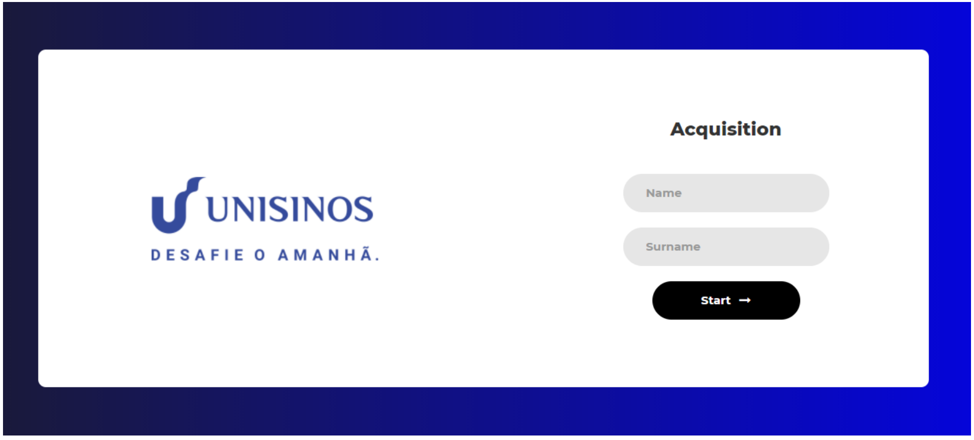


Source: Elaborated by the Author

The Python scripts are configured directly in Cron[4] to start with the Raspberry startup, so the Python codes remain running continuously in the background. As mentioned, the core of the architecture remains attentive to any request received via socket, so the first step is to open a server communication socket, with $localhost$ defined as the socket's IP address and, as previously informed, the port 18000, as a way to avoid conflict risks.

The architecture starts the monitoring and acquiring data after receiving a notification from the web page. We developed a web page to generate requests to the architecture, as can be seen in Figure 25. The user only needs to enter the participant's name and surname, and the web page sends a request. We developed the web page using ngnix[5] as the base, with the backend also developed in Python with only one Application Programming Interface (API) route for sending a socket via the network, notifying that the user wants the acquisition to start. The web page was developed in a separate application thinking about the context, which means that the architecture may have more wearable sensors and SBCs in the same ecosystem. For this reason, it is more convenient that the application that initiates the acquisition is separate.

Figure 25 – Web Page Developed to Acquisition Block



Source: Elaborated by the Author

Upon receiving the requisition, the architecture checks the received package. Suppose the information received is included in the request to start the acquisition. In that case, the architecture calls the acquisition program, passing as a parameter the name and surname of the participant. At the end of the program, the code cycle returns to the waiting point for the next package. The next expected package is with instructions to read the text file.

After reading the data, they must be filtered and processed. This step extracts the relevant data to make this process simpler and more efficient we are using the BioSPPY[6] for this process. The BioSPPY (CARREIRAS et al., 2015) is a library dedicated precisely to the physiological signals processing, for various biosignals, like ECG, EDA, RSP, EEG, EMG, and PPG, whose

[4] https://man7.org/linux/man-pages/man5/crontab.5.html
[5] https://nginx.org/en/docs/
[6] https://github.com/PIA-Group/BioSPPy

main characteristic is the reduction of noise, distortions' correction, through frequency analysis, and, finally, the extraction of information.

Analyzing the procedure performed with the BioSPPy library, we verified how the library applies the filters in each sign. The library utilizes a filter to remove noise and interference, starting with the ECG. The applied filter is a Finite Impulse Response (FIR) digital filter of the bandpass type, and the order is defined based on the sampling frequency. We are 30% of the sampling, as we have used a frequency of 1 kHz, the FIR filter of order 300. As a bandpass filter, it is essential to define the limits. The lower limit is 3 Hz, and the upper limit is 45 Hz.

Then, for the extraction of ECG data, the Hamilton (2002) method is used, the first step of which is the application of a Butterworth low-pass filter. The filter has order four and a cutoff frequency of 25 Hz. The next step is to apply a high-pass filter, Butterworth of order 4. The library defines a cutoff frequency of 3 Hz. Lastly, a derivative function obtains a moving average of 80 ms. Finally, peak detection and QRS complex detection are applied.

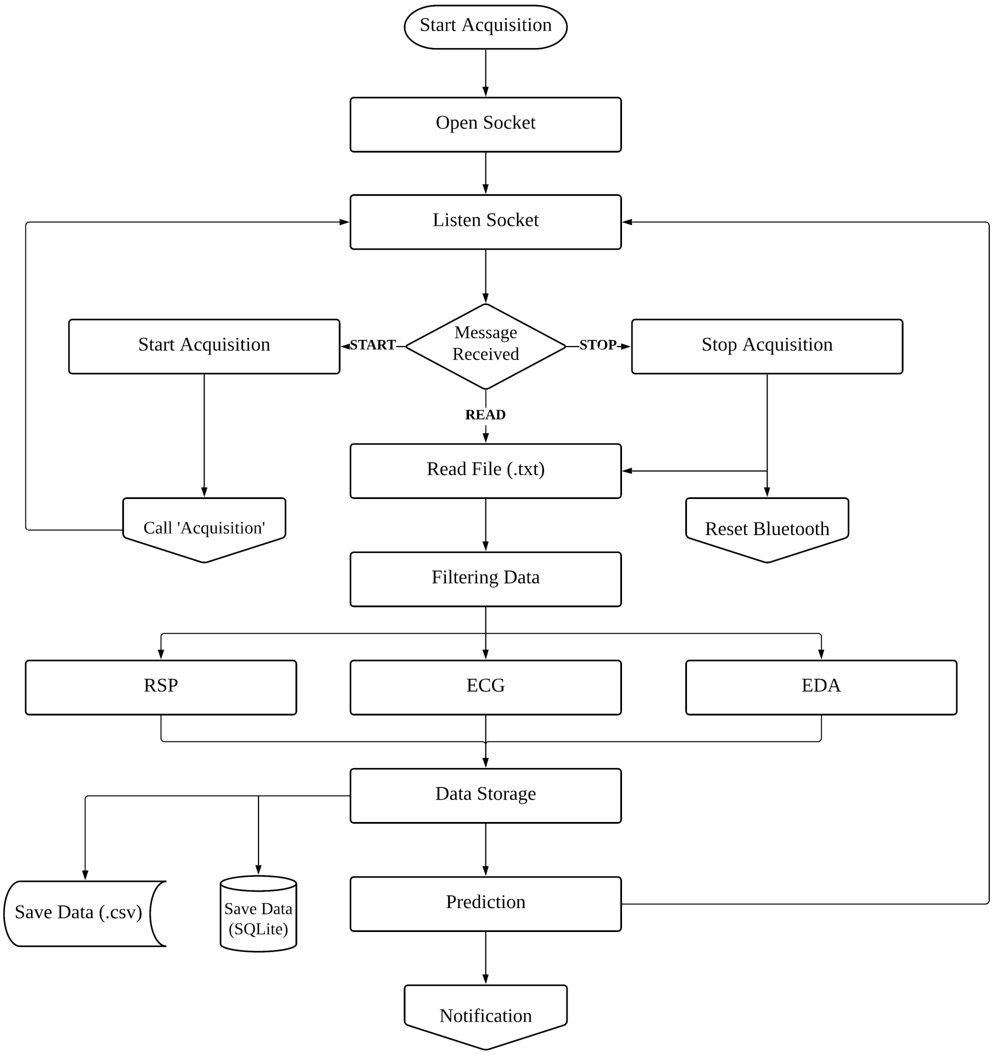
The first step is to apply a Butterworth low-pass filter of order four, with a cutoff frequency of five Hz when it comes to the EDA filter. Furthermore, a moving average function of N terms is applied to smooth out the distortions. Finally, the library applies differential functions with an analysis in frequency, and it extracts the intrinsic values through zeros of the transfer functions. Regarding the processing of the RSP, it is relatively more superficial than the two previous signals. A second-order Butterworth bandpass filter is applied, with cutoff frequencies set to 0.1 Hz and 0.35 Hz. Then, a zero-crossings function is utilized, which checks the points at which the signal intercepts the axis. Finally, a derivative function and smoothing with a moving average are applied.

After filtering and processing, the architecture must save. For this, we have used the Python library for SQLite3. The details of how the storage happens are in the section 5.4. After finishing the data storage, if the architecture is operating on Offline Mode, the code returns to the initial block to wait for the receipt of another packet via socket. So that this cycle repeats successively until the acquisition ends. However, if the architecture has the dataset ready and the artificial intelligence model is loaded, the next block performs pattern detection.

Finally, after completing the acquisition and ending the communication between Raspberry and BITalino, the code in Python receives a package informing the end of the communication. Upon receiving this information, the code terminates the communication with the acquisition program and restarts the Bluetooth interface to clear any stuck record, as was verified in some tests. During the tests, we verified problems with the next connection. It happened when the last acquisition stopped restarting the services. The verified problems are loss of communication, not pair, and data distortion. In parallel to the restart of the Bluetooth reset, the filtering, processing, storage, and detection process is carried out one last time.

To clarify how the code works in Online Mode, Figure 26 shows the flowchart of the three blocks in operation. This flowchart differs mainly from the one shown in Figure 24 by the presence of the classification function, which sends a notification via socket or a message to the cloud periodically. Then returns to listen to the socket coming from the acquisition block. We describe the operation of the classification step in detail in section 5.5.

Figure 26 – Online Mode Flowchart



Source: Elaborated by the Author

5.4 Storage Block

Data storage is a fundamental step in any process because it is precisely the architecture stores data. That data is the main objective of the architecture. Due to this importance, there must be safe storage means, minimizing the risk of loss as much as possible. For this reason, the architecture has redundancy in the forms of storage. Bearing in mind that the blocks do not necessarily need to be on the same device, they can operate on different devices. For this reason, each must-have blocks part of the data. Therefore, we intend to provide the data in case of problems in any part of the architecture.

First, we can mention, even outside the storage block, the storage performed by the Acquisition block, which, in addition to serving as a way to obtain data from one service to the other, also serves as a backup. Text files take up little space concerning other storage media since they have no formatting. Besides, allowing quick access to the content and can be easily read by any program or application, being able to be considered universal (DALE; LEWIS, 2007). The data in these files are stored only raw, without any processing. All files have a header to guide the kind of data. The columns on the files have semicolons to separate them. The architecture stores the acquired signals (name, surname, ECG, EDA, RSP, and Time) on a new line. As already mentioned, for each change in the reference minute, a new file is created, resulting in a large volume of files.

The second storage medium used is the CSV files. These files are separated one by the participant. The architecture creates the files with the participant's name used for the requisition to start the acquisition. CSV files are easy to use for data science and a structured visualization of data through spreadsheet software. However, they have the disadvantage of a higher consumption for storage. This storage medium is optional and can be disabled if it is not necessary. The CSV file store only raw data, with no filtering nor processing like the text file.

Finally, we have the principal storage, which consists of storing the data in a database. For this process, we opted to use SQLite[7] because it is a lightweight database and developed for applications that need storage that does not take up so much space. Among other advantages that SQLite has is being compatible with several platforms, such as Linux, Windows, macOS, Android, and iOS. In addition to being a database that does not require any previous configurations and can even easily create new tables within the code that stores the data if they do not exist. The database stores the data as the principal storage medium; it saves both processed and raw data. For the processed data, as its temporality is different, the acquisition time is removed, but the time obtained through processing keep as a temporal reference.

5.5 Prediction Block

The classification block has two possible operating modes. The offline mode in which we use the acquired data to generate an AI model for classification. The second mode of operation is the online mode. The architecture loads the generated ML model, and the acquired data is analyzed in real-time to evaluate the classification of the data. The block always operates by acquiring and storing data. In this way, we can expand the dataset with each new data acquisition. It is possible to define where the artificial intelligence model will be loaded through the configuration file. The architecture will only operate in Offline Mode if no model is loaded.

[7] https://www.sqlite.org/index.html

5.5.1 Offline Mode

Prediction Block in offline mode uses the data acquired and stored in Storage Block to generate a Machine Learning model that will be later loaded into the architecture so that it can operate in online mode. The architecture acquires a dataset in offline mode, and the data is stored internally to generate a Machine Learning model. To generate the model, we apply a windowing process to all data from participants to emphasize the characteristics of the data. After this adjustment of the data, the data undergo a set of model tests to evaluate the algorithm with the most incredible accuracy. Among them are Decision Tree (DT), K-Nearest Neighbors (kNN), Random Forest (RF), Ada Boost (AB), and Multilayer perceptron (MLP).

5.5.2 Online Mode

We must have the ML model previously trained to use the architecture in online mode. The model is then loaded into the architecture to predict the data. The architecture analysis the acquired data every 30 seconds, evaluating whether the participant is in a stressed state or not. After processing and storage steps, the architecture sends the data to the classification block. We reshape the data into the matrix format used in the model training step.

The signals are analyzed every 30 seconds; however, the model classifies the values in periods of 300 ms. Consequently, for each period of 30 seconds, 300 classifications need to be analyzed to generate a single classification. Ranking results are stored for future reference in log files. In rating analysis, rating results need to be at least 70% rated overall for status to be considered stress. If classified as stress, the data sends an alert. This alert is under implementation work in future work.

In the next chapter (chapter 6), we present the scenarios developed to evaluate the architecture and carry out experiments. We present three scenarios designed to evaluate architectural stability. In sequence, we present the other scenarios. The first is to carry out data acquisition experiments through a stress induction protocol, and the second is a model training experiment using artificial intelligence. Finally, a new stress-inducing experiment for real-time acquisition detection validation.

6 EXPERIMENTS

This chapter deals with the experiments carried out with the developed prototype. We performed experiments with each block to guarantee the architecture stability. For this reason, the chapter has five sections that address each of the blocks reporting the problems found in the preliminary tests. Moreover, in the last section, we describe how we performed the practical experiment to acquire the dataset, detailing the followed protocol and presenting relevant information. All the results obtained in the experiments are analyzed and discussed in Chapter 7.

6.1 Scenario 01 - Evaluation of Communication Stability

As a first step after developing the code, it was necessary to test the stability of the communication. We have tested both communications: the Bluetooth communication, between BITalino and Raspberry, for the signals acquisition and the communication between applications. The stability of the Bluetooth communication is the most crucial point of the architecture because problems in the acquisition can generate several other issues in the following steps. From a flawed dataset, even an unsuccessful detection. With previous experiences in the acquisition of physiological signals (RODRIGUES et al., 2020), we already knew how fundamental it is that the data is complete, with no continuity failure in the acquisition, as this fact has a direct impact on the performance and accuracy of the application of ML techniques.

We must emphasize that data being in an integral state is essential for an adequate evaluation without loss of continuity. We have to remember that the physiological signals are continuous, and they have as a principal feature the level variation. The absence of this signal represents an abrupt change from one level to the next, without continuity, mischaracterizing the signal, or even deforming it. This absence may lead to a diagnostic error, depending on the amount of data lost in a period.

This section has two sub-sections that address each topic separately, initially focusing on the stability of the Bluetooth communication and the observed results. Furthermore, the sequence describes the socket communication and the changes needed to correct the data communication strategy between applications.

6.1.1 Bluetooth Communication

As previously reported, the Bluetooth communication must be stable for the architecture. The code responsible for the communication between both devices is as short as possible, performing only the necessary operations. Part of the objective is consuming the minimum processing possible to avoid slowdowns, thus causing data loss.

During development, we performed some tests and experiments that required some adjustments to the code, such as reducing the number of operations performed within the script. While testing, we found that using an acquisition rate of 1 kHz and the architecture trying to send the data acquired via socket caused slowness. This slowness caused less than 1000 records per second, thus presenting significant data loss. The evaluation took place by analyzing the volume of data generated per minute of acquisition. As all data are stored right after the acquisition, it is possible to validate the acquisition stability. We expect approximately 60,000 rows of data in the text file for every full minute.

We observed good communication stability during the Bluetooth communication experiment, making acquisitions of different periods and varying the distance between the devices. Acquisition periods ranged from 30 seconds acquisitions to 2-hour acquisitions, without communication loss or failure in data continuity. Among the tests carried out by varying the distance, the distance variation was 30 centimeters between both devices, up to a distance of 5 meters, with no walls in the way, only objects.

6.1.2 Socket Signaling

Another relevant point of the architecture is the communication between applications. After ensuring good communication between the devices, it is essential to ensure that the data is well synchronized between both applications so that there are no problems in processing and detection. The first approach tested was sockets communication for data transmission. However, even using a more dynamic protocol has not shown sufficiently good results. When the architecture used an acquisition rate higher than 100 Hz, the architecture began to present a slowdown in transferring the data from one application to the other, causing a significant loss of data.

We decided to store the data in a text file to solve the socket issue and make it available in a shared directory. Thus, it was enough for the application of any other block to consult these files. However, it was still necessary to ensure synchronicity between applications to allow real-time application of data. We used the existing socket structure to send a package according to the defined period as a solution. In the case developed, it occurs from minute to minute, according to the reference minute, as explained earlier in the section 5.2.

After the adjustments, the communication and access to the file became fluid. We verified a perfect synchronization of the communication without impacting the acquisition or the stability of the Bluetooth communication. It is relevant to highlight the option concerning the protocol used for Ethernet communication. We defined that the socket would use the UDP protocol. When implementing the TCP protocol, there was a slight slowness due to the waiting for confirmation of receipt of the packet by part of the Python application.

Another important reason for choosing UDP instead of TCP was because if the application in Python presents any problem, we do not lose the acquired data. It did not happen because the UDP sends a package, and it does not wait for the confirmation, not stopping the process.

in other words, the acquisition block would continue to get the data and store it in the text files. It allows certain independence of the block acquisition with the processing block. It is worth noting that we can use the acquisition block separately to collect raw data without processing without the need to use another application to start it.

6.2 Scenario 02 - Evaluation of Data Processing to Real-Time

This section covers the experiments carried out to test the data processing. We also evaluate the results obtained, validating the minimum amount of data necessary so that the filters and the processing performed can be applied.

6.2.1 BioSPPy

The tests related to processing sought to validate the plausibility of the processed data. However, the architecture must have minimal data to apply the BioSPPy library. During the tests, we have observed that with an acquisition rate of 1 kHz, it would be necessary to acquire at least 30 seconds of data, about 30,000 lines of raw data, to obtain a good analysis result.

We have decided to acquire data referring to at least one minute before applying any processing to guarantee a sufficient volume of data. Thus, the block is responsible for processing about 60,000 lines for each signal acquired, providing a reasonable volume of data. It allows the data to be processed satisfactorily concerning the context of the data. In this way, the data volume does not require much processing time to filter and process the data on the Single-Board Computer.

6.3 Scenario 03 - Evaluation of Storage

The third test scenario aimed to analyze data storage and its different forms. The block focuses on storing data in CSV files per participant and the database. Nevertheless, we also carry out a brief analysis of the storage process in a text file, evaluating performance. We also present some information about the volume of storage generated. However, we discuss this topic in the chapter 4, section 7.2.

6.3.1 Text File

Firstly, analyzing the data storage in text files, as can be concluded based on the two previous scenarios, satisfied the two primary purposes of its implementation. First, the text file provided an intermediate database between applications due to having a large volume of data acquired by one application and read by another. So because it is a light and unformatted file, its opening and reading are very dynamic.

The second point we opted for its use is a backup of the acquisition block. It proved to be a reliable way of storing the data, mainly thinking about a scenario in which the acquisition and processing blocks are on different devices, perhaps in a context with more devices.

As a negative aspect, we have a large volume of files generated by each acquisition. Keep in mind that each minute the block generates a new file. If the acquisition lasts 10 minutes, the process generates at least ten files. If we extrapolate the acquisition period to one hour, we have about 60 files. On the other hand, despite being a large volume of files, each file requires an average of 2.5 MB of storage.

6.3.2 Structured Data File

Regarding CSV files, there is little to be analyzed. The way the storage happens is quite similar to that used for storage in text files. However, the main difference that we have is that the data become structured due to the characteristic of this type of file, as the name indicates. Another aspect to be highlighted as a differential is that this file makes up the storage block. In theory, it can act on devices other than the one that performed the data acquisition.

CSV files are generated one per participant, with no segmentation of files by time. Like text files, CSV files have a date column to guide the temporality of the data. A standard CSV file with a 10-minute acquisition contains approximately 27.5 MB. It represents an increase of 2 MB if we compare it to the same data acquisition period in text files (comparative value of 24.8 MB on average for text files).

6.3.3 Database

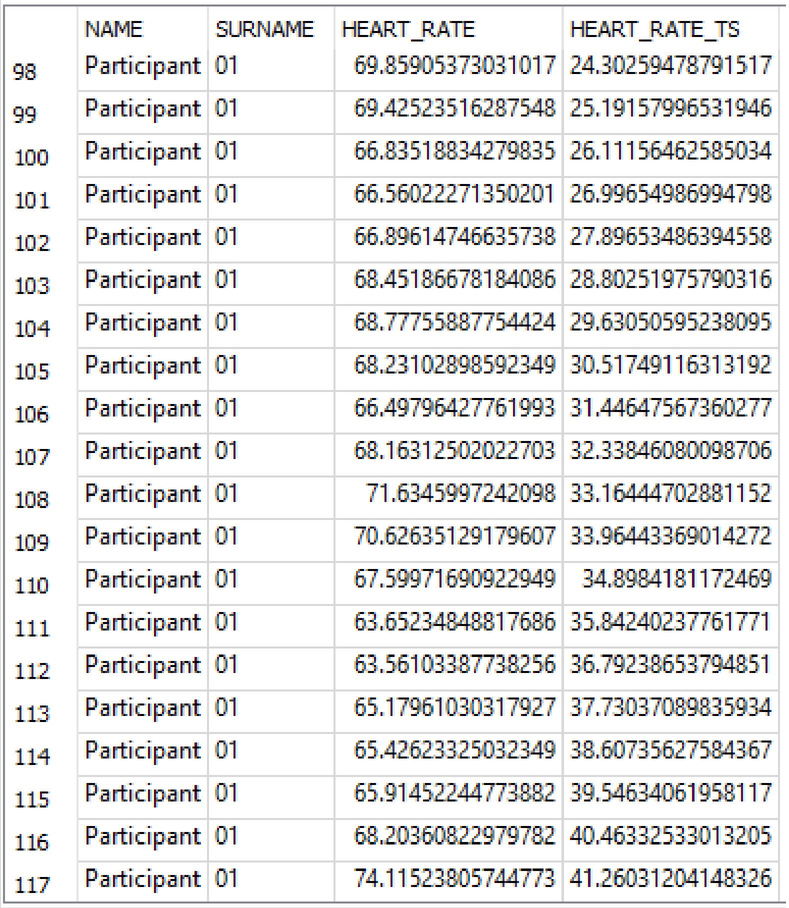
The database is the principal storage medium used by the architecture. All the data of all the participants are concentrated. The architecture store the data in raw and processed forms for each signal. For this reason, the database tends to occupy a larger volume on disk for the same period of a participant by other means. However, it has more tables than the other storage media.

The principal advantage of the database is the possibility of exporting the entire database to another device. Besides, we can use the data in any format without applying filters or processing. The data are ready for graphical analysis, analysis in the domain frequency, spectral analysis, or use on dashboards.

The architecture showed excellent stability in recording the data regarding efficiency and performance. The architecture recorded all the data in all the tests that we performed practically directly. The functions for creating the database and creating the tables also work. In cases where we have removed the database, the script created the database without problems.

During the collections carried out, both short collections, from 1 to 5 minutes, and longer acquisitions, from one to three hours. The data were stored correctly, without presenting data losses or discontinuities. The database also had no problems such as access denied or database corruption. To compare the disk space occupied by the database, when it acquired data over 10 minutes, it occupied approximately 55.5 MB, remembering that the database stored a much larger volume of data. In the context of the experiments, there were at least three more tables of data. We can check the database view in figure 27.

Figure 27 – Example of the Data Collected in the Database



	NAME	SURNAME	HEART_RATE	HEART_RATE_TS
98	Participant	01	69.85905373031017	24.30259478791517
99	Participant	01	69.42523516287548	25.19157996531946
100	Participant	01	66.83518834279835	26.11156462585034
101	Participant	01	66.56022271350201	26.99654986994798
102	Participant	01	66.89614746635738	27.89653486394558
103	Participant	01	68.45186678184086	28.80251975790316
104	Participant	01	68.77755887754424	29.63050595238095
105	Participant	01	68.23102898592349	30.51749116313192
106	Participant	01	66.49796427761993	31.44647567360277
107	Participant	01	68.16312502022703	32.33846080098706
108	Participant	01	71.6345997242098	33.16444702881152
109	Participant	01	70.62635129179607	33.96443369014272
110	Participant	01	67.59971690922949	34.8984181172469
111	Participant	01	63.65234848817686	35.84240237761771
112	Participant	01	63.56103387738256	36.79238653794851
113	Participant	01	65.17961030317927	37.73037089835934
114	Participant	01	65.42623325032349	38.60735627584367
115	Participant	01	65.91452244773882	39.54634061958117
116	Participant	01	68.20360822979782	40.46332533013205
117	Participant	01	74.11523805744773	41.26031204148326

Source: Elaborated by the Author

6.4 Scenario 04 - Data Acquisition Following Stress Induction Protocol

In this section, we present the dataset acquisition process in practice. To perform this experiment, we followed a stress induction protocol.

We have performed this practical experiment in a multi-discipline partnership with the Graduate Program in Psychology at the University of Vale do Rio dos Sinos, conducted by a master's student, Daiane Rocha de Oliveira. The ethics committee (CAAE number 40555420.0.0000.5344) approved the experiment to further studies regarding repetitive negative thoughts (RNT) on the body. The experiment had the objective of measuring the HRV at rest. However, in the experiments, we have included the sensors such as ECG, EDA, and RSP. Due to the pandemic scenario, we followed all safety protocols, and the procedure happened only with the participant and the psychologist in the room. Another aspect that we can highlight about the prototype is the test of the architecture's stability and to assess simplicity because guidance about the use of the prototype happened remotely regarding the use of the architecture and all procedures.

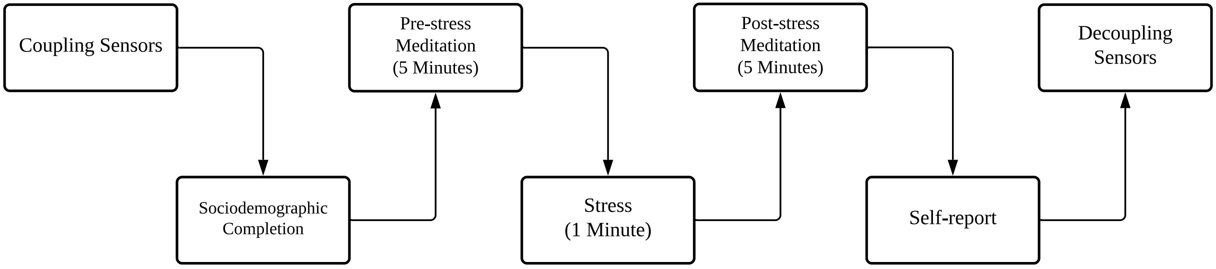
We intend to compose a proprietary dataset with the experiment. With the stress induction protocol, it is possible to obtain three classes of data: pre-stress, stress, and post-stress (meditation). This experiment was also part of the master's research of the psychologist Daiane Rocha de Oliveira. She aims to evaluate the heart rate variability in clinical and non-clinical participants (without a prior diagnosis) and verify the behavior of this signal throughout a process that leaves the normal state, with orientation to concentrate, going to a moment of stress, and then returning to the normal state, trying to remove the stress.

6.4.1 Protocol

The protocol developed for this experiment starts from the premise of inducing stress based on negative thoughts. Initially trying to prove this hypothesis, we performed an initial experiment using the commercial wearable sensor Polar H10 to detect the change of state regular to stress based on HRV. The initial results confirmed the hypotheses raised, making it possible to verify the difference in the R-R intervals when under stress.

With the hypothesis of the protocol confirmed, we performed the experiments with a larger volume of participants. The protocol seeks to assess stress induction in clinical and non-clinical participants, comparing the body's responses to the effect of stress. We can see Figure 28 the steps of the experiment carried out to acquire the data for the compilation of the dataset. We have considered clinical participants who have some psychological follow-up previously diagnosed, whereas non-clinical participants who have nothing diagnosed. We have not made any previous tests to assess whether non-clinical participants have something to be diagnosed.

Figure 28 – Stress Induction Protocol Flowchart

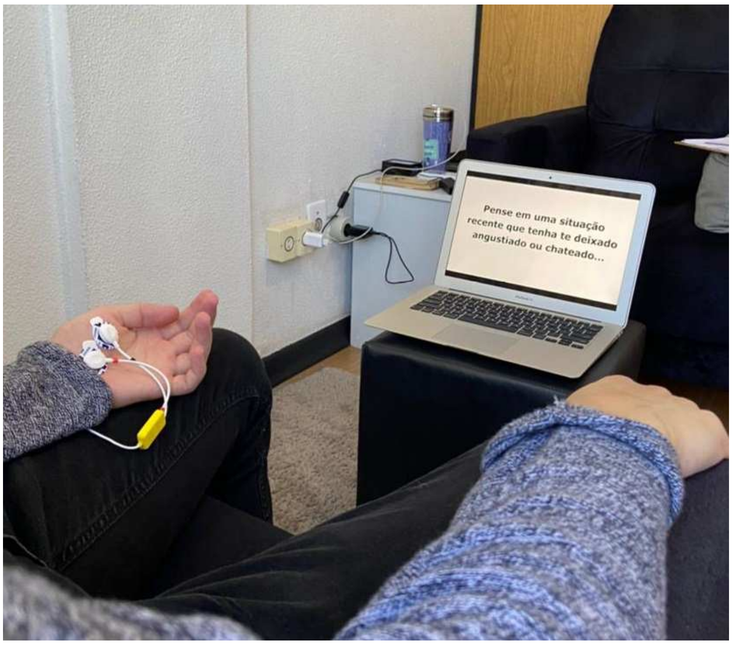

Coupling Sensors
Sociodemographic Completion
Pre-stress Meditation (5 Minutes)
Stress (1 Minute)
Post-stress Meditation (5 Minutes)
Self-report
Decoupling Sensors

Source: Elaborated by the Author

The first stage of the protocol is the coupling of the wearable sensor. We decided to place the electrodes at the beginning of the experiment. This way, the participants would get used to using the sensors. We intended to reduce the impact of the bother of the sensors during the acquisition. In the next step, participants filled in sociodemographic data and relevant data for inclusion in the model as the social context and physiological information tend to impact the physiological stress response.

After completing the sociodemographic data, the pre-stress meditation process begins, in which the participant remains looking at a white screen, oriented to try not to think about anything specific. The participant remains that way for five minutes. After completing the five pre-stress minutes, the white screen changes to a screen with the phrase "Think about a recent situation that has made you distressed or upset ..." (Figure 29). In this stage, the induction of stress occurs through negative thinking. The participant spends 1 minute thinking about the current situation that has made him go through some unpleasant moments.

Figure 29 – Photo with part of the experiment, participant looking at the laptop screen with the orientation

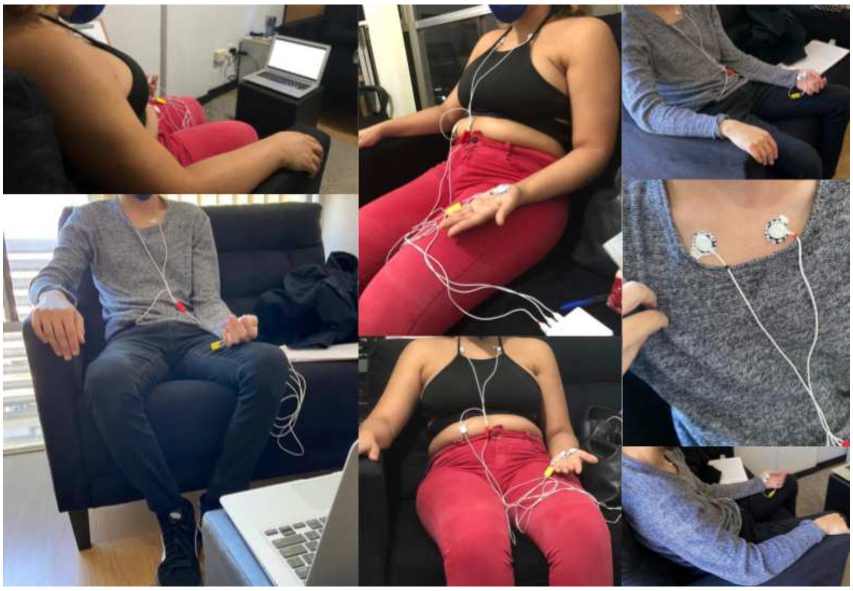


Source: Elaborated by the Author

After completing this minute of stress induction, the screen returns to a white screen, in which the participant again tries to ward off any thinking, constituting a post-stress stage. In this stage, as in pre-stress, it occurs for 5 minutes to return to a neutral state. When completing the period time of the post-stress stage, the participant performs a self-report. He explains what he thought and what was the feeling he felt during the process. This step also is relevant because it can explain possible differences in data for the same classifier during a stage.

Moreover, it may show differences in the groups analyzed. Finally, we removed the sensors, and thanks for participating in the experiment and research. It can be seen in Figure 30 some photos taken during an experiment with the participants.

Figure 30 – Photos of the participants during the stress induction experiment with the sensors placed



Source: Elaborated by the Author

6.4.2 Participants

We have performed the experiments with 27 participants, 12 non-clinical and 15 clinical cases. Of these 27 participants, 66.67% of the participants were women ages 18 and 38. Male participants represent 33.33%, aged between 19 and 38 years. We can check the relation of some data of the participants in the Table 3. Among the data present in the table are the participant's reference name, which for privacy reasons, we kept as 'Participant', and the experiment number. The second column presents whether the participant is a clinical participant or not. The third column presents the participant's gender and valuable information when evaluating the data directly related to the levels of the measured signals. The data of age (fourth column), weight (fifth column), height (sixth column), and Body Mass Index (BMI) is an international measure used to calculate whether a person is an ideal weight (DISEASE CONTROL AND PREVENTION, 2020).

6.5 Scenario 05 - Prediction - Offline

This section presents a description of the data preparation to generate the artificial intelligence model. In the preparation of the data, we first separate the ranges of values used in training, and then we apply the windowing technique to reinforce the characteristics of the data. In the next step, we present the algorithms used in training to demonstrate the algorithm with the best performance among Machine Learning approaches to generate the model. We used the data obtained through the stress induction experiment described in the 6.4 section in a controlled experiment; all carried out in the exact location.

Table 3 – List of Participants - First Experiment

Participants	Clinical	Gender	Age [y]	Weight [kg]	Height [m]	BMI[kg/m2]
Participant 01	No	Woman	26	91	1.65	33.43
Participant 02	No	Woman	31	62	1.60	25.39
Participant 03	No	Woman	26	58	1.56	23.83
Participant 04	No	Woman	24	73	1.59	28.88
Participant 05	No	Woman	24	61	1.58	24.44
Participant 06	No	Woman	24	64	1.55	26.64
Participant 07	Yes	Woman	38	68	1.57	23.53
Participant 08	No	Woman	32	58	1.64	21.56
Participant 09	Yes	Woman	33	90	1.73	30.07
Participant 10	No	Man	20	86	1.67	30.84
Participant 11	No	Woman	30	52	1.58	21.10
Participant 12	Yes	Woman	36	67	1.63	25.22
Participant 13	No	Man	37	79	1.75	25.80
Participant 14	Yes	Man	21	71	1.75	23.18
Participant 15	Yes	Man	34	94	1.86	27.17
Participant 16	Yes	Woman	28	78	1.59	30.85
Participant 17	Yes	Woman	37	62	1.60	24.22
Participant 18	Yes	Woman	18	49	1.57	19.88
Participant 19	Yes	Woman	22	62	1.58	24.84
Participant 20	Yes	Woman	38	83	1.70	28.72
Participant 21	No	Woman	32	64	1.63	24.09
Participant 22	Yes	Woman	20	64	1.65	23.51
Participant 23	Yes	Man	23	85	1.77	27.13
Participant 24	Yes	Man	23	58	1.72	19.61
Participant 25	Yes	Woman	34	57	1.53	24.35
Participant 26	Yes	Man	19	102	1.71	34.88
Participant 27	No	Man	31	82	1.77	27.45

Source: Elaborated by the Author

6.5.1 Data Selection

The data selection step ensures that all categories have the same amount of data. This step has relevance because it is crucial for proper training that all categories provide the same amount of data for the training. An adequate balance of the dataset prevents the model from having a classification tendency. The dataset has 13 minutes of acquisition for each participant. Of these 13 minutes, the first and last minutes are guard values.

Removing these values, we have 11 minutes of data, of which the initial 5 minutes refer to pre-stress, 1 minute of stress, and the last 5 minutes of post-stress. To balance the data, we defined that the data that we would use for pre-and post-stress would be the third minute because in an evaluation carried out, these points correspond better to the objective of the classifier step. In other words, of the 11 minutes acquired in the experiment, we selected the third, sixth, and ninth minutes. Thus, the data we used during the AI experiment totaled 3 minutes of acquisition per participant, with 27 participants.

6.5.2 Windowing

The windowing process aims to reinforce the characteristics of physiological signals. For this, the data, after going through the filtering process, are arranged into matrices. Each column is a signal, and the lines its instant in time. We defined 600 ms windows for each of the signals with overlapping of 50% for each window, according to Figure 31.

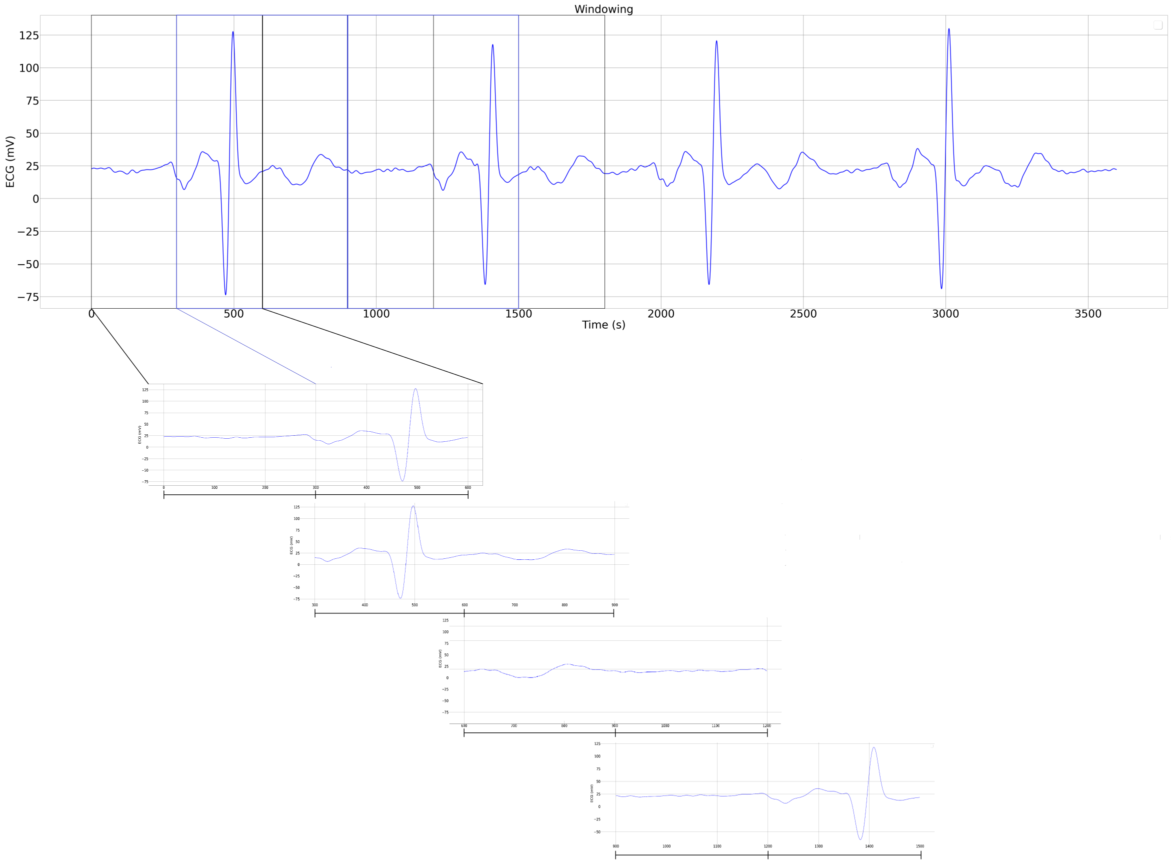


Figure 31 – Example of Data Windowing Technique

As seen in Figure 31, we split the data into 600 ms windows, and windowing occurs with the overlapping of these windows by sliding one over the other. For example, the first window occurs from time 0 ms to 600 ms, so we relocate these selected data to the new dataset. The second window has an overlapping of 50%. This overlapping of 50% means that the window advances 300 ms, thus forming a time window of 300 ms to 600 ms. We relocate this second window again into the new dataset, repeating 300 ms of data. This technique tends to reinforce the characteristics of the trained data. This process repeats for the entire dataset.

6.5.3 Machine Learning Algorithms and Evaluation

We used eight different machine learning algorithms for the model training to explore the approach to present the best results since the training performed had multiple physiological signals. The Machine Learning algorithms used are SVM, NB, DT, Ada Boost Classifier, Multilayer Perceptron, kNN, RF, and Gaussian Process (RUSSELL; NORVIG, 2009). In addition to the variation of algorithms used, we performed experiments with different approaches regarding the dataset.

We performed the ML experiments with variations regarding the participants and the classification objectives. We performed experiments with all and part of the dataset dedicated to identifying the clinical and non-clinical behavior. We also evaluated classification with the three classes (pre-stress, stress, and post-stress) and with some binary combinations. We described the complete set of experiments in the following.

We performed a series of ML experiments using five different classification algorithms. Among the algorithms used, we can mention DT, kNN, RF, Ada Boost, and MLP. We separated these experiments into four sets. The first was using binary classification with all participants. The second experiment had all participants and classification using the three classes. Finally, we performed experiments separating the dataset into clinical and non-clinical participants. For each of these sets, we carried out experiments to detect stress.

Furthermore, it is essential to evaluate broadly the results obtained. In the evaluation step, we used the accuracy, precision, F1, and Recall metrics. In addition to the before-mentioned metrics, we also used the confusion matrix to assess how the model classified the data. Moreover, we use a cross-evaluation method to evaluate the possible overfitting. We selected the kFold method for the cross-evaluation.

6.6 Scenario 06 - Prediction - Online

This section presents the methodology used in the experiments to validate the detection in real-time. To evaluate the real-time detection, we performed new experiments for stress induction. The new experiments followed the same steps as the first experiment to compose the dataset. In this new experiment, we performed 20 acquisitions; 8 participants were clinical and 12 non-clinical participants. Of the analyzed participants, 7 participants were in the first experiment, and 13 we acquired for the first time.

6.6.1 Real-time Detection

We can check the data of the participants in Table 4. Among the data present in the table are the participant's reference name, which for privacy reasons, we kept as 'Participant', and the new experiment number. The second column is whether the participant is a clinical participant or not. The third column presents the participant's gender and valuable information when evaluating the data directly related to the levels of the measured signals. The data of age (fourth column), weight (fifth column), height (sixth column), and Body Mass Index (BMI) is an international measure used to calculate whether a person is an ideal weight (DISEASE CONTROL AND PREVENTION, 2020). The last column indicates if the participant was in the first experiment and which was his serial identification in the first experiment.

Table 4 – List of Participants - Second Experiment

Part.	**Clinical**	**Gender**	**Age [y]**	**Weight [kg]**	**Height [m]**	**BMI[kg/m2]**	**New Part.**
Part. 01	No	Man	32	1.77	89	28.41	No (27)
Part. 02	No	Man	47	1.65	88	32.32	Yes
Part. 03	No	Woman	28	1.67	70	25.09	Yes
Part. 04	No	Man	29	1.79	100	31.21	Yes
Part. 05	No	Woman	31	1.60	65	25.39	No (2)
Part. 06	No	Man	43	1.82	100	30.18	Yes
Part. 07	No	Woman	25	1.55	66	27.47	No (6)
Part. 08	Yes	Man	35	1.86	98	28.32	No (15)
Part. 09	Yes	Woman	33	1.73	85	28.40	No (9)
Part. 10	Yes	Woman	19	1.58	50	20.02	No (18)
Part. 11	Yes	Woman	38	1.57	78	31.64	Yes
Part. 12	Yes	Woman	33	1.71	73	24.96	Yes
Part. 13	No	Woman	25	1.53	50	21.35	Yes
Part. 14	No	Woman	29	1.64	64	23.79	Yes
Part. 15	No	Woman	32	1.63	65	24.46	No (21)
Part. 16	Yes	Woman	43	1.60	84	32.81	Yes
Part. 17	Yes	Woman	35	1.60	77	30.07	Yes
Part. 18	No	Woman	25	1.63	73	27.47	Yes
Part. 19	No	Woman	35	1.68	64	22.67	Yes
Part. 20	Yes	Woman	33	1.60	69	26.95	Yes

Source: Elaborated by the Author

The first step of the experiment is the placement of ATHENA I electrodes on the participant, after which the participant fills in the sociodemographic data. As in the first experiment, we instructed the participant to remain in their relaxed state for 5 minutes. After this period, we again instruct the participant to think about a situation that has recently made him/her distressed or upset to acquire the stress stage. After 1 minute, we instruct the participant to return to their normal state; this step lasts for 5 minutes.

After carrying out the experiments, we analyzed the data obtained. ATHENA I generated a record file for each participant with the classification values for each analyzed moment. To automatically analyze the classified values, we developed a code to evaluate them classified values. The developed code considers the start of the experiment and, according to the time, evaluates whether the architecture classified the signals correctly.

Finally, we performed new training and classification experiments using Machine Learning for the data from both stress induction experiments. Again, we carried out the experiments using the algorithms Decision Tree, kNN, Random Forest, Ada Boost, and Multilayer Perceptron. In the next chapter, chapter 7, we will present the results obtained in the evaluation scenarios throughout the experiments. We present the results and evaluate them to verify plausibility.

7 RESULTS

In this chapter, we present results obtained in the experimentation stages. We describe the results obtained during the stability testing and the results obtained in the practical experiments. Concerning the data obtained in the practical experiments, we perform a more in-depth analysis, evaluating the results and the prototype. The chapter has seven sections, of which three address subjects such as the dataset, the volume of data generated, and an analysis of the signals obtained. The other four sections cover the Machine Learning experiments, real-time acquisition experiment, new ML experiment with all the data, and a brief discussion justifying the results.

7.1 Description of Dataset Acquired Through Stress Induction Experiments

The acquired dataset has a total of 27 participants. For each participant, the acquisition period performed was approximately 13 minutes. We must remove the first and last minute because these data refer to pre and post-experiment. For each participant, we obtained about 700,000 lines of raw data. As mentioned earlier, the architecture provided the data in three ways, via text files (.txt), separated by semicolons, in CSV files, separated in the same way as was done in the text file, separated by semicolons. Finally, we have the database, which provides the data from the dataset already processed.

The developed dataset has columns called participant's name and surname, hidden for privacy reasons. The dataset also has columns with electrocardiograms, electrodermal activity, respiratory pattern, date, and time.

For simple analysis, we recommend using the database. Because, in this case, it is not necessary to apply filters or processing. Since the data has already been previously filtered and processed, it allows the graphical analysis of the data more directly. However, if we want to use the raw data, we recommend using the data in CSV format because we can select which participants we want to use. One of the reasons is that we do not need to move the entire database, which has the data of all participants and, in addition, also has the data already processed, as in the section 7.2.

7.2 Volume of Data Generated by the Architecture

Based on the acquisition made for the composition of the training dataset, we can obtain some relevant data. It is possible to extrapolate the values for a longer acquisition period or even a period of continuous acquisition. We estimate the storage space required for a given project or evaluate the possible acquisition period time. Table 5 contains the values obtained during the experiment. We compare the values using the three storage methods used by the architecture.

Regarding the complete dataset, the total values were slightly higher since the period ac-

Table 5 – Amount of Data

Time	txt [MB]	CSV [MB]	SQLite [MB]
1 min.	2.5	2.6	5.6
5 min.	12.8	13.5	27.3
10 min.	25.6	27.1	56.8
15 min.	37.9	39.2	86.3
30 min.	75.6	77,2	172.6
60 min.	151.0	154,3	346.7
120 min.	299.7	305.5	691.5

Source: Elaborated by the Author

quired was longer. The volume of acquired data referring to the text files generated approximately 824.4 MB of storage. The experiment generated 357 files for 27 participants acquired. Regarding CSV files, 27 files were generated, totaling 842.6 MB of files, representing an increase of 2.21% concerning CSV files. Finally, the database generated a single file, with all the data organized in tables. The generated dataset file contains 1,544.3 MB. It represents an increase of 83.27% in the CSV data and an increase of 87.32% in data stored in text form.

7.3 Analysis of Acquired Signals

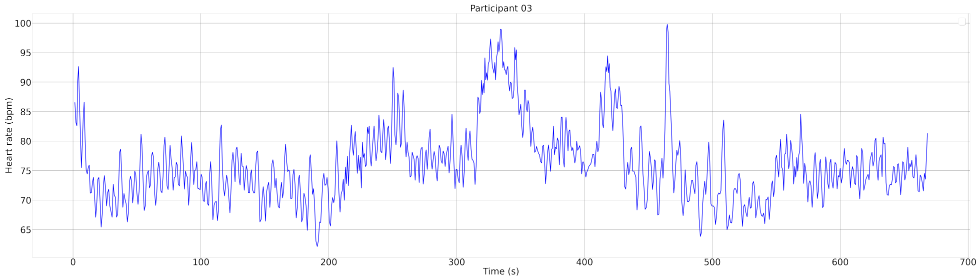
After collecting the data for the dataset, it is necessary to evaluate the data to verify its integrity of the data. We also must check whether the data represents the expected theory. We performed an initial graphic analysis of the data for all participants. We found that the non-clinical participants in the study showed a more significant change in the physiological signals during the stress induction period, and soon after the stress reduction period, the signals returned to normal. However, we observed that clinical participants had a lower variation during the stress-inducing period.

It is important to note that the data evaluated in this stage we delimited to show only the period of the experiment itself, which comprises 660 seconds (11 minutes). There are 300 seconds of pre-stress, 60 seconds of stress induction, and 300 seconds post-induction in each experiment. To assess the differences between the two groups, we selected two participants, with Participant 03 being a non-clinical case and Participant 23 as a clinical case.

We can evaluate in Figures 32 (Participant 03) and 33 (Participant 23) the ECG signals for the two participants. In analysis, for Participant 03, it is evident that the heart rate increased at 300 s at the beginning of the stress induction period, and there was a decrease at 360 s. For Participant 23 (Figure 33), the increase in heart rate does not have a significant increase as observed in Participant 03. It is important to note that the beginning of Participant 03's signal increases the heart rate. It happens because the participant is agitated at the beginning of the process, which during the meditation period decrease.

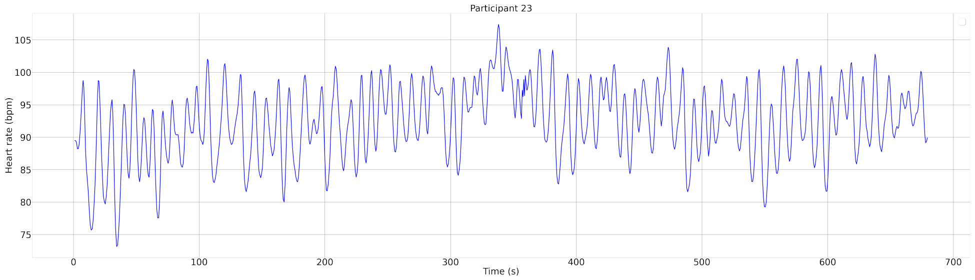
We can verify a similar behavior in the signals related to RSP as we have seen in the Heart

Figure 32 – Participant 03 (Non-clinical) - ECG



Source: Elaborated by the Author

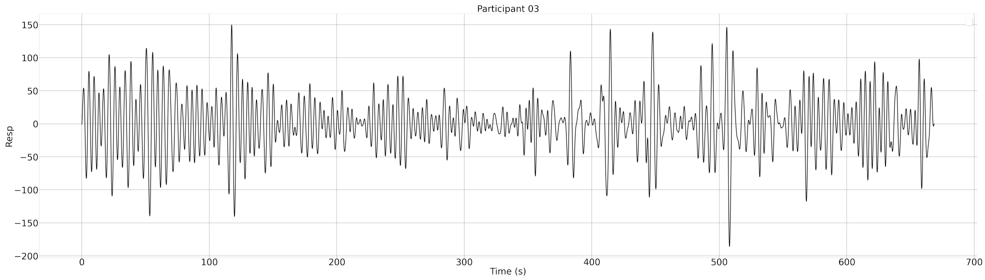
Figure 33 – Participant 23 (Clinical) - ECG



Source: Elaborated by the Author

Rate. The non-clinical participant (Figure 34) shows a significant change in frequency and amplitude during the stress induction stage at 300 seconds. This behavior is not evident in the clinical participant (Figure 35). There is an increase in the clinical participant's signal amplitude during the stress induction stage. This increase is the opposite behavior observed in the non-clinical participant. Another aspect that we can point out is that the signal related to Participant 23 shows a lower respiratory rate during the entire period observed.

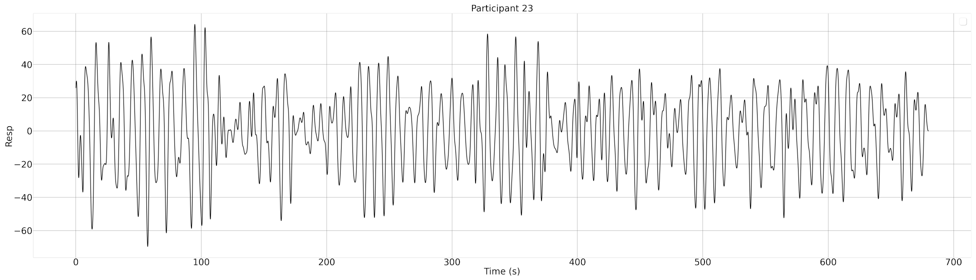
Figure 34 – Participant 03 (Non-clinical) - RSP



Source: Elaborated by the Author

Finally, the last signal we obey is the EDA for the same two participants observed previously. We may observe similar behavior in both participants. However, for the non-clinical participant

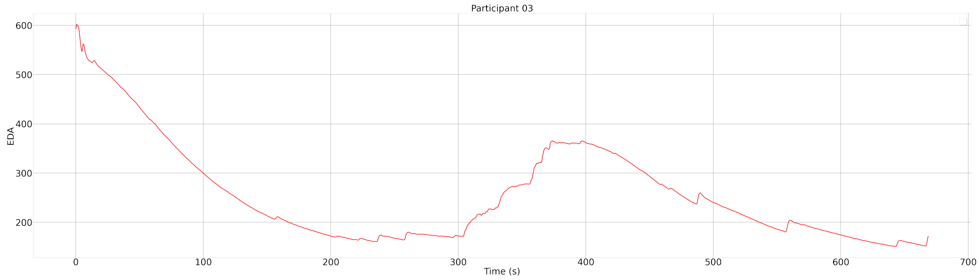
Figure 35 – Participant 23 (Clinical) - RSP



Source: Elaborated by the Author

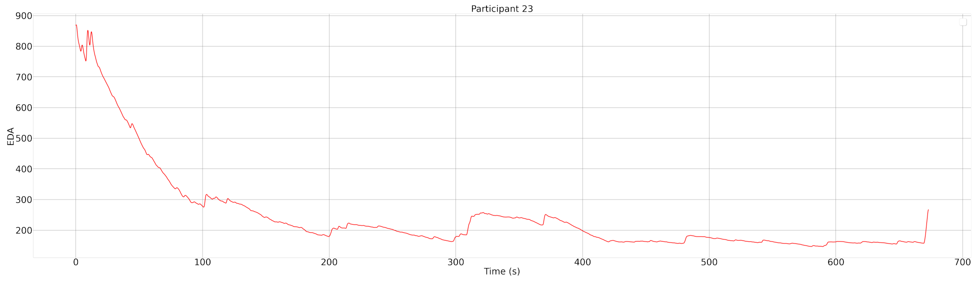
(Figure 36), the electrical conductivity of the skin reaches a higher value when compared to the electrical conductivity of the clinical participant (Figure 37). Another difference that we must point out between the two cases is that Participant 03 takes longer to return to the natural level of electrical conductivity of the skin compared to the behavior of the EDA for Participant 23.

Figure 36 – Participant 03 (Non-clinical) - EDA



Source: Elaborated by the Author

Figure 37 – Participant 23 (Clinical) - EDA



Source: Elaborated by the Author

To represent the results, we calculated the average of the signals for each participant. To get the most accurate data possible and to be able to carry out a relative comparison, for each of the three-stage, we have used the same period (1 min). As the pre and post-stress stages have 5 minutes, we used the period between the 3rd and 4th minutes of the experiment (pre-stress).

The period we selected to represent the post-stress was from 8 to 9. In the data comparisons, we have added relevant information to compare the data between participants, such as gender and whether the participant is a clinical or non-clinical case.

In the Table 6, check the participant data for the ECG data. In the first column, we have the participant. In the second and third columns, we have the information on whether it is a clinical case or not and the participant's gender. In columns 4, 5, and 6, we can see the data for the pre-stress, stress, and post-stress classes, all measured in bpm.

Table 6 – Participants' ECG Averages

Participant	Clinical	Gender	Pre-Stress [bpm]	Stress [bpm]	Post-Stress [bpm]
Participant 01	No	Woman	66.592	78.049	63.836
Participant 02	No	Woman	68.123	84.257	95.049
Participant 03	No	Woman	73.104	87.008	72.252
Participant 04	No	Woman	84.391	84.424	89.686
Participant 05	No	Woman	88.400	93.082	82.289
Participant 06	No	Woman	76.672	80.065	76.989
Participant 07	Yes	Woman	79.733	81.085	79.564
Participant 08	No	Woman	89.247	87.351	82.599
Participant 09	Yes	Woman	73.378	73.891	73.360
Participant 10	No	Man	65.476	68.611	67.273
Participant 11	No	Woman	89.447	88.094	86.193
Participant 12	Yes	Woman	82.818	83.528	84.947
Participant 13	No	Man	58.879	59.066	59.854
Participant 14	Yes	Man	103.680	104.695	107.264
Participant 15	Yes	Man	84.596	86.497	82.593
Participant 16	Yes	Woman	79.052	83.439	82.047
Participant 17	Yes	Woman	85.897	84.154	86.257
Participant 18	Yes	Woman	103.470	105.820	103.517
Participant 19	Yes	Woman	75.679	77.410	73.893
Participant 20	Yes	Woman	90.271	91.906	93.488
Participant 21	No	Woman	84.718	85.428	86.716
Participant 22	Yes	Woman	89.880	89.992	88.209
Participant 23	Yes	Man	91.731	97.519	91.738
Participant 24	Yes	Man	70.135	71.968	73.600
Participant 25	Yes	Woman	77.758	78.437	78.266
Participant 26	Yes	Man	71.486	66.985	69.151
Participant 27	No	Man	69.526	73.542	68.677

Source: Elaborated by the Author

Table 7 presents the averages of the values obtained for the RSP signal. As with the ECG data (Table 6), the first, second, and third columns present participant, clinical status, and gender data, respectively. Columns 4, 5, and 6 present the RSP data for the pre-stress, stress, and post-stress classes, all in volts [V], since we have gotten the signals through the piezoelectric sensor.

Finally, we have the Table 8 Table that demonstrates the EDA averages, as well as As Tables 6 and 7 have the same three starting columns, participant, clinical status and gender. However,

Table 7 – Participants' RSP Averages

Participant	Clinical	Gender	Pre-Stress [V]	Stress [V]	Post-Stress [V]
Participant 01	No	Woman	0.251	0.298	0.289
Participant 02	No	Woman	0.256	0.282	0.287
Participant 03	No	Woman	0.252	0.242	0.231
Participant 04	No	Woman	0.284	0.282	0.276
Participant 05	No	Woman	0.241	0.227	0.235
Participant 06	No	Woman	0.178	0.233	0.217
Participant 07	Yes	Woman	0.162	0.181	0.186
Participant 08	No	Woman	0.263	0.268	0.243
Participant 09	Yes	Woman	0.164	0.185	0.245
Participant 10	No	Man	0.258	0.219	0.250
Participant 11	No	Woman	0.224	0.262	0.226
Participant 12	Yes	Woman	0.293	0.291	0.320
Participant 13	No	Man	0.263	0.267	0.259
Participant 14	Yes	Man	0.296	0.291	0.292
Participant 15	Yes	Man	0.245	0.303	0.294
Participant 16	Yes	Woman	0.219	0.246	0.214
Participant 17	Yes	Woman	0.322	0.317	0.327
Participant 18	Yes	Woman	0.246	0.235	0.222
Participant 19	Yes	Woman	0.213	0.204	0.225
Participant 20	Yes	Woman	0.278	0.274	0.270
Participant 21	No	Woman	0.222	0.272	0.224
Participant 22	Yes	Woman	0.216	0.240	0.264
Participant 23	Yes	Man	0.196	0.141	0.112
Participant 24	Yes	Man	0.262	0.264	0.250
Participant 25	Yes	Woman	0.242	0.243	0.255
Participant 26	Yes	Man	0.293	0.299	0.296
Participant 27	No	Man	0.167	0.205	0.218

Source: Elaborated by the Author

columns 4, 5, and 6 present EDA data for the same classes, measured in microsiemens [µS].

7.4 Machine Learning Model Training

We performed a series of Machine Learning training experiments evaluating classification approaches and exploring the results with the acquired dataset. The new dataset has two different groups of participants, clinical and non-clinical participants, so we also sought to evaluate them in isolation. According to the allostatic load theory (CORRIGAN et al., 2021), people with some mental disorders may present different behavior in stressful situations. Furthermore, clinical people may not feel anything in a stressful situation. Alternatively, after starting a stressful situation, they can take more time to return to a normal state or even do not return.

We separated the results obtained in the Machine Learning experiments according to the set of participants used and the type of classification. In Table 9, we listed the results obtained in the

Table 8 – Participants' EDA Averages

Participant	Clinical	Gender	Pre-Stress [μS]	Stress [μS]	Post-Stress [μS]
Participant 01	No	Woman	1.655	1.239	0.957
Participant 02	No	Woman	3.230	3.195	3.030
Participant 03	No	Woman	1.701	3.363	2.003
Participant 04	No	Woman	1.427	1.237	1.143
Participant 05	No	Woman	4.747	3.795	2.585
Participant 06	No	Woman	5.846	6.885	5.363
Participant 07	Yes	Woman	0.580	0.552	0.538
Participant 08	No	Woman	10.20	10.200	10.200
Participant 09	Yes	Woman	5.487	4.923	3.890
Participant 10	No	Man	6.378	6.371	4.145
Participant 11	No	Woman	7.535	7.634	5.822
Participant 12	Yes	Woman	0.856	0.826	0.807
Participant 13	No	Man	2.959	3.136	2.777
Participant 14	Yes	Man	6.764	6.869	9.165
Participant 15	Yes	Man	6.79	6.294	5.702
Participant 16	Yes	Woman	0.751	0.661	0.647
Participant 17	Yes	Woman	1.644	1.533	1.574
Participant 18	Yes	Woman	2.370	1.895	1.981
Participant 19	Yes	Woman	8.626	8.532	8.402
Participant 20	Yes	Woman	1.950	1.989	1.604
Participant 21	No	Woman	1.878	1.878	2.013
Participant 22	Yes	Woman	9.082	7.778	7.278
Participant 23	Yes	Man	10.200	10.117	9.876
Participant 24	Yes	Man	2.627	3.130	3.010
Participant 25	Yes	Woman	3.399	2.543	2.375
Participant 26	Yes	Man	8.962	7.917	6.560
Participant 27	No	Man	6.286	6.793	5.362

Source: Elaborated by the Author

experiments using all participants to perform binary classification. The two classes considered are baseline and stress. The Decision Tree and Random Forest classification algorithms stand out, with results above 95%. Algorithms tested such as kNN and Ada Boost also showed promising results for binary classification but in the range of 92%.

Table 9 – Results from all participants - First Experiment | Classes: Baseline and Stress

Algorithm	Accuracy	F1	Precision	Recall	kFold
DT	97.512	97.512	97.502	97.512	97.117
kNN	91.212	91.211	91.233	91.212	89.988
RF	98.207	98.207	98.207	98.207	97.320
Ada Boost	92.601	92.598	92.632	92.601	92.655
MLP	55.858	52.887	58.119	55.858	56.723

Source: Elaborated by the Author

The second set of tests performed involved all participants, classifying three classes. The predicted classes are pre-stress, stress, and post-stress. The results verified (Table 10) in this set of experiments showed significantly lower accuracy when compared with the first experiment. However, we consider this result expected, considering that the classification went from two to three classes. In this set of tests, the algorithms that stand out are still Decision Tree and Random Forest, with 90% and 92% accuracy, respectively.

Table 10 – Results from all participants - First Experiment | Classes: Pre-stress, Stress, and Post-stress

Algorithm	Accuracy	F1	Precision	Recall	kFold
DT	90.522	90.521	90.521	90.522	86.341
kNN	72.046	72.075	72.136	72.046	69.992
RF	92.720	92.724	92.725	92.725	89.697
Ada Boost	57.605	57.279	57.273	57.605	58.042
MLP	33.551	18.173	33.357	33.551	33.806

Source: Elaborated by the Author

In the third and fourth Machine Learning experiments performed, we divided the dataset into non-clinical and clinical participants, using the three classes. This division aimed to identify whether any of these two sets had any impact on the trained values obtained. According to the allostatic load theory, we expected lower results with the clinical participant set. The third experiment, with non-clinical participants (Table 11), showed better values when compared to training values with all participants. The accuracy for DT and RF was above 93%.

Table 11 – Results of non-clinical participants - First Experiment | Classes: Pre-stress, Stress, and Post-stress

Algorithm	Accuracy	F1	Precision	Recall	kFold
DT	93.035	93.036	93.037	93.035	90.532
kNN	80.861	80.898	80.962	80.861	79.536
RF	94.564	94.564	94.564	94.564	92.845
Ada Boost	64.917	64.703	64.817	64.917	65.157
MLP	41.428	39.836	43.532	41.428	39.561

Source: Elaborated by the Author

In the fourth experiment we carried out with clinical participants, the values obtained were lower than non-clinical ones, according to Table 12. We expected this range of values, as the signals of clinical participants tend not to respond in a standard way in stressful situations. Therefore the model presents difficulties in classifying these individuals correctly. Even though the values obtained are lower than those of the previous experiments, the accuracy and precision values for Decision Tree and Random Forest continued to be above 90%.

The values obtained in Cross-Validation are a means of varying the model's generalization. Thus, in these experiments, all values obtained are close to the accuracy and precision values for the data sets and the algorithms, with an average of 3% difference.

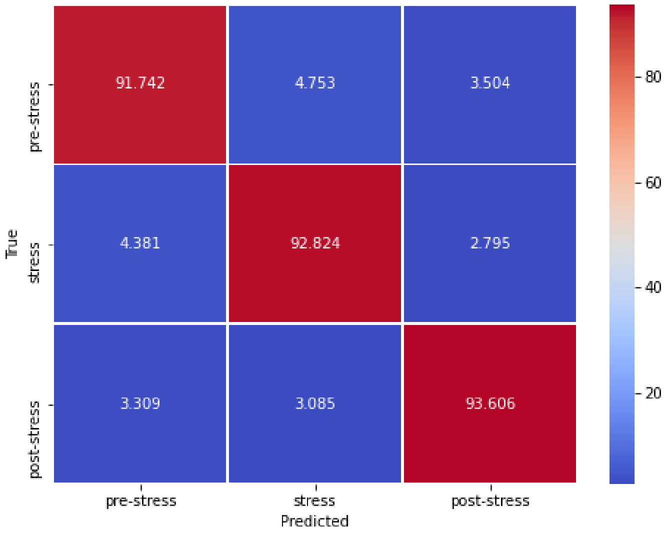
Table 12 – Results of clinical participants - First Experiment | Classes: Pre-stress, Stress, and Post-stress

Algorithm	**Accuracy**	**F1**	**Precision**	**Recall**	**kFold**
DT	90.081	90.081	90.083	90.081	86.547
kNN	70.276	70.289	70.322	70.276	68.407
RF	92.083	92.081	92.083	92.081	88.940
Ada Boost	60.140	59.986	60.071	60.140	59.046
MLP	36.344	30.248	35.039	36.344	34.242

Source: Elaborated by the Author

Another way to validate the values obtained in training is using confusion matrices. We selected the algorithm Random Forest Classifier with three classes for this verification. We first analyzed the results with all participants. Next, we compared the set of non-clinical and clinical participants. In Figure 38, we look at the predicted values in each class in percentage terms using DT. The number of occurrences is related to the different data obtained with the windowing process.

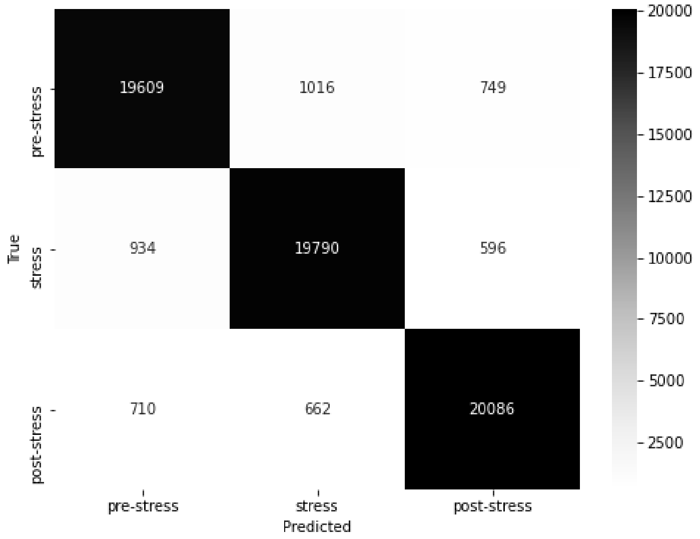
Figure 38 – Confusion Matrix Percentage Values of all participants | Classes: Pre-stress, Stress, and Post-stress



Source: Elaborated by the Author

As we can see, the classification occurred adequately, with the Post-Stress classifications with the highest number of correct classifications. In the Figure 39, we check the confusion matrix with absolute values. By analyzing the matrix, we verified that the correctly predicted values in each class were very close. We can also highlight that the class that presented the greatest error was pre-stress, in which the model classified it as stress. The confusion between the two classes occurred in the opposite direction because the second-biggest misclassification was the model's stress values classified as pre-stress.

Figure 39 – Confusion Matrix Absolute Values of all participants | Classes: Pre-stress, Stress, and Post-stress



Source: Elaborated by the Author

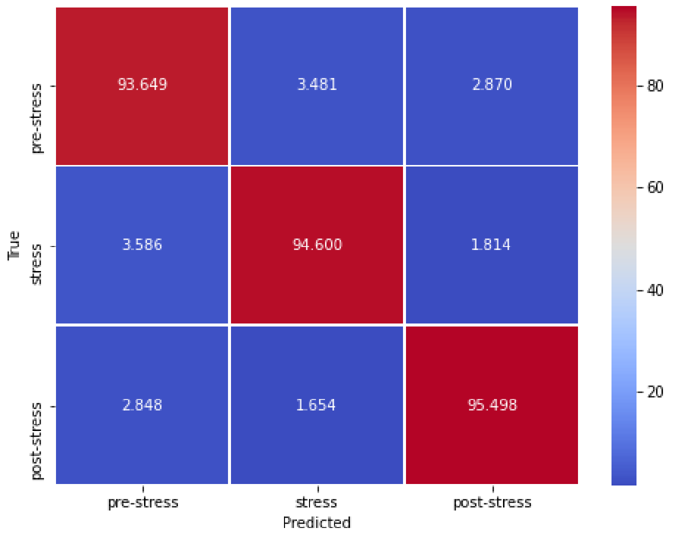
On the other hand, the model made fewer classifications between post-stress and stress. We can justify it because the final period of participant stress is at its maximum stress level. The post-stress used has already passed 2 minutes of the participant trying to get back to his basal state.

We can analyze the impact of allostatic load on non-clinical and clinical participants by comparing the confusion matrix. In Figure 40, we have the confusion matrix in percentage values for non-clinical participants with a three-class classification. In figure 41, we see the confusion matrix for clinical participants, also in percentage values for three classes. Both matrices are obtained from the training using RF. Comparing the values of the two matrices, we verified an average decrease of 2.5% in the model accuracy with clinical participants compared to non-clinical participants. The most significant difference found is in post-stress, in which the classification for non-clinical participants reached 95.49% and dropped to 92.71% with clinical participants.

If we analyze the matrices of non-clinical (Figure 42) and clinical (Figure 43) participants in absolute values, we verify the magnitude of the difference in classifications. Before performing this direct comparison of absolute values, it is necessary to highlight the number of participants in each group in the dataset. The dataset has 12 non-clinical and 15 clinical participants, totaling three more participants for the set of clinical participants. One of the possible comparisons between the two groups is the significant increase in post-stress classification errors. This aspect aligns with our expectations because clinical participants have greater difficulty returning to their normal state.

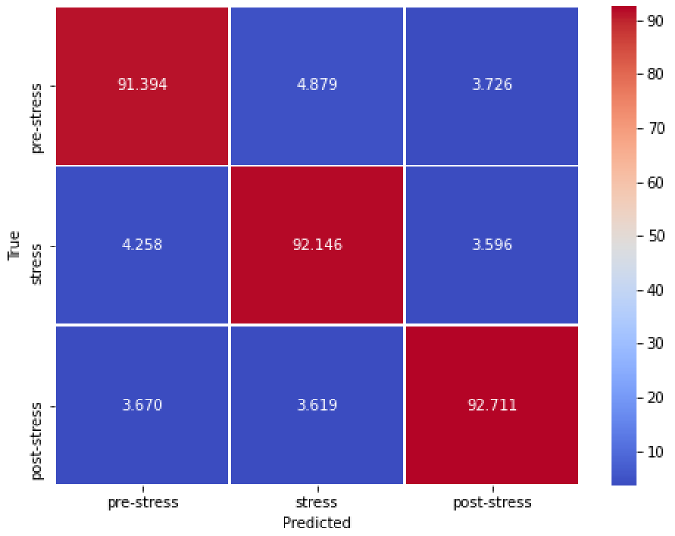
Furthermore, clinical participants already have an altered baseline state. When compared

Figure 40 – Confusion Matrix Percentage Values of non-clinical participants - Classes: Pre-stress, Stress, and Post-stress



Source: Elaborated by the Author

Figure 41 – Confusion Matrix Percentage Values of clinical participants - Classes: Pre-stress, Stress, and Post-stress

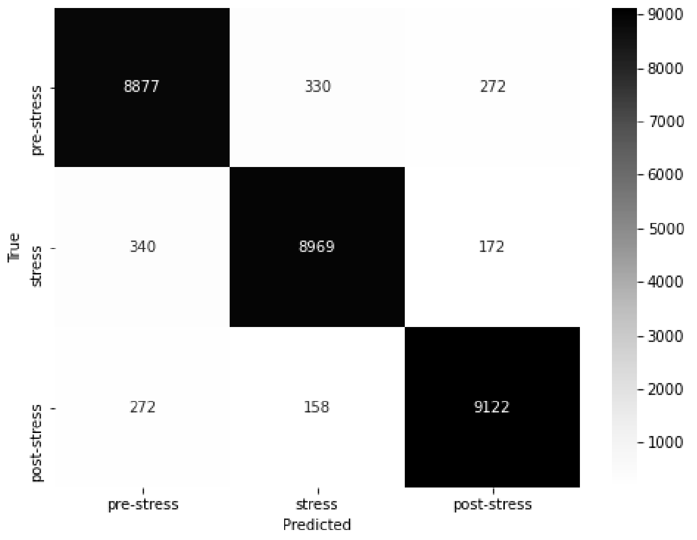


Source: Elaborated by the Author

to clinical participants, we found fewer errors in this classification—the previously verified characteristic of the most significant number of errors occurring when classifying stress and pre-stress remains. The highlight is the increase in post-stress classification errors.

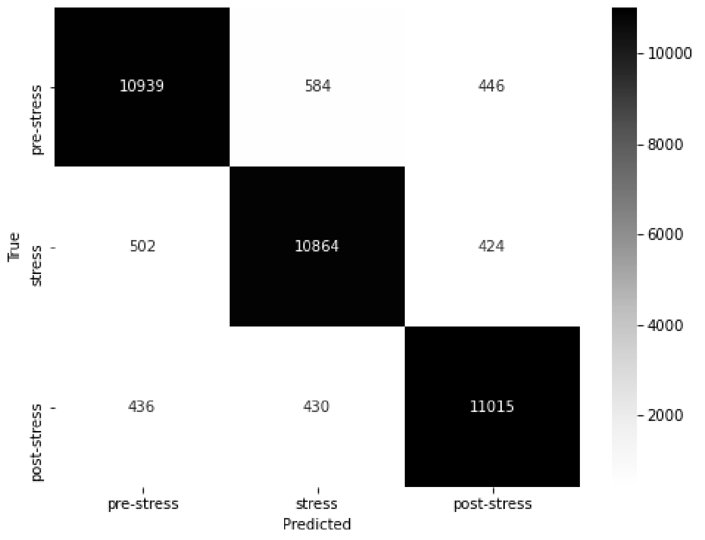
We performed a brief analysis of the means of some clinical and non-clinical participants

Figure 42 – Confusion Matrix Absolute Values of non-clinical participants - Classes: Pre-stress, Stress, and Post-stress



Source: Elaborated by the Author

Figure 43 – Confusion Matrix Absolute Values of clinical participants - Classes: Pre-stress, Stress, and Post-stress



Source: Elaborated by the Author

in a comparative way. Using ECG data as a basis, we can verify the behavior of the signs. It justifies the drop in the accuracy value. In Table 13, we have the average heart rate of eight participants for the three stages of the protocol. Four of these participants are non-clinical cases, and the other four are clinical cases. By analyzing the non-clinical participants, we verified the expected behavior in the stress-inducing protocol. Starting with lower heart signals, during the stress period, an increase in the signals, and during the stress period again a relaxation, thus decreasing the heart rate levels.

Table 13 – Comparison of ECG values between participants

Participant	Clinical	Pre-Stress [bpm]	Stress [bpm]	Post-Stress [bpm]
Part. 01	No	66.592	78.049	63.836
Part. 03	No	73.104	87.008	72.252
Part. 05	No	88.400	93.082	82.289
Part. 06	No	76.672	80.065	76.989
Part. 12	Yes	82.818	83.528	84.947
Part. 17	Yes	85.897	84.154	86.257
Part. 18	Yes	103.470	105.820	103.517
Part. 20	Yes	90.271	91.906	93.488

Source: Elaborated by the Author

However, if we analyze the clinical participants' heart rates, we see different behaviors in the signs. For example, participant 12 has an average of 82.81 bpm in pre-stress, the heart hates rises to 83.53 bpm, and in post-stress, instead of from lower, the heart rate rises to 84.95 bpm. Participant 17, on the other hand, has a heartbeat behavior that is the opposite of what we expected, starting with 85.90 bpm, decreasing to 84.15 bpm, and rising again to 86.26 bpm. Analyzing participant 18, we verified that the behavior is as expected but at much higher levels when compared to the other participants. Finally, participant 20 had similar behavior to participant 12. The data start lower in the first stage, in the second stage, the signals rise, and in the third stage, there is a new average increase in heart rate, showing that the participant could not return to his basal state.

It is essential to highlight that clinical participants showed signals as expected and non-clinical participants with unexpected behaviors. There are two possible justifications for the non-clinical participants to have shown different signs. The first is the experiment's effectiveness, as the experiment requires the participant's concentration. The second possibility is that the non-clinical participant may have some undiagnosed or untreated clinical condition, as we did not perform previous evaluations of the non-clinical participants.

7.5 Real-time Detection

In real-time detection experiments, we performed acquisitions with 20 participants; 8 participants were clinical and 12 non-clinical participants. Seven participants were in the first experiment, and 13 were acquired for the first time. For each 30-second acquisition, the architecture generates a unique overall rating. For this unique classification to be generated, the architecture evaluated the signals in this period, classifying them from 300 ms to 300 ms. This generated 300 ms by 300 ms ratings are evaluated and analyzed to generate the overall rating for the period. In order to be classified as a given class, the values must be at least 70% of the classified results.

We can verify the classification accuracy values in the second stress induction experiment in Table 14. The Table presents the participant reference number in the first column, starting with the participants in the first stress-inducing experiment. The second column presents the information if the participant is a clinical or non-clinical case. The third column highlights the data if the participant is new to the experiment or if he is participating again in the experiment. Finally, in the fourth column, we have the accuracy value obtained for each participant.

Table 14 – Real-time Classification

Participant	Clinical	New Part.	Accuracy
Part. 01	No	No - 27	64.52
Part. 02	No	Yes	47.12
Part. 03	No	Yes	31.51
Part. 04	No	Yes	53.29
Part. 05	No	No - 2	73.67
Part. 06	No	Yes	52.32
Part. 07	No	No - 6	60.13
Part. 08	Yes	No - 15	71.52
Part. 09	Yes	No - 9	62.97
Part. 10	Yes	No - 18	78.93
Part. 11	Yes	Yes	61.51
Part. 12	Yes	Yes	38.63
Part. 13	No	Yes	21.78
Part. 14	No	Yes	46.31
Part. 15	No	No - 21	71.29
Part. 16	Yes	Yes	31.37
Part. 17	Yes	Yes	52.94
Part. 18	No	Yes	43.39
Part. 19	No	Yes	59.51
Part. 20	Yes	Yes	57.82

Source: Elaborated by the Author

7.6 Machine Learning Model Update

After performing the second induction experiment, we performed a new Machine Learning training to update the model. The new experiments followed the previously-formed format, training the model using five different Machine Learning algorithms and comparing the values.

As in the previous Machine Learning experiment, we separated the results obtained in the Machine Learning experiments according to the set of participants used and the type of classification. In Table 15, we list the results for all participants using binary classification. The two classes considered are baseline and stress. The new experiments showed a little improved result in binary classification values but still showed better results with the Decision Tree and

Random Forest classification algorithms, with results of 92.38% and 94.62%.

Table 15 – Results from all participants - Second Experiment | Classes: Baseline and Stress

Algorithm	Accuracy	F1	Precision	Recall	kFold
DT	98.351	98.351	98.351	98.351	97.580
kNN	91.592	91.611	91.613	91.592	89.988
RF	98.686	98.686	98.687	98.686	98.016
Ada Boost	92.701	92.691	92.681	92.701	92.925
MLP	56.578	53.817	58.699	56.578	56.723

Source: Elaborated by the Author

Finally, we performed tests with all participants of both stress-inducing experiments, using the three classes. Classes are pre-stress, stress, and post-stress. The verified results (Table 16) showed greater accuracy than previously presented. The results with the best accuracy remain Decision Tree and Random Forest, with 91% and 93% of correct answers, respectively.

Table 16 – Results from all participants - Second Experiment | Classes: Pre-stress, Stress, and Post-stress

Algorithm	Accuracy	F1	Precision	Recall	kFold
DT	91.478	90.521	90.521	91.478	89.617
kNN	72.046	72.361	72.375	72.046	70.620
RF	93.648	93.648	93.649	93.648	91.761
Ada Boost	54.278	54.043	54.030	54.278	55.692
MLP	34.667	20.702	49.006	34.667	36.883

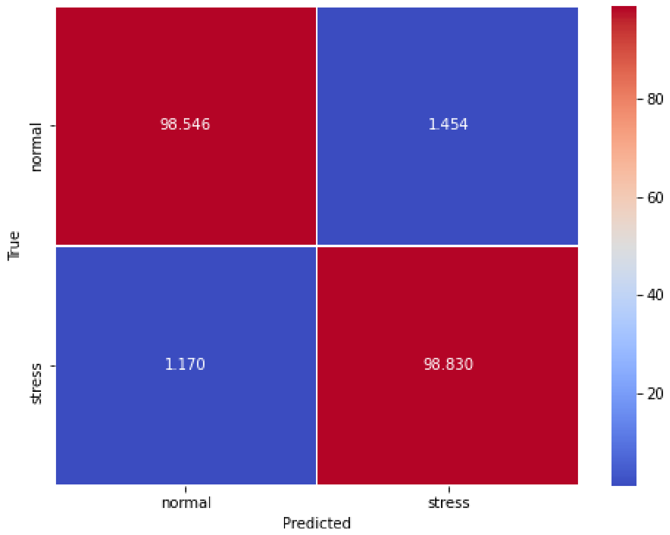
Source: Elaborated by the Author

To evaluate the classified values, we can analyze Figures 44 and 45 that present the confusion matrices for classification using Random Forest. The two matrices present the rank values in percentage terms. Figure 44 presents the confusion matrix with the values of the binary classification, analyzing the Baseline and Stress classes. In Figure 45 the confusion matrix presents the classification for Pre-stress, Stress, and Post-stress.

7.7 Critical Analysis and Discussion

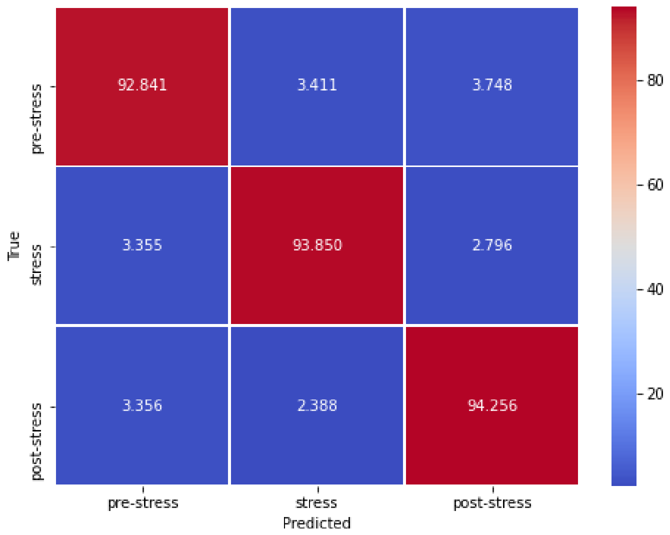
We justified the difference between the values obtained in the ML test experiments and the real-time detection experiments due to the participants in the first experiment not knowing what the experiment would be like. In the second experiment, the participants who had participated in the first experiment no longer had the exact expectations, which could affect the classification result. In addition, there is a one-minute period before and after the stress induction, which tends to present more uncertain signals. In a new training of the model, this time adding the second experiment participants, the model showed a slight improvement in the classifications, with values of 98,686% for binary classification and 93,648% for classification using three classes, both using the Random Forest algorithm.

Figure 44 – Confusion Matrix - Second Experiment - Classes: Baseline and Stress



Source: Elaborated by the Author

Figure 45 – Confusion Matrix - Second Experiment - Classes: Pre-stress, Stress, and Post-stress



Source: Elaborated by the Author

The different values verified for clinical and non-clinical participants are also worth mentioning. If we compare the training model with the set of clinical and non-clinical participants, the values obtained with non-clinical participants were significantly better. However, the values agree with the allostatic load theory. We expected clinical participants to have different physiological signal behavior than non-clinical people. Clinical participants tend to have a naturally higher (non-stress) baseline and do not return to baseline as quickly. The following chapter

(chapter 8) presents the conclusions obtained from the results, in addition to defining the contributions of this work and possible future works.

8 CONCLUSION

With the rapid advancement of technology and society, routines have become much more full of tasks and activities, a dynamic world that does not stop, that requires lots of effort. As a consequence of this effort and continuous exhaustion, we have stress and physical and psychological exhaustion. When a person is under stress, his performance in all aspects of life is affected. Several papers have already demonstrated that stress manifests itself in different ways in the human body. The stress may have a direct and indirect impact on several vital signs.

One way of monitoring these people is through wearable sensors, which can measure physiological signals and thus, identify stress. Stress, like depression, is pointed out as one of the principal diseases of modern society. Wearable sensors have made great strides in recent years, improving their accuracy, flexibility, and autonomy. Furthermore, allowing the user to monitoring a simplified way.

It is essential to highlight that wearable sensors can be used in several areas, such as in the hospital and the industry, to monitor professionals in places requiring monitoring and sports to assess physical performance, among other fields. Even within the hospital area, wearable sensors have a wide applications variety, such as remote monitoring of a patient in rehabilitation or post-surgery. Another alternative is the patients monitoring who need psychological support or even the health professionals' monitoring.

This work seeks to find a solution to these aspects before listed. To achieve this objective, we propose the development of a complete architecture. This architecture makes the acquisition, processing, and detection of physiological signals. The applied context of this work is, already highlighted, stress detection. However, another objective of this work is to provide a flexible architecture that can be retrofitted to other Wearable platforms, including more devices and equipment, if desired.

For this reason, the architecture has four blocks, which can operate synchronously or, with few adjustments, may operate separately. In addition, the architecture can acquire a proprietary dataset, which is the most advisable since the levels of the signals can vary according to the application's purpose or the wearable sensor used in the process.

We can affirm that the developed architecture has excellent stability based on the performed experiments. The architecture did not present a loss of communication or crashes during the experiments. We carried out more than forty-seven practical experiments in which all the acquisitions did not have any oscillation of communication or failure data acquisition. Another relevant aspect we must mention is data storage, which proved to be a reliable form of storage.

The study showed good values in detecting stress in clinical and non-clinical individuals using ML algorithms for classification. Among the algorithms used, we can highlight the Random Forest with 92.72% accuracy in the classification using three classes and the cross-validation test with 89.69%. The three classes we used for detection were pre-stress, stress, and post-stress.

For binary classification, the model using RF Classifier obtained accuracy and cross-validation of 98.20%.

We also verified which classes the model failed the most when analyzing the results. The model showed more wrong classifications among the classes stress and pre-stress. A hypothesis for this misclassification is an exit transition from the pre-stress in part of the stress-inducing period. Thus, the initial stress values tend to be lower and, consequently, more similar to pre-stress.

Another conclusion highlighted is the impact that clinical participants have on the model. If we compare the training model with the set of clinical and non-clinical participants, the obtained values with non-clinical participants were significantly better. Nonetheless, the values are in line with expectations. According to the allostatic load theory, clinical participants tend to have a different behavior toward physiological signals than non-clinical people. Clinical participants may have a naturally higher basal (non-stress) level. Alternatively, their physiological signal levels do not return to normal without intervention when going through a stressful situation.

We can also cite the results obtained with the real-time detection experiments. We performed experiments with 20 participants following the same protocol used in the first experiment. Of these 20 participants, seven were present in the first experiment, and we used their data to train the model. The average accuracy obtained in the experiments with all 20 participants was 52.33%. However, if we only analyze the participants in the first experiment, the average accuracy goes up to 69.00%. These data prove the importance of adjusting the model for each new participant, having a direct impact on the model's performance.

Regarding the analysis of the data acquired for the dataset composition, in collaboration with the Graduate Program in Psychology at UNISINOS, it was concluded that there are significant differences in the data values for clinical and non-clinical participants. The behavior of the data in the two groups is quite different. Non-clinical participants, in general, have a more significant difference in the stress induction stage. The clinical participants have shown a much smaller change. Another verified aspect is that the non-clinical participants tend to have a slower and gradual return to the normal state than clinical participants.

It is imminent to evaluate how we treat these differences by the Machine Learning model with the presented aspects. Another relevant aspect that we must mention is acquiring physiological information from the participants. The physiological aspect collected are age, weight, height, gender, and BMI. As verified in the literature, physiological aspects such as those mentioned directly affect physiological signals. The data must be carefully evaluated during the experiments to generate an efficient artificial intelligence model. We know that maybe the obtained model does not work with every people. There is the possibility that the model must have a target group based on the physiological aspects of the participant. Something similar happens with mechanical prostheses, in which the signals need to be obtained from the user of the prosthesis because the level of the signals is characteristic for each individual (EDWARDS et al., 2015, LEMOYNE et al., 2015).

In synthase, we can list some positive aspects obtained with the work developed so far. The architecture is divided into blocks so that each block can operate autonomously and on different devices. Easy reproducibility, so installation and configuration on different devices are simplified. We can also highlight the integration of continuous monitoring with real-time detection through artificial intelligence. On the negative side, the architecture may need to train a new model with data from the participant.

8.1 Contribution

Among the contributions of this work, we can list at least seven main contributions. Firstly we can mention the mapping of the state-of-the-art approaches for physiological pattern detection, with a literature review conducted considering both computing and healthcare.

Regarding the second contribution, the developed architecture, ATHENA I, is a complete architecture to perform the acquisition, filtering, processing, storage, and detection of physiological signals, reliably and adaptively. We can highlight the way that we had developed the architecture. Focused on being a modular architecture and designed in blocks, thus allowing easy changes, such as changing the wearable sensor used and adding signals to measure. Also, it can change a block without impacting the rest of the architecture. The blocks can operate so that, with some requirements, we can implement another block in place of that one.

Regarding the third contribution, we can highlight the acquisition of a dataset in collaboration with the Graduate Program in Psychology at the University of Vale do Rio dos Sinos (UNISINOS). This dataset aims to detect stress in clinical and non-clinical participants through stress induction based on negative thoughts. Based on the results evaluated, it is possible to verify the difference in signals in non-clinical participants in normal states and under stress. However, in clinical participants, it is necessary to observe other aspects. There is the possibility that a specific group is not possible to classify.

The fourth contribution that we can mention is the validation system process. As each of the blocks was extensively tested and validated, we can guarantee the functionality of each block independently. Since the flexibility between blocks is one of the differentials of this work, it is essential that if one block changes, the others remain stable. As we described the validation process in detail, it is possible to validate the block in the same way if someone reproduces the development process.

As a fifth contribution, we can cite the development of real-time detection. Real-time detection experiments showed promising results for participants in the first experiment, thus highlighting the need to evaluate a method to make classification more efficient for participants outside the group used in training.

The sixth contribution of the work is the comparative analysis of the results of clinical and non-clinical participants. We found that non-clinical participants present more periodic signals and similar behavior. On the other hand, non-clinical participants do not present well-defined behavior and may or may not have an increase in signals after the incidence of stress. Alternatively, clinical participants may not return to baseline after a stressful situation, as observed during the experiments. As highlighted in the text, this is due to disturbances in the participants' allostatic load.

Finally, we can cite the submitted papers. We submitted the paper entitled 'Hybrid Deep Learning Model for Pattern Detection - a Multiple Physiological Signal Approach using Wearable Sensors' to IEEE - BSN 2021, which performs a comparison between datasets using ML for classification. The paper 'SIF Dataset: A Dataset Based on Stress Induction' was submitted and accepted to LVI Congresso Anual da SBFI. This paper describes the dataset acquired during the stress induction experiments performed. We also submitted the papers entitled 'The Impact of Allostatic Load of Clinical Patients on Machine Learning models' to the International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) and 'ATHENA I - Architecture for Healthcare reinforced by Artificial Intelligence' to Frontiers of Information Technology and Electronic Engineering (FITEE). The first one relates the impact of allostatic load on physiological signals with the classification of signals through the use of machine learning algorithms; the second paper describes the developed architecture and its modules.

8.2 Future Work

The present work presents an architecture for acquiring, processing, storing, and classifying physiological signals with stress detection using Machine Learning algorithms. This architecture was named ATHENA I, an acronym for Architecture for Healthcare Reinforced by Artificial Intelligence. We found that the generated model presented a lower performance when classifying participants who were not within the dataset used in training during the final experiments carried out. The problem encountered was already expected since each individual has physiological signals with their characteristics, and their signals have different levels.

As a topic for future work, we intend to deepen our approaches to Machine Learning algorithms to develop a robust enough model that does not require an adjustment for each new participant. We may need to merge classification techniques or expand the dataset to encompass the broadest possible range of cases.

Another suggestion for future work is the implementation of the cloud storage and notification system. With the notification through a cloud system, it would be possible for any professional to be informed if the architecture identifies anomalies in the signals. For greater architecture robustness, we suggest developing a mobile application that could perform the interface between architecture, cloud, and user, allowing a more transparent use of the entire ecosystem.

Another possibility to expand the architecture is implementing a mobile application for the notification of detections on the cell phone. In addition, the mobile application would allow the interaction between the health professional and the participant, allowing the health professional to make recommendations and the participant to give feedback on how they are feeling.

Finally, another suggestion to expand the work would be to develop an architecture integration with more devices on the same network, which could be managed in a simplified way. Expanding the architecture to use multiple wearable sensors and SBCs would allow real applications with a more significant number of possibilities synchronously and without conflicts in data communication.

REFERENCES

AHUJA, R.; BANGA, A. Mental stress detection in university students using machine learning algorithms. **Procedia Computer Science**, [S.l.], v. 152, n. 2, p. 349 – 353, 2019.

AL-SHARGIE, F. et al. Towards multilevel mental stress assessment using svm with ecoc: an eeg approach. **Medical and Biological Engineering and Computing**, [S.l.], v. 56, p. 125 – 136, Oct. 2017.

ALI, F. et al. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. **Information Fusion**, [S.l.], v. 63, p. 208 – 222, Nov. 2020.

ALI, F. et al. An intelligent healthcare monitoring framework using wearable sensors and social networking data. **Future Generation Computer Systems**, [S.l.], v. 114, p. 23 – 43, Jan. 2021.

APPLE WATCH, I. **Apple watch**. Available in: <https://www.apple.com/br/watch/>. Access in: may. 2021.

BARRETT, K. E. et al. **Ganong's review of medical physiology**. Twenty-Third. ed. Oxford: The McGraw-Hill Companies, 2010. 727 p.

BISHOP, C. M. **Pattern recognition and machine learning**. 1. ed. [S.l.]: Springer, 2006.

BLIKSTEIN, P. Using learning analytics to assess students' behavior in open-ended programming tasks. **Proceedings of the 1st International Conference on Learning Analytics and Knowledge**, [S.l.], v. 11, p. 110 – 116, Feb. 2011.

BRINGAS, S. et al. Alzheimer's disease stage identification using deep learning models. **Journal of Biomedical Informatics**, [S.l.], v. 109, Sept. 2020.

CAN, Y. S.; ARNRICH, B.; ERSOYA, C. Stress detection in daily life scenarios using smart phones and wearable sensors: a survey. **Journal of Biomedical Informatics**, [S.l.], v. 92, Apr. 2019.

CARREIRAS, C. et al. **Biosppy**: biosignal processing in python. Available in: <https://github.com/PIA-Group/BioSPPy/>. Access in: may. 2021.

COHEN, S.; JANICKI-DEVERTS, D.; MILLER, G. E. Psychological stress and disease. **JAMA The Journal of the American Medical Association**, [S.l.], v. 298, n. 14, p. 1685 – 1687, Nov. 2007.

CORRIGAN, S. L. et al. Monitoring stress and allostatic load in first responders and tactical operators using heart rate variability: a systematic review. **BMC Public Health**, [S.l.], v. 21, Sept. 2021.

COSTA, C. A. d. et al. Psychological stress and disease. **Artificial Intelligence in Medicine**, [S.l.], v. 89, p. 61 – 69, July 2018.

DALE, N.; LEWIS, J. **Computer science illuminated**. 3. ed. Massachusetts, USA: Jones and Bartlett Publishers, 2007.

DAWSON, M. E.; SCHELL, A. M.; FILION, D. L. The electrodermal system. In: _____. **Handbook of psychophysiology**. [S.l.]: Cambridge University Press, 2007. p. 159 – 181.

DI MARTINO, B. et al. Internet of things reference architectures, security and interoperability: a survey. **Internet of Things**, [S.l.], v. 1, p. 99 – 112, Sept. 2018.

DISEASE CONTROL AND PREVENTION, U. Centers for. **Body mass index (bmi)**. Available in: <https://www.cdc.gov/healthyweight/assessing/bmi/>. Access in: may. 2021.

DUNN, J.; RUNGE, R.; SNYDER, M. Wearables and the medical revolution. **Personalized Medicine**, [S.l.], v. 15, n. 5, p. 429 – 448, Sept. 2018.

EDWARDS, A. L. et al. Application of real-time machine learning to myoelectric prosthesis control: a case series in adaptive switching. **Prosthetics and Orthotics International**, [S.l.], v. 40, n. 5, p. 573 – 581, July 2015.

ELENKO, E.; UNDERWOOD, L.; ZOHAR, D. Defining digital medicine. **Nature Biotechnology**, [S.l.], v. 33, p. 456 – 461, May 2015.

ELMALAKI, S. et al. Towards internet-of-things for wearable neurotechnology. **2021 22nd International Symposium on Quality Electronic Design (ISQED)**, [S.l.], May 2021.

FARIAS, S. M. d. C. et al. Caracterização dos sintomas físicos de estresse na equipe de pronto atendimento. **Revista da Escola de Enfermagem da USP**, São Paulo, v. 45, n. 3, June 2011.

FAUST, O. et al. Deep learning for healthcare applications based on physiological signals: a review. **Computer Methods and Programs in Biomedicine**, [S.l.], v. 161, p. 1 – 13, July 2018.

FELDMAN, J.; GOLDWASSER, G. P. Eletrocardiograma: recomendações para a sua interpretação. **Revista da SOCERJ**, São Paulo, v. 17, n. 4, p. 251 – 256, Nov. 2004.

FERREIRA, R. Y. K.; ATTROT, W.; SAKURAY, F. Extração de características de eletrocardiogramas. **Revista da Escola de Enfermagem da USP**, São Paulo, v. 45, n. 3, June 2011.

GAO, W.; BROOKS, G. A.; KLONOFF, D. C. Wearable physiological systems and technologies for metabolic monitoring. **Journal of Applied Physiology**, [S.l.], v. 124, n. 3, p. 548 – 556, Mar. 2018.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Porto Alegre, Brazil: UFRGS, 2009. 120 p.

GIBBS, B. B. et al. Definition, measurement, and health risks associated with sedentary behavior. **Medicine and Science in Sports and Exercise**, [S.l.], v. 47, n. 6, p. 1295 – 1300, June 2015.

GUIDI, J. et al. Allostatic load and its impact on health: a systematic review. **Psychother Psychosom**, [S.l.], v. 90, n. 1, p. 11 – 27, Aug. 2021.

HAMILTON, P. S. **Open source ecg analysis software documentation**. Limited. ed. Somerville, MA: [s.n.], 2002.

HAN, L. et al. Detecting work-related stress with a wearable device. **Computers in Industry**, [S.l.], v. 90, p. 42 – 49, 2017.

HAYKIN, S. **Neural networks and learning machines**. 3. ed. Ontario, Canada: Pearson - Prentice Hall, 2009.

HEATON, J. **Deep learning and neural networks**. 3. ed. [S.l.]: Heaton Research, Inc., 2015.

HEIDEL, A.; HAGIST, C. Potential benefits and risks resulting from the introduction of health apps and wearables into the german statutory health care system: scoping review. **JMIR Mhealth Uhealth 202**, [S.l.], v. 8, n. 9, p. 1735 – 1780, 2020.

JACOBSEN, M. et al. Wearable technology: a promising opportunity to improve inpatient psychiatry safety and outcomes. **Journal of Diabetes Science and Technology**, [S.l.], v. 15, n. 1, p. 34 – 43, Jan. 2021.

JOVANOV, E. et al. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. **Journal of NeuroEngineering and Rehabilitation**, [S.l.], v. 2, n. 6, Mar. 2005.

JOVANOVIC, P. et al. Applications of the single board computers in the software defined radio systems. **Sinteza 2014**, [S.l.], June 2014.

KAPLAN, A.; HAENLEIN, M. Siri, siri, in my hand: who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence. **Business Horizons**, [S.l.], v. 62, n. 1, p. 15–25, Feb. 2019.

KIM, N. H.; KO, J. S. Introduction of wearable device in cardiovascular field for monitoring arrhythmia. **Chonnam Medical Journal**, [S.l.], v. 57, n. 3, p. 1 – 6, Jan. 2021.

KNOPFEL, A. **Fundamental modeling concepts - fmc quick introduction**. Available in: <http://www.fmc-modeling.org/download/quick-intro/FMC-QuickIntroduction.pdf>. Access in: may. 2021.

KOCHE, J. C. **Fundamentos de metodologia científica** : teoria da ciência e iniciação à pesquisa. Petrópolis, Brazil: Editora Vozes, 2011.

KWON, Y.-H.; SHIN, S.-B.; KIM, S.-D. Electroencephalography based fusion two-dimensional (2d)-convolution neural networks (cnn) model for emotion recognition system. **Sensors**, [S.l.], v. 18, n. 5, p. 1383, May 2018.

LEGG, T. J. **Everything you need to know about stress**. Available in: <https://www.healthline.com/health/stress>. Access in: may. 2021.

LEMOYNE, R. et al. Implementation of machine learning for classifying prosthesis type through conventional gait analysis. **2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**, [S.l.], v. 37, Nov. 2015.

LI, R.; LIU, Z. Stress detection using deep neural networks. **BMC Medical Informatics and Decision Making**, [S.l.], v. 20, n. 11, Dec. 2020.

MERLETTI, R. et al. Technology and instrumentation for detection and conditioning of the surface electromyographic signal: state of the art. **Clinical Biomechanics**, [S.l.], v. 24, n. 2, p. 122 – 134, Feb. 2009.

MISHRA, T. et al. Pre-symptomatic detection of covid-19 from smartwatch data. **Nature Biomedical Engineering**, [S.l.], v. 4, p. 1208 – 1220, Nov. 2020.

MOODY, G. B.; MARK, R. G. The impact of the mit-bih arrhythmia database. **IEEE Engineering in Medicine and Biology Magazine**, [S.l.], v. 20, n. 3, p. 45 – 50, June 2015.

MOZOS, O. M. et al. Stress detection using wearable physiological and sociometric sensors. **International Journal of Neural Systems**, [S.l.], v. 27, n. 2, May 2017.

MURAT, F. et al. Application of deep learning techniques for heartbeats detection using ecg signals-analysis and review. **Computers in Biology and Medicine**, [S.l.], v. 120, May 2020.

NASIRI, N. Wearable technologies for healthcare monitoring. In: WEARABLE DEVICES - THE BIG WAVE OF INNOVATION, 2019. **Anais...** IntechOpen, 2019.

NISWAR, M. et al. A low cost wearable medical device for vital signs monitoring in low-resource settings. **International Journal of Electrical and Computer Engineering (IJECE)**, [S.l.], v. 9, n. 4, p. 2321–2327, Aug. 2019.

NOUR, B. et al. A survey of internet of things communication using icn: a use case perspective. **Computer Communications**, [S.l.], v. 143, p. 95–123, May 2019.

PLUX WIRELESS BIOSIGNALS, S. **Bitalino**. Available in: <https://bitalino.com>. Access in: may. 2021.

PLUX – WIRELESS BIOSIGNALS, S.A. **Electrocardiography (ecg) sensor data sheet**. Lisbon, 2020. Available in: <https://bitalino.com/storage/uploads/media/revolution-ecg-sensor-datasheet-revb-1.pdf>. Access in: may. 2021.

PLUX – WIRELESS BIOSIGNALS, S.A. **Electromyography (emg) sensor user manual**. Lisbon, 2020. Available in: <https://bitalino.com/storage/uploads/media/electromyography-emg-user-manual.pdf>. Access in: may. 2021.

PLUX – WIRELESS BIOSIGNALS, S.A. **Electrodermal activity (eda) sensor user manual**. Lisbon, 2020. Available in: <https://bitalino.com/storage/uploads/media/electrodermal-activity-eda-user-manual.pdf>. Access in: may. 2021.

PLUX – WIRELESS BIOSIGNALS, S.A. **Respiration (pzt) sensor data sheet**. Lisbon, 2020. Available in: <https://bitalino.com/storage/uploads/media/pzt-sensor-datasheet-revb.pdf>. Access in: may. 2021.

POLAR. **Polar**. Available in: <https://www.polar.com/us-en>. Access in: may. 2021.

PONCIANO, V. et al. Detection of diseases based on electrocardiography and electroencephalography signals embedded in different devices: an exploratory study. **Brazilian Journal of Development**, [S.l.], v. 6, n. 5, p. 27212 – 27231, May 2020.

RASPBERRY PI FOUNDATION, U. **Raspberry pi**. Available in: <https://www.raspberrypi.org>. Access in: may. 2021.

REIS, H. J. L. et al. **Manual prático de eletrocardiograma**. 1. ed. ão Paulo, Brazil: Atheneu, 2013. 121 p.

RIENZO, M. D.; MUKKAMALA, R. Wearable and nearable biosensors and systems for healthcare. **Sensors**, [S.l.], v. 21, n. 4, Feb. 2021.

RODRIGUES, C. et al. Evaluating a new approach to data fusion in wearable physiological sensors for stress monitoring. In: ______. **Bracis 2020**: intelligent systems. [S.l.]: Springer, 2020. p. 544 – 557.

RUSSELL, S.; NORVIG, P. **Artificial intelligence**: a modern approach. 3. ed. [S.l.]: Pearson - Prentice Hall, 2009.

SALEHZADEH, A.; CALITZ, A. P.; GREYLING, J. Human activity recognition using deep electroencephalography learning. **Biomedical Signal Processing and Control**, [S.l.], v. 62, Sept. 2020.

SANTAMARIA-GRANADOS, L. et al. Using deep convolutional neural network for emotion detection on a physiological signals dataset (amigos). **IEEE Access**, [S.l.], v. 7, p. 50 – 67, Nov. 2018.

SAP AG, I. **Standardized technical architecture modeling conceptual and design level**. Available in: <http://www.fmc-modeling.org/download/fmc-andtam/SAP-TAM_Standard.pdf>. Access in: may. 2021.

SCHMIDT, P. et al. Introducing wesad, a multimodal dataset for wearable stress and affect detection. **ICMI '18: Proceedings of the 20th ACM International Conference on Multimodal Interaction**, [S.l.], v. 7, p. 400 – 408, Oct. 2018.

SCHOMER, D. L.; SILVA, F. H. L. d. **Niedermeyer's electroencephalography**: basic principles, clinical applications, and related fields. 7. ed. Oxford: Oxford University Press, 2017.

SETHI, P.; SARANGI, S. R. Internet of things: architectures, protocols, and applications. **Journal of Electrical and Computer Engineering 2017**, [S.l.], v. 1, p. 1 – 25, Jan. 2017.

SHARMA, M.; KACKER, S.; SHARMA, M. A brief introduction and review on galvanic skin response. **International Journal of Medical Research Professionals**, [S.l.], v. 2, n. 6, p. 13 – 17, Oct. 2016.

SREENILAYAM, S. P. et al. Advanced materials of printed wearables for physiological parameter monitoring. **Materials Today**, [S.l.], v. 32, p. 147 – 177, Feb. 2020.

STEVENS, W. R.; FENNER, B.; RUDOFF, A. M. **Unix network programming**: the sockets networking api. 3. ed. Ontario, Canada: Addison Wesley, 2003.

TEXAS INSTRUMENTS, I. **Beagleboard**. Available in: <https://beagleboard.org>. Access in: may. 2021.

TRIPATHI, S. et al. Using deep and convolutional neural networks for accurate emotion classification on deap dataset. **AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence**, [S.l.], v. 17, p. 4746 – 4752, Feb. 2018.

WANG, K.; GUO, P. An ensemble classification model with unsupervised representation learning for driving stress recognition using physiological signals. **Materials Today**, [S.l.], v. 18, p. 1 – 13, Mar. 2020.

WAZLAWICK, R. S. **Metodologia de pesquisa para ciência da computação**. 2. ed. Rio de Janeiro, Brazil: Elsevier, 2014.

WILSON, S.; LAING, R. M. Wearable technology: present and future. **91st World Conference of The Textile Institute**, Leeds, UK, July 2018.

YANG, J. et al. Smart wearable monitoring system based on multi-type sensors for motion recognition. **Smart Materials and Structures**, [S.l.], v. 30, n. 3, p. 59 – 68, Feb. 2021.