Department of Computer Architecture and Technology
Konputagailuen Arkitektura eta Teknologia saila
Departamento de Arquitectura y Tecnología de Computadores

Universidad del País Vasco
Euskal Herriko Unibertsitatea

University of the Basque Country

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Contributions to physiological computing by means of automatic learning

Ph.D. Dissertation presented by
Asier Salazar Ramírez

Supervised by
Andoni Arruti Ilarramendi
Jose Ignacio Martín Aramburu

Donostia, March 2020

Department of Computer Architecture and Technology
Konputagailuen Arkitektura eta Teknologia saila
Departamento de Arquitectura y Tecnología de Computadores

Universidad del País Vasco
Euskal Herriko Unibertsitatea

University of the Basque Country

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Contributions to physiological computing by means of automatic learning

Ph.D. Dissertation presented by
Asier Salazar Ramírez

Supervised by
Andoni Arruti Ilarramendi
Jose Ignacio Martín Aramburu

Donostia, March 2020

This work was partially supported by the ERDF / Spanish Ministry of Science, Innovation and Universities - National Research Agency / PhysComp project, TIN2017-85409-P and by the Department of Education, Universities and Research of the Basque Government (ADIAN research group, grant IT980-16).

"Intelligence is an accident of evolution,
and not necessarily an advantage."

Isaac Asimov

Abstract

Since the last decades the world's population is suffering a shift in the age distribution of its demography. There is no doubt that this change in the demographic distribution is going to have a great impact on various aspects of society, as are, for example, education, the size and productivity of the labour force and public health. In fact, it is already possible to observe certain consequences of this phenomenon, as it can be seen, for instance, in the debate around the quantity and sustainability of the pensions. In this way, it is clear that, in the coming decades, our society is going to face serious challenges towards maintaining the life-quality and autonomy levels that today's elderly population is having.

Taking this premise as a base, this doctoral thesis intends to do its little contribution to the state of the art of physiological computing. This discipline is found in an intermediate point between medicine and engineering and its objective is to enable the automatic processing of the changes in the physiological signals of the human body. This way, the final objective of applying computation techniques in the world of physiology is not other than being able to give a technological solution to the diverse problems that people with special needs and their caregivers face in their everyday life.

In this sense, the work presented in this thesis works on two main lines. The first line is focused on the computation of different variables of the body to detect those situations in which an individual could be getting involved in a situation of maintained stress. Besides, this line also develops its work in the opposite direction to the detection of stress, the detection of relaxation. After all, the detection of the changes that relaxation produces in the body can be useful in the area of preventive medicine and, as such, the new tendencies in modern medicine highlight the importance of both physical and mental well-being for avoiding the appearance of certain diseases. Being able to automatically detect when and how a subject is getting relaxed will undoubtedly make it easier to know which relaxation techniques fit best and bring better results to that person.

On the other hand, the other research line present in this thesis focuses on the search for alternative ways to interact with machines beyond the use of the mouse or the keyboard. More precisely, this second line gets into the world of Brain-Computer Interfaces or, as is best known in the literature, BCI. In this context, this document presents the work developed in the field of the classification of encephalographic patterns for imaginary movement problems. The work related to this subject focused on maximising the accuracy of motor im-

agery movement classification while maintaining the number of false movement intentionality detections within acceptable values.

Laburpena

Azkenengo hamarkadetatik hona munduko biztanleria zahartze prozesu nabari bat sufritzen ari da. Zalantza barik, distribuzio demografikoan ematen ari den aldaketa honek eragin handia izango du sozietateko zenbait esparrutan, hara non, hezkuntzan, biztanleria aktiboaren tamainan eta bere produktibitatean edota osasun publikoan. Izan ere, bada posible zenbait ondorio aurkitzea, adibidez, pentsioen kopuruaren eta iraunkortasunaren inguruan egunero ikusten dugun tirabiretan. Hortaz, argi dago datorren hamarkadetan biztanleriak erronka sozial eta teknologiko handiak aurrean izango dituela gaur egungo hirugarren adinekoek disfrutatzen dituzten bizi kalitate eta autonomia mailak mantentzeko.

Premisa hau oinarritzat hartuta, tesi honek bere ekarpen txikia egin nahi dio konputazio fisiologikoko esparruari. Diziplina hau medikuntzako eta injinerutzako munduen artean aurkitzen da eta giza gorputzeko seinale fisiologikoetan ematen diren aldaketak automatikoki prozesatzea du helburu. Hau honela izanda, fisiologiako munduari konputazio teknikak aplikatzearen xede nagusia behar bereziak dituztenek eta euren begiraleek egunero aurkitzen dituzten arazoei irtenbide teknologiko bat ematea da.

Zentzu honetan, tesi honen bidez aurkezten ari den lan honek bi ildo nagusi jorratuko ditu. Lehenengo ikerketa lerroak gorputzeko aldagai ezberdinen konputazioa du ardatz bezala pertsona bat luzaroan mantendutako estres egoera batean sartzen ari den detektatzeko. Era berean, ikerketa lerro honek estresaren detekzioaren kontrakoa ere lantzen du, hara non, erlaxazioaren detekzioa. Azken finean, erlaxazioaren eraginez gorputzean ematen diren aldaketak detektatzea baliogarria izan daiteke medikuntza prebentiborako, eta, horrela izanda, medikuntzako tendentzia berriek ongizate fisiko eta psikologikoaren garrantzia nabarmentzen dute zenbait gaixotasunen agerpena ekiditeko. Zalantza barik, noiz erlaxatzen garen eta nola egiten dugun detektatzeak erabat erraztuko du norberari hobekien egokitzen zaizkion eta onura gehien egiten dizkion erlaxazio teknikak aurkitzearen bidea.

Bestalde, tesi honetan ageri den beste ikerketa lerroak makinekin interakzionatzeko bide alternatiboak bilatzearen ildotik jarraitzen du, arratoiaren eta teklatuaren erabilera alde batera uzten duena. Zehazki, bigarren lerro hau Burmuin Ordenagailu Interfazearen munduan barneratzen da. Mundu honi Literaturan Brain-Computer Interface bezala ezagutzen zaio (edo, are sinpleago, BCI siglen bitartez). Testuinguru honetan, dokumentu honek imajinazio mo-

torrerako patroi entzefalografikoen klasifikazioan egindako lana aurkeztuko du. Gai honen inguruan egindako lanak imajinazio motorreko mugimenduak klasifikatzerakoan ahal bezain zehaztasun handiena lortzea bilatzen du, detekzio faltsuen kopurua tarte onargarri baten barruan mantenduz.

Resumen

Durante las últimas décadas la población mundial ha estado sufriendo un claro envejecimiento en su distribución demográfica. Sin duda, este cambio en la distribución de la demografía traerá consigo un gran impacto en varios aspectos de la sociedad, como por ejemplo, la educación, el tamaño de fuerza laboral y su productividad o la sanidad pública. De hecho, ya es posible observar ciertas consecuencias en el día a día, como se puede ver en el cada vez más común debate alrededor de la cuantía y sostenibilidad de las pensiones. De esta manera, queda patente que en las décadas venideras la sociedad se enfrentará a importantes retos sociales y tecnológicos y cuyo objetivo será el mantenimiento del nivel de calidad de vida y autonomía de los cuales disfruta la actual tercera edad.

Tomando esta premisa como base, esta tesis doctoral pretende aportar su granito de arena al estado del arte de la computación fisiológica. Esta disciplina se encuentra en un punto intermedio entre los mundos de la medicina y de la ingeniería y tiene como objetivo posibilitar el procesamiento automático de los cambios en las distintas señales fisiológicas del cuerpo humano. De esta manera, el objetivo final de la aplicación de las técnicas de la computación al mundo de la fisiología no es otro que poder dar solución tecnológica a los diversos problemas a los que se enfrentan las personas con discapacidades especiales y sus cuidadores.

En este aspecto, el trabajo presentado en esta tesis trabajará dos líneas principales. La primera línea se centra en la computación de diferentes variables corporales para detectar las situaciones en las que una persona se esté viendo implicada en un estrés continuado. Así mismo, la línea también desarrolla el aspecto opuesto a detectar estrés, esto es, la detección de la relajación. Después de todo, la detección de los cambios producidos por la relajación en el cuerpo puede ser útil en el área de la medicina preventiva y como tal, las nuevas tendencias de la medicina moderna están haciendo especial hincapié en la importancia del bienestar corporal y mental para prevenir la aparición de ciertas enfermedades. Discernir de manera automática cuándo nos relajamos y cómo lo hacemos, sin duda alguna allanará el camino para que cada uno halle las técnicas de relajación que mejor se adapten a su ser y que mayor beneficio le aporten.

Por otra parte, la otra línea presente en esta tesis se centra en el ámbito de la búsqueda de formas alternativas de interacción con las máquinas más allá del uso del teclado o del ratón. Más concretamente, esta segunda línea se adentra en

el mundo del Interfaz Cerebro Computadora, más conocido en la literatura como Brain-Computer Interface o simplemente por sus siglas, BCI. En este contexto, este documento presentará el trabajo realizado en el ámbito de la clasificación de patrones encefalográficos en problemas de imaginación motora. El trabajo realizado sobre este tema se centra en obtener la máxima precisión posible en la detecccion de movimientos imaginarios, intentando al mismo tiempo mantener la cantidad de falsas detecciones de movimientos intencionados dentro de márgenes aceptables.

Acknowledgements

I want to thank you, reader, for reaching this point. Either if you are another PhD student or not, if you have this document in your hands it means that you have some kind of interest in this area. I hope you enjoy the thesis and that you do not want to kill me after reading it. Good luck ;)

- First of all, I would like to thank my thesis directors and tutor for their enormous help. They have guided me throughout this whole thesis period and, for sure, if it was not for their help, this work would have been much harder for me. Thank you, Natxo, Andoni and Javi.
- Then I want to thank Aldapa and Adian. People here are so nice. Who would tell a guy from Bilbao that he would feel so good in Donostia!
- I want to give thanks to the jury of this thesis that has made the effort to be there for me and accepting to come so quickly.
- Also, I have to give special thanks to Raquel, who always believed in me. Even when I decided not to research any more she kept on sending me things to work on. She was who found the possibility for me to be a part of Aldapa research team. Saying thank you is not enough in exchange for what you have done for me.
- Then, I want to say thank you to the people in Lisbon who were so welcoming. Thanks: Hugo, Ana, Patricia, Mariana, Andre, Joana, Agaph... and to the Erasmus people I met there. Thank you also to Joao for hosting me in Castelo Branco.
- Next, I would like to thank Eloy, Unai and Mikel from GICI research team for still considering me a part of their team. You are nice, guys.
- I want to give thanks to Ainhoa Yera and Ainhoa Etxautz for their help in the last desperate moments before the deposit of this thesis.
- My music band colleagues also deserve a mention for being understanding. At last, I will get back to the rehearsals. Raust!
- Thanks also to all my friends for their interest regarding my thesis and their help offers. Thank you also to Txarraska Gaztetxea, whose people is like a family to me. Long live.

Thank you to my relatives. They have always believed that I could do this and they have always pushed me forward.

Thank you, Maria. You have been my psychological support. Living with you I feel better and safer.

Thank you, Jon, my dear brother. Being with you always makes me laugh.

Thanks, dad. For your infinite patience. For cooking for me. For believing in me. For becoming both a father and a mother. For everything. You are the greatest.

And finally, most especially... Thank you, mom. I miss you. This one is for you.

Contents

Part I

Introduction

Chapter 1

Introduction to physiological computing

Electronics and, especially computer devices, have vastly developed during the last decades. From computers that occupied whole rooms to nowadays' small smart telephones, the computational power of computer devices has grown exponentially and so, it is now possible to implement complex calculations and algorithms which could not be thought of in the last century. Accordingly, the use of computers has long ago left the experimental environment of the labs and has totally become a part of our daily life, to the point that losing connection to the internet might translate into important economic loses for some people.

This technological revolution has had impact in every professional discipline, not only in the more technical areas, but also in those disciplines in which the human being catches all the attention, such as medicine or psychology. In this sense, it is common to see how clinicians collect signals coming from their patients' bodies. These signals permit doctors, nurses and other professionals to assess the health of their patients. Furthermore, technologies related to human signal analysis have broken the economical and complexity barriers and have reached even further than expected. For instance, it is getting common for young diabetic patients to constantly have a glucose sensor in their arms which connects to their smartphones and calculates the proper quantity of insulin that they have to inject.

However, the expansion of technology and computation techniques has not yet covered all human-related and social needs. In this sense, there is still a whole span of problems that await for a technological solution and, this way, improve the life-quality of a part of the society. For instance, many researchers study neural activity and brain-computer interfaces to find systems that help people with neurodegenerative diseases interact with their environment [1]. Also, a lot could be achieved towards better understanding and interacting with people with cognitive disabilities if a way to determine how these people feel was found. Related to it, some researchers consider the possibility to

detect emotions and feelings lies in the automatic analysis of the physiological signals with computation techniques.

Sometimes, the application of computational and other engineering techniques does not cross the frontiers of the laboratory. A reason for this might be that the problems to solve are complex and so, the results of the solutions are still far from perfect. As stated in [2], a human-machine interface has to be effective, efficient, safe and easy to learn and to remember. Anyway, most of the physiological computing solutions that are proposed in the literature imply using sensors, tools and algorithms that are uncommon to the normal user and so, they are cannot be applied to real-life situations. Hence, research must be done to improve the already existing solutions or even creating new proposals for solving problems that have remained untreated. Taking this statement as a motivation, the work presented in this thesis was mainly focused on improving and giving solution to two specific problems related to physiological computing.

The first contribution presented in this thesis is related to the detection and estimation of human mental stress. Stress is a very common syndrome that every day affects people of all genders and ages, producing both physical and psychological symptoms. Despite it is a common health problem, it is still complex for researchers to estimate its level using automatic methods. Accordingly, this first contribution proposes a methodology to give a continuous estimation of the stress level robust to artefacts. First, it presents an automatic method that detects R peaks in electrocardiograms, rejecting offset noises and artefacts and producing trustworthy heart-rate and heart-rate variability signals. Besides, the contribution proposes to extract a new feature out of the respiration signal. The values of this feature vary depending on whether a person is under stressful conditions or not and so, it is a good candidate to be used for assessing stress. Last, the contribution presents a modification of an already existing fuzzy-logic system for estimating the transitions between non-stressed mental states and stressed mental states in a less aggressive way than in the original system. This was done by adding new membership functions to the system as well as making a simple yet logic inference rule system. In summary, this contribution concluded in a fuzzy-logic system for continuously assessing mental stress that is less reactive to sudden physiological signal changes and noises.

As a continuation of the subject related to stress, the second contribution is centred towards the detection of the opposite mental phenomenon, which is no other than relaxation. The study of relaxation from a technical perspective is a subject that has not been commonly studied in the literature, and, when done, it has was mostly linked to the study of stress rather than trying to automatically detect it. Anyway, the World Health Organization states that being healthy implies not only not having illnesses [3], but also having both mental and physical well-being. In these fast-living times, achieving relaxation is a key aspect to have a good mental health. Thus, seeing that there was a lack of studies related to this theme, in this part of the thesis, the team collaborated with a team of psychologists and physiologists to study the variations that occur in the physiology when relaxation is taking place. This study focuses on those physiological changes happening either during a stressful situation or when us-

ing relaxation techniques in a less aggressive situation. The team analysed how to detect these changes using only the electro-dermal activity signal of sweating. From this signal, they ended up extracting two features that are not computationally costly. Later, these features were used with different machine learning techniques to detect the presence of those physiological variations meaningful for relaxation and to classify their intensity level.

Finally, the last contribution of this thesis targeted a different area of study. It is related to finding alternative human-machine or human-computer interaction systems. More precisely, it focuses on motor imagery brain-computer interface (BCI) systems, which are those systems that try to detect the brain activity signal patterns that happen the user thinks of moving a specific part of the body. The operation principle is simple. First, the computer learns which are the patterns that repeat when the user thinks of a movement. Then, once learnt, whenever a new pattern is detected, the machine gives user-defined commands to an actuator that will interact with the environment. Thus, BCI systems allow the user interact with the environment without muscular activity, which has applications not only for assisting people with special needs but also for industrial or commercial purposes, such as in the video-game industry. For this contribution, the team took a well-known dataset in the literature which proposes the classification problem of differentiating among thinking of moving four parts of the body: right hand, left hand, tongue and feet. Also, as these systems have to remain still when the user is not thinking of moving, the team added an extra type of pattern to the problem: the absence of movement intentionality. Then, the team tried to solve the problem with different machine learning solutions: first, only using supervised learning algorithms in different system topologies; second, using a hierarchical classification topology that combined both clustering and supervised learning algorithms. Finally, as this second proposal obtained satisfactory results, it was compared to other solutions in the literature and it got good comparative results.

1.1 Organization of this dissertation

Apart from the part corresponding to this introduction, this thesis has been divided into three more parts. Part II will present the key aspects related to anatomy, physiology, physiological computing and machine learning that are necessary to later understand the contributions that will be presented in this thesis. It will begin with Chapter 2, which focuses on explaining the key anatomy and physiology principles that rule the operation of the organ systems involved in the analyses of this work. Then, Chapter 3 will be centred in presenting the signals that measure the different variables of the human body that are related to the studies done in this thesis. This chapter will also give a review of the features that are commonly extracted from those signals.

After Part II, the document will continue with the part related to the contributions of this thesis to the literature: Part III. This part has been divided into three chapters, one for each of the main contribution works of this thesis.

First, Chapter 4 will present the work done towards creating a methodology that permits to estimate the user's mental stress level in a continuous manner. Second, the opposite reaction to stress will be studied in Chapter 5: relaxation. This chapter will focus on the study and classification of the physiological reaction patterns that are indicative of human relaxation. Finally, Part III will conclude with Chapter 6, in which the thesis will explain contributions done to motor imagery brain-computer interfaces.

Last, in Part IV, Chapter 7, this document will present the main conclusions reached by the author during his PhD thesis period. Besides, this chapter will also explain the future lines that opened while researching on the different subjects of this thesis. Finally, Chapter 7 will also close the thesis with the listing of the references of the works that support this thesis in the literature.

Part II

Background: Human anatomy, physiology and related computing

Chapter 2

Basic human anatomy and physiology

2.1 Introduction

Several scientific disciplines focus on the study of the body. One of the most well-known is anatomy, which focuses on the structure of the body itself. Possibly, anatomy is the oldest of the disciplines in medicine and it was developed even during the ancient times by looking at the wounds of the soldiers damaged in combat. Later, science got into more modern times, the physicians started also to analyse the inner composition of the human body thanks to dissections. At the beginning, dissection was forbidden but with the time physicians were allowed to practice it in academic environments. Nowadays, it is a commonly practised technique, for instance, for analysing in the structure of cancerous tumour. Depending on the focus of the research, anatomy splits in several branches, such as, gross-anatomy (the study of the large structures that can be seen without microscope), microscopic anatomy (the study of the morphology of cells and tissues) or regional anatomy (the study interactions between the structures within a same region of the body such as the head), among others.

Whereas anatomy centres its attention to the study of structures, another discipline is more focused on the study of the functions of the different parts of the body. The name of that discipline is physiology. Physiologists study the functions of the body at all the scales, from a bigger perspective such as studying whole organs, to the analysis of the smaller phenomena reaching even molecular level. As it happens in anatomy, physiology also divides into several different branches depending on the field of study. For instance, neurophysiologists focus on the study of the nervous system which is comprised of the brain, the spinal cord, the nerves... Nevertheless, most of the content treated in this dissertation is related to the branch of electrophysiology, which studies the electrical properties of the components of the body at different scales, from molecular and cellular scale to whole systems of the organism.

After having introduced these two big disciplines within the study of the human body, in the following Section 2.2 this thesis will give the reader further information about certain organs and functions of the body. Although these contents might seem unrelated to computing disciplines, the work presented in this thesis focuses in automatically processing, detecting or classifying some of the phenomena that take place in the human body. Thus, it has been considered to be interesting (and also necessary) to give some basic anatomy and physiology contents and, this way, help the reader understand some of the phenomena that will be later studied with computational methods.

2.2 The organ systems

There are several ways in which it is possible to analyse the human body. Depending on the scales of the components and the functions analysed, the structure of the body can be analysed from the smallest components, cells, for instance, to the whole organism. Therefore, it is possible to organise the body following a down to top criteria, in which atoms and molecules are the smallest part of the body and the organism is the whole. Following these criteria, molecules combine to form cells, and, at the same time, cells group with other cells creating different types of tissues. Then, every anatomically different combination involving two or more types of tissues is said to form an organ. Each organ is responsible for one or more specific physiological task and the combinations of those organs related to the same major physiological functions create an organ system. Finally, the organism is found at the top level of this classification, which is the sum of all the organ systems.

As analysing the organisation of the body in all these previously mentioned levels is a too extensive task, this thesis will remain mostly at the organ system level. There are eleven organ systems in the human body: the muscular, the skeletal, the immune, the endocrine, the urinary/excretory, the reproductive, the digestive, the integumentary, the nervous, the circulatory and the respiratory systems.

However, this thesis will only focus in the last four, in the following Subsections 2.2.1, 2.2.2, 2.2.3 and 2.2.4. It is important to consider that although the content in this work is closely related to the medical world, the scope of this thesis belongs to the background of informatics. Thus, those readers experienced in healthcare-related fields might find the content of the following sections to be slightly vague. This document will only discuss those aspects related to the previously mentioned organ systems that the author considers to be relevant for understanding the contributions that will be explained in part III. Most of the contents of this chapter have been extracted from [4] and from [5], being both sources very extended books with valuable information if the reader desires to expand what has been mentioned in this document. Also, unless otherwise stated, all the figures of this chapter have been extracted from [4].

2.2.1 The integumentary system

The integumentary system is the system that covers the skin, the hair and the nails. Concerning the hair, it has the function of protecting the body from UV radiation and it is also useful for the body's thermal regulation as it helps warm air to be kept close to the skin preventing the body from getting cold. Talking about nails, they play a protective role by hardening the end of both fingers and toes and enabling people to manipulate and scrape objects.

However, for the concerns of this thesis, the most interesting part of the integumentary system is the skin. The skin is the body's largest organ, weighing approximately 16% of the body's total weight. Apart from giving us the physical appearance, the skin and its accessory elements are responsible for several other vital functions, such as protection the inner organs of the body or giving us the capacity of sensing (the skin is full of nerve fibres connected to the brain as it will be further explained in Subsection 2.2.2), among others.

According to its structure, the skin can be divided into three main layers (see Figure 2.1): the epidermis, the dermis and the hypodermis. Being the epidermis the outer layer and the hypodermis the inner layer of the skin, the dermis is the layer trapped between the other two and it is the thickest of them all.

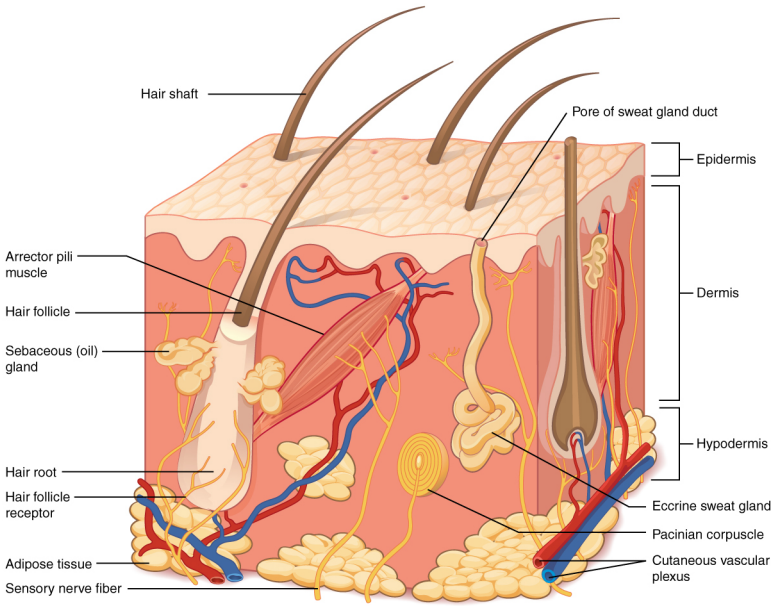

Figure 2.1: The three layers of the skin: epidermis, dermis and hypodermis.

As shown in Figure 2.1, the epidermis is the thinnest part of the skin (it is around 0.1mm thick) and it covers almost all the surface of the body. There are no blood vessels in the epidermis and its main role is to protect the layer

below, the dermis. Keratinocytes are the most common type of cells in the epidermis (around %90 of all the cells) and they are, with other types of cells, responsible for that task as they are specialised in producing and accumulating keratin, which is a protein that gives keratinocytes the property of being hard, scaly, tough and water-resistant.

There are other cells in the epidermis such as melanocytes, Langerhans cells and Merkel cells. The first ones are in charge of avoiding the skin from getting sunburn and they protect it from UV radiation. The seconds are responsible for detecting and fighting pathogens trying to enter the body through the skin. Finally, the third ones are the less common and they are those that are connected to the nerves in the dermis and give the body the capability to sense light contacts with the skin.

Continuing with the structure of the skin, the layer under the epidermis is the dermis. The dermis is the thickest part of the skin and it is composed of tissues, nerves and blood vessels. Being full of blood vessels, it is the layer responsible for giving the nutrients and oxygen to the cells of the epidermis. Besides, the connection of the sensing cells of the epidermis is done by the nerves in the dermis and they are used for sensing touch (pressure), pain and temperature. In addition, the deepest part of the dermis is mainly composed of connected tissues that have collagen and elastic fibres. Thanks to these tissues the skin is strong and elastic at the same time.

Apart from the above-mentioned elements within the dermis, sudoriferous glands are also found in the dermis (they are also known as sweat glands) and they deserve a special mention as they are strongly related to some of the contributions of this dissertation. For the moment, this explanation will limit to just the anatomical and physiological aspects of sweat glands and it will be in part III when the reader will be able to see the connection between sweat glands and computational techniques.

There are two types of sweat glands: the eccrine (see Figure 2.1) and the apocrine. On the one side, the eccrine glands are all over the body surface (especially in the hand palms, feet soles and forehead) and they are the ones in charge of producing the sweat for thermoregulation. On the other side, the apocrine are located in densely haired areas and they eject sweat under nervous and hormonal control. Thus, these glands are also related to the pheromone response of the body.

Finally, the last layer of the skin is the so-called hypodermis. The role of this layer is acting as a connector between the skin and the muscles underneath. This layer also plays the role of a warehouse; the fats of the body are normally stored in this layer, which also helps the body to keep the heat produced by the muscles under the skin.

To finish with the integumentary, it is important to recall all the functions it develops. First of all, it has a protective role against physical attacks, pathogens and radiation. In addition, it has a major role in the sensory of the body as it enables touch, pain and temperature feeling. Finally, thanks to its close relationship with the sympathetic nervous system (it is continuously monitoring skin temperature and giving orders to respond accordingly), the integumentary

system helps in the thermal regulation of the body through sweat ejection.

2.2.2 The nervous system

The nervous system is another organ system within the human body. This one is a very complex organ system which is in charge of very different functions. Indeed, the nervous system could be said to be composed of the brain, the spinal cord and nerve ramifications. However, the brain is such a complex organ that it could be said to be a lot of same-looking organs but, although they look the same from a macroscopic perspective, each of them have very different functions.

Through the following sub-subsection this document will give a brief explanation of the distribution and the different approaches to organise the nervous system. Later, the autonomic nervous system will be further explained in the next sub-subsection as it is the part of the nervous system more related to the work that was developed during the research period covered in this thesis.

Organising the structure of the nervous system

Anatomically talking in a very simplistic way, the nervous system could be divided into two parts (see Figure 2.2): the central nervous system (CNS, related to the brain and spinal cord) and the peripheral nervous system (nerves and ganglia). It is important to note that this division is not entirely true as some elements corresponding to the peripheral nervous system that are located in the brain itself. However, it is an easy approach to understand and useful for what this explanatory chapter is intended for.

Besides, it is also possible to classify the nervous system from a functional perspective. Again, trying to keep it as simple as possible, there are two ways to classify the functions of the nervous system. The first approach classifies the parts of the nervous system regarding the basic functionalities each element has. The second approach is related to the type of control of the body, that can be either somatic or autonomic.

Approaching the nervous system from the perspective of the basic functionalities, there are three main basic functions: sensing, responding and integrating functions. The sensing function is focused on getting information from both inside and outside the body so that any change in the balance of the physiological variables or the environment is registered. As said, stimuli can come from outside the body as, for instance, a bad taste from a rotten fruit; or from inside the body, as feeling hunger due to the increase of certain hormones in the organism. Then, the responding function is the one in charge of producing a reaction to the sensed stimulus. The reactions to stimuli can broadly vary, from a muscular movement (that can be either skeletal, a smooth muscle in the intestines or a cardiac muscle in the heart) to activating the previously mentioned eccrine or apocrine glands to eject sweat, among others. Finally, the last basic function, the integrating function, is the one in charge of relating the information obtained from stimuli and reactions to previously obtained information from other phenomena. This function allows to stablish correlation with previous

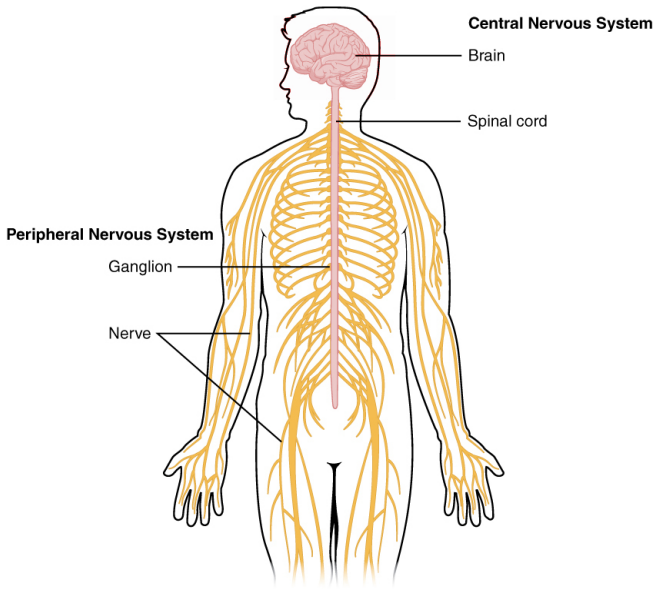

Figure 2.2: Simplistic anatomical division of the nervous system: the central nervous system (in pink) and the peripheral nervous system (in yellow).

experience and memories so that the person can respond better in future situations. Therefore, the integration function is related to high cognitive processes such as learning, memories and emotions.

Apart from dividing the nervous system according to the basic functionalities, there is another approach which classifies it according to how the control of the body is done. There are two main divisions of the nervous system depending on how the body does control the responses: the somatic nervous system and the autonomic nervous system (ANS from now on). The first one controls the responses that are done voluntarily, such as moving an arm. The somatic nervous system also controls certain responses although they might not be voluntary as it happens with reflex movements or with breathing. On the other side, the ANS is in charge of the involuntary control of the body. The ANS regulates the responses of the organism so that the physiological variables get to a certain balance at which the body works correctly. This self-regulating process is also known as homeostasis. Thus, the ANS is mostly related to homeostatic control of the body as sweat ejection for thermal regulation or cardiac pace acceleration.

In addition to the somatic nervous system and the ANS, there is one more system which is known as the enteric nervous system and it is responsible for the control of the smooth muscles and glands of the digestive system. This system is only dependent on the peripheral nervous system and sometimes it can be considered to be part of the ANS (not always). However, neither the

somatic nor the enteric nervous system have big importance to the technical development presented in this thesis; it will mainly focus on the responses of the ANS.

As a summary, the following Figure 2.3 can be useful to graphically see the interrelationship between all those divisions of the nervous system mentioned so far in this subsection.

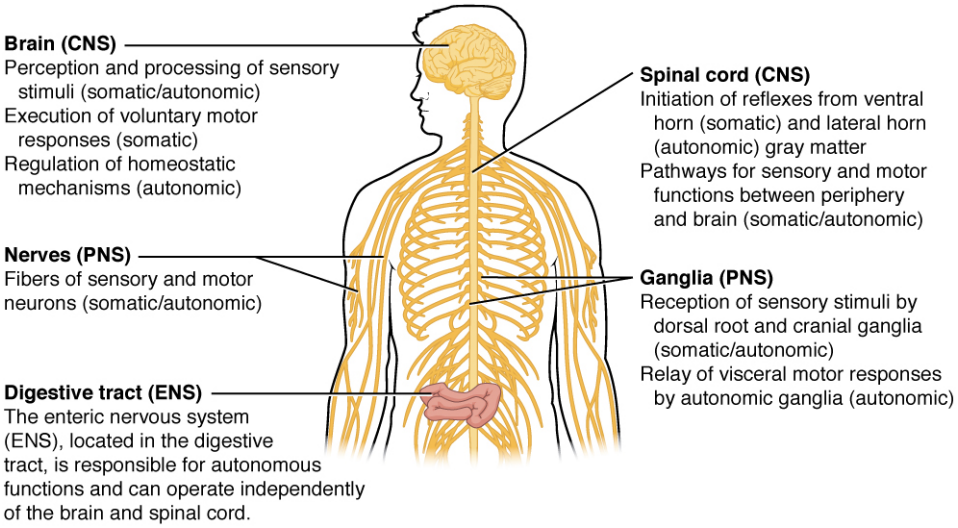

Figure 2.3: Generalised view of the locations and functions related to the different parts of the nervous system depending on the chosen organisation approach. In this figure, PNS stands for the peripeheral nervous system (to avoid the confusion with the parasympathetic nervous system, explained later).

The autonomic nervous system (ANS)

In nature, the expression "fight or flight" clearly defines the reactions of preys towards a threat coming from their predator: should they prepare to run or will they face their enemy and fight? This expression was first introduced by Walter Cannon [6] establishing parallelism between the reactions of humans and other animals in the wild. Although humans are no longer threatened by predators (there can be exceptions), reacting in such ways is still printed in our gene-code. Thus, when feeling threatened the cardiac pace of people tends to increase, adrenaline prepares the body for alert and many similar reactions take place in the organism. If analysed from a functional perspective, it is the autonomic nervous system (ANS) the one in charge of creating those changes in the organism.

However, there are also situations in which the body responses in the oppo-

site direction. For example, after having run to get on a bus that was already departing, the response of the body is to get back to calm and so the heart beating pace reduces, sweating decreases and respiration slows down. These kind of responses, which are more closely related to the expression of the wild "rest and digest", are also controlled by the ANS.

Besides, apart from "fight or flight" and "rest and digest" situations, the ANS also reacts to changes in emotional states. Mainly, this is due because most of the homeostatic functionalities of both the endocrine and ANS systems are controlled in the hypothalamus (a specific part of the brain, see Figure 2.4 obtained from [7]) and, respectively, the hypothalamus is highly connected to the amygdala, which is the part of the brain in charge of processing memory, decision-making and emotional responses [8].

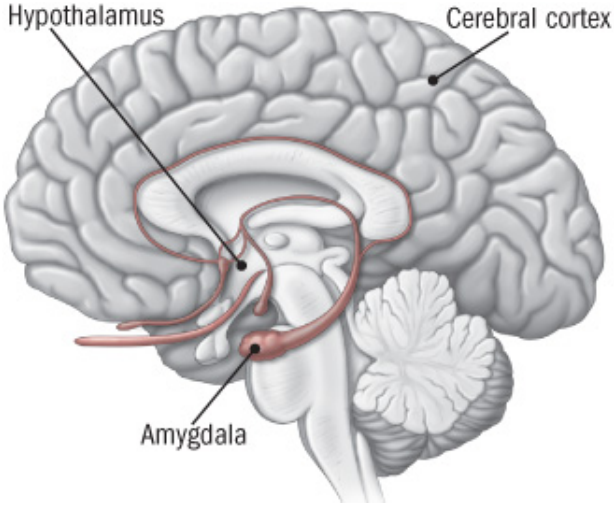

Figure 2.4: Locations of the hypothalamus and amygdala in the brain.

Thus, some physiological responses related to emotional changes and responses related to "fight or flight" and "rest and digest" are controlled by the ANS. Nevertheless, depending on the reaction, these are regulated by either the sympathetic or the parasympathetic parts of the nervous system.

The sympathetic nervous system (SNS)

The sympathetic nervous system (SNS) is the part responsible to those reactions related to the "fight or flight" expression. When feeling under pressure or in a situation perceived as threatening, the SNS activates several different organ functions: the heart and respiratory paces get faster, sweating glands get active, the digestive system stops working so that all the oxygen in the blood goes to muscles... All these functions get active at the same time. To enable it, the SNS ramifies from the brain to many different ganglia that are mostly

located alongside the vertebral column. Figure 2.5 shows a map of the vertebral column where the names of all vertebrae are given:

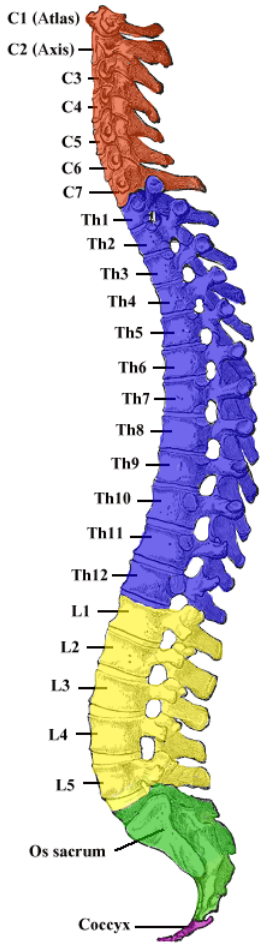

Figure 2.5: Map of the vertebral column.

As shown in Figure 2.5 (developed by Henry Gray in [9]), "C" vertebrae correspond to cervical, "T" to thoracic, "L" to lumbar and "S" to sacral areas. It is interesting to know about this naming system as the nerve connections also follow this system. Following this criterion, thoracic, lumbar, and sacral nerves are then numbered by the vertebra above them, e.g., the spinal nerve T1, which is just under T1 vertebra, connects to T1 ganglion, and, from there, it ramifies to different effector organs. This can be seen in Figure 2.6 (at the end of the subsection), where the connections of the SNS are depicted.

As shown in Figure 2.6, the SNS has connections to a very extensive set of organs which are in charge of very different functions. Not all connections

work in the same way. Sometimes, nerves connect to ganglia and from there, the connect directly to organs (as it happens with the heart or trachea). There are other cases in which nerves connect to a certain ganglion that, at the same time connects to other ganglia to reach the final organ. For example, this phenomenon happens for the eyes, that they connect to the spinal cord using the left chain ganglia. Finally, there is a third case in which the organ is connected to a ganglion that is located in a middle position on the way to the spinal cord ganglia. The liver constitutes an example of this type of connections, which connects to the celiac ganglion before getting to the spinal cord.

The parasympathetic nervous system (PNS)

Acting on the opposite direction to the SNS, the PNS works complementarily to the SNS in those cases in which the body needs to calm (the so-called "rest and digest" situations) instead of getting prepared to fight or run away from problems. Therefore, if the SNS was the one in charge of accelerating the pace of heart beating and breathing, the PNS is the part of the ANS that makes those paces to reduce to normal ratings.

Figure 2.7 (also at the end of this subsection) shows that the PNS is connections are different to how the SNS was connected; instead of being branched all along the spinal cord, most of the effector organs are connected through cranial nerves directly to the brain.

In Figure 2.7, is it important to note that sweat glands are not connected to the PNS and that they are only connected to the sympathetic part of the ANS. This information will be useful for later explanations in part III.

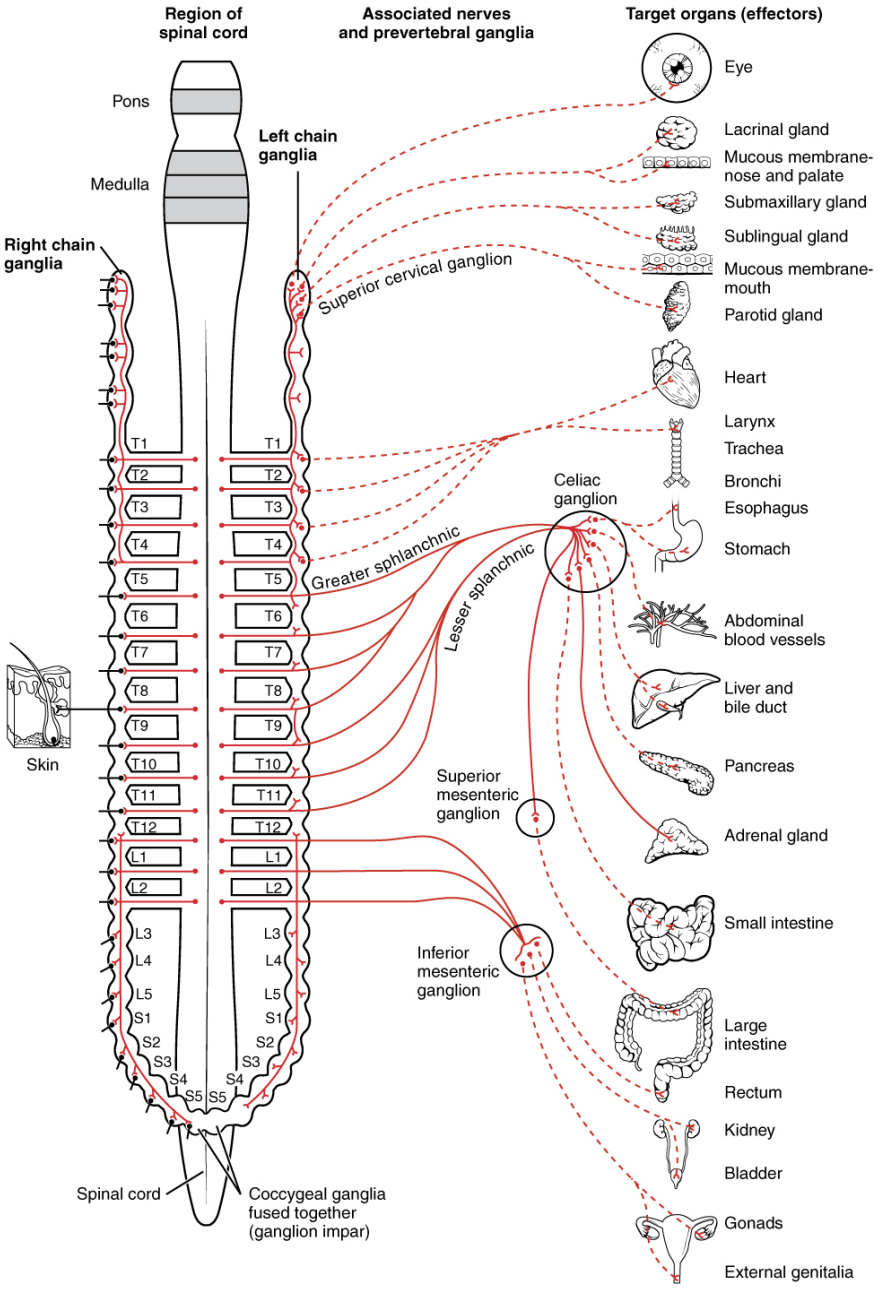

Figure 2.6: Connections of the sympathetic nervous system (SNS).

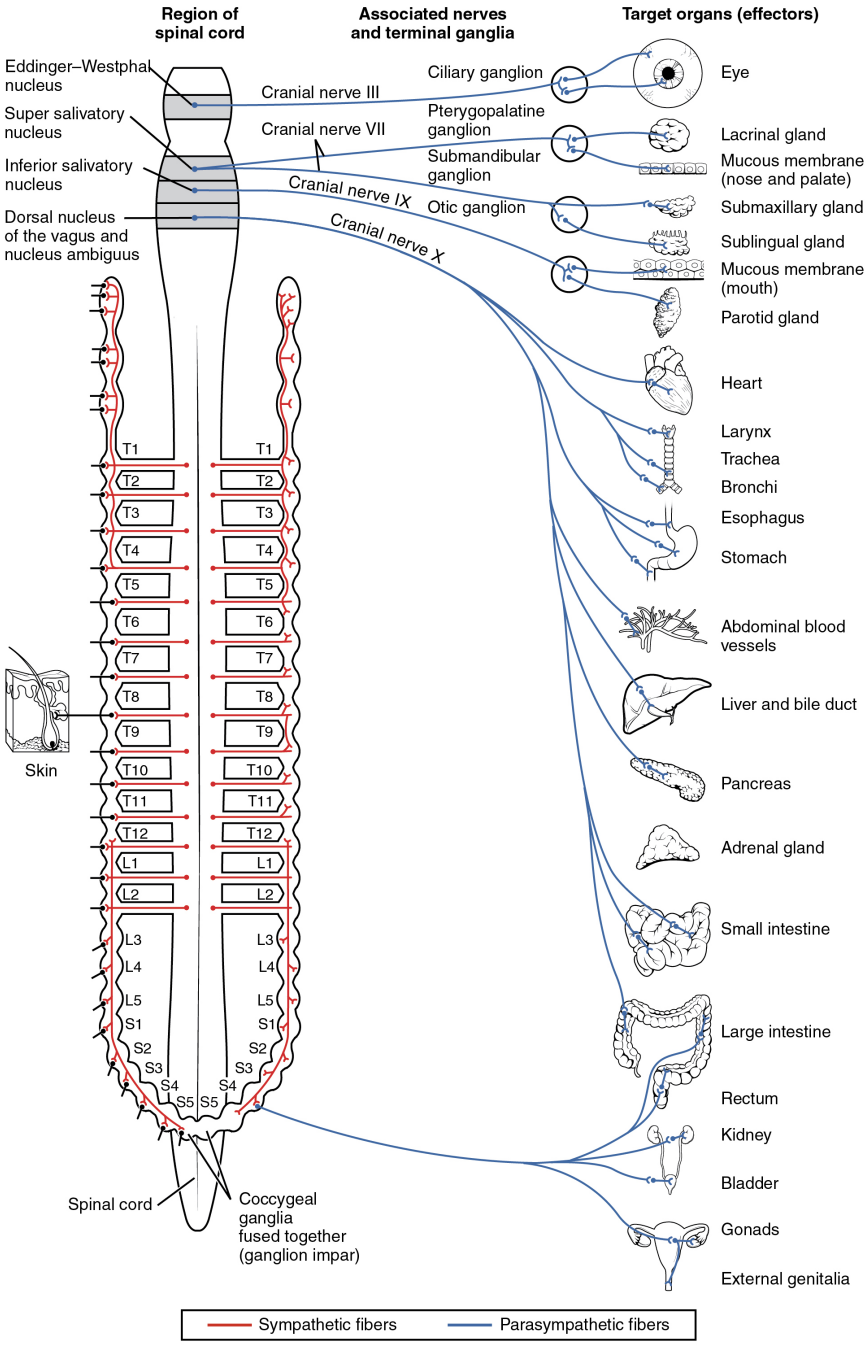

Figure 2.7: Connections of the parasympathetic nervous system (PNS).

2.2.3 The circulatory system

The third organ system that this document is going to overview corresponds to the circulatory system. The circulatory system is composed of the lymphatic and the cardiovascular system.

The lymphatic system is the main part of the immunological system but it also in charge of draining excess body fluids body, returning them to the bloodstream and distributing the cells of the immune system through the body. The flow of fluids produced by the lymphatic system is not actively pumped by the heart. Instead, it is the body movements such as muscular movements or breathing the ones that produce this flow.

The other part of the circulatory system is the cardiovascular system, which is going to be the part of the circulatory system in which this document is going to focus on. The cardiovascular system is the organ system in charge of distributing blood to all the parts of the body so that cells can obtain nutrients and get rid of wastes. Three main components form the cardiovascular system: the blood, the heart and the vessels.

The blood

Blood is a curious connective tissue as it is the only connective tissue in the body that is fluid instead of solid. This phenomenon is due to plasma (a 92% is water), the main component in the blood. The other components in blood are red blood cells, white blood cells and platelets. The average proportions of these elements in blood are shown in Figure 2.8 extracted from [10]:

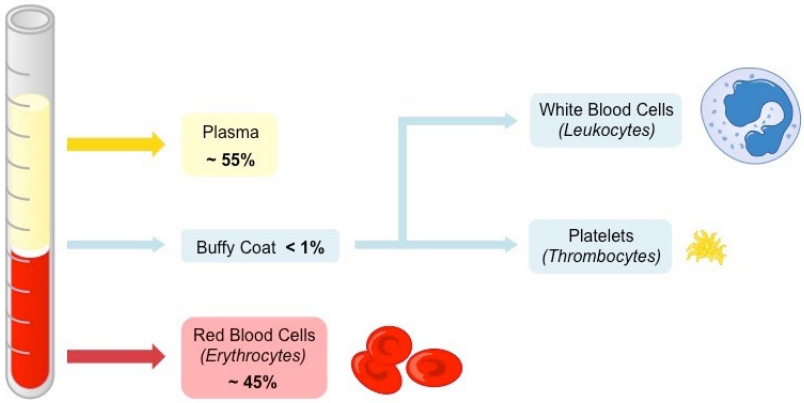

Figure 2.8: The main components in blood are plasma and red blood cells. White blood cells and platelets only constitute a very small part of blood's total composition.

Regarding body functions, the most well-known function is the one of taking wastes from cells (carbon dioxide, for instance) and given them in exchange

nutrients and oxygen so that they can continue functioning correctly. This function is mainly done by red blood cells. However, there are other functions in which blood plays an important role. For instances, it has an important defensive role as white blood cells fight external bacteria or inner threats such as virus propagation. Moreover, whenever there is a broken vessel or tissue, platelets and some proteins in plasma react to close the broken area and to stop bleeding. Finally, it is also important for the thermal regulation of the body. When the body core gets too hot, the flow of blood towards the limbs increases producing a decrease of the core. Also, whenever the core is getting cold, the circulatory system reacts in the opposite direction by reducing the blood flow going to the limbs to keep the body core as warm as possible.

The vessels

Vessels play an important role in the circulatory system as it is thanks to them that blood can reach all the different parts of the body. Mainly, there are two types of vessels: arteries and veins. The first ones correspond to the vessels that go from the heart to another part of the body. On the contrary, veins are those vessels that go to the heart coming from a different part of the body.

Getting deeper in vessel anatomy, both arteries and veins respectively branch into smaller vessel: arterioles and venules. These two have similar structures to arteries and veins, it is mostly that they are smaller. Finally, capillaries are found at the lowest level of vessel ramification and are the thinnest vessels. It is through capillaries that nutrients and oxygen are transferred from blood to other cells and that carbon dioxide and other wastes are passed from cells to blood. These transferences take place thanks to the permeability of capillaries (some are more permeable than others) and to the pressure differences between the blood in the capillaries and the outer environment.

Looking at vessels from a constructive point of view, arteries and veins share similar properties. However, as arteries get blood from the heart, that blood has a greater pressure. Thus, to handle those higher pressures, arteries have thicker walls compared to veins. It is thanks to this major difference that the previously mentioned gas and nutrient exchange takes place. In fact, from arterioles blood arrives at capillaries applying a 35mmHg pressure against capillary walls. Normally, plasma outside capillaries is around 25mmHg. Subsequently, as pressure is higher inside capillaries, the matter in blood moves out to the outer environment. However, as blood flow continues, pressure in capillaries gets smaller, reaching only about 18mmHg at the beginning of the venous part of the capillary. Therefore, in this part of the capillary, the pressure difference forces matter to go from the outer environment into the capillary: this is how carbon dioxide is absorbed into blood. Finally, in a middle point, there is a part of the capillaries where both pressures inside and outside capillaries is the same, and, so, there is no matter exchange. It is important to note that in the arterial part of the capillaries the pressure differences with the outer environment are greater compared to those of the venereal part. Hence, there is more fluid ex-

change in the arterial part. Thus, that is why the lymphatic system takes also part in the circulatory system, as it compensates that fluid loss by getting the excess fluids back into the bloodstream. The whole matter exchange process is depicted in Figure 2.9.

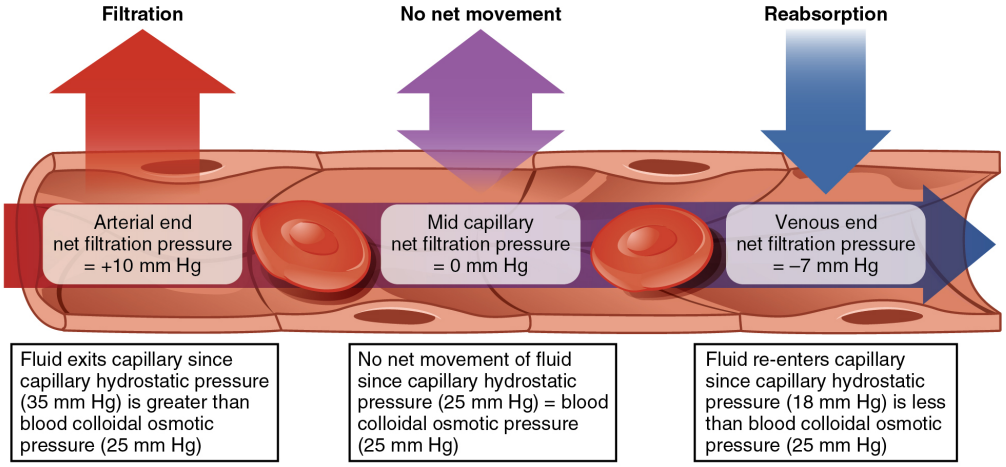

Figure 2.9: Matter filtrates from an environment to the other as pressure applied on capillary walls changes.

Lastly, regarding the bloodstream, there are two main circuits through which blood travels: the pulmonary circuit and the systemic circuit. Whereas the pulmonary circuit is the one that transports blood between the heart and lungs, the systemic circuit transports blood between the heart and the tissues of the rest of the body. Generally speaking, arteries carry blood with a higher oxygen concentration and veins carry blood with a low amount of oxygen and a high concentration of carbon dioxide. As the systemic circuit is larger, people are referring to it when they make this statement. However, arteries and veins belonging to the pulmonary circuit are an exception to this rule: pulmonary arteries take deoxygenated blood from the heart and send it to the lungs and pulmonary veins do the opposite, the take oxygenated blood from the lungs into the heart.

The heart

If the cardiac system was to be explained to someone who had no knowledge about biology, it could be easy to compare it to a closed-circuit water system: blood would act as water, vessels would do the job of pipes and the heart would be the one in charge of forcing the movement, i.e., the pump. Apart from being an organ, the heart is also a muscle and it is located in the thoracic cavity, at the level comprised between thoracic vertebrae T5 and T8 (please recall Figure

2.5 in Subsection 2.2.2). In a sense, it is found between the two lungs, as shown in Figure 2.10 (obtained from [11]).

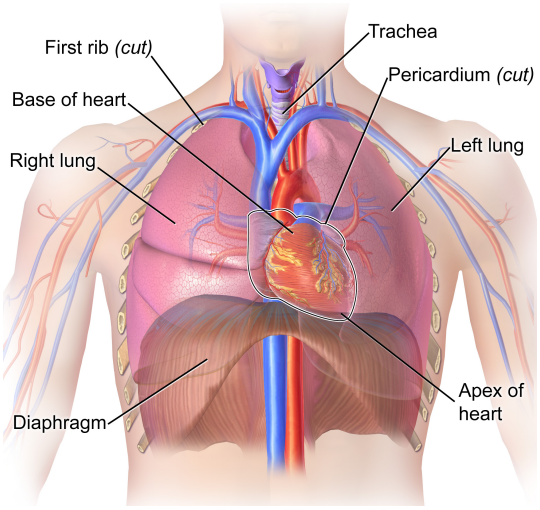

Figure 2.10: The heart is located between the lungs, at a middle level in the thoracic cavity.

There are four main chamber in the heart, as if they were divided into four quadrants with respect to the symmetric axe of the heart. Whereas the upper two chambers are named atria (atrium in singular), the lower two chambers are the ventricles. So, there is one atrium and one ventricle per each side of the heart, left and right.

Looking into the functionality of these chambers, atria are responsible for receiving the blood coming to the heart and pushing it into the ventricles. After that, ventricles pump blood out of the heart either to the lungs or to the rest of the body, depending on the side of the ventricles. Then, after all the oxygen, nutrient and waste transferences have been done, blood gets back again to the atria to start another cycle in this closed-circuit system. However, as the heart receives/pumps blood from/to both the lungs and rest of the body, this circuit is not as simple as it might seem at a first sight.

One could arbitrarily choose the right atrium to be a good starting point to explain how blood travels along the whole circuit. First, deoxygenated blood coming from all the body reaches the right atrium both via superior and inferior cava veins. There, blood goes through the tricuspid valve onto the right ventricle. Later, the ventricle pumps the deoxygenated blood through the pulmonary semilunar valve into the pulmonary trunk and arteries that go to the lungs. In the lungs, the blood releases carbon dioxide and is filled back with oxygen. After the gas transfer has been completed, pulmonary veins send the newly oxygenated blood back into the heart through the left atrium.

At this point, the left atrium pushes blood through the mitral valve into the

left ventricle. When the ventricle gets full it contracts and subsequently pumps oxygenated blood through the aortic semilunar valve into the rest of the body via the aorta artery and its many branches. Finally, at the terminal capillaries of the body blood releases oxygen and nutrients, and, after getting carbon dioxide from local cells, it travels back again to the right atrium to start a new cycle of the circuit.

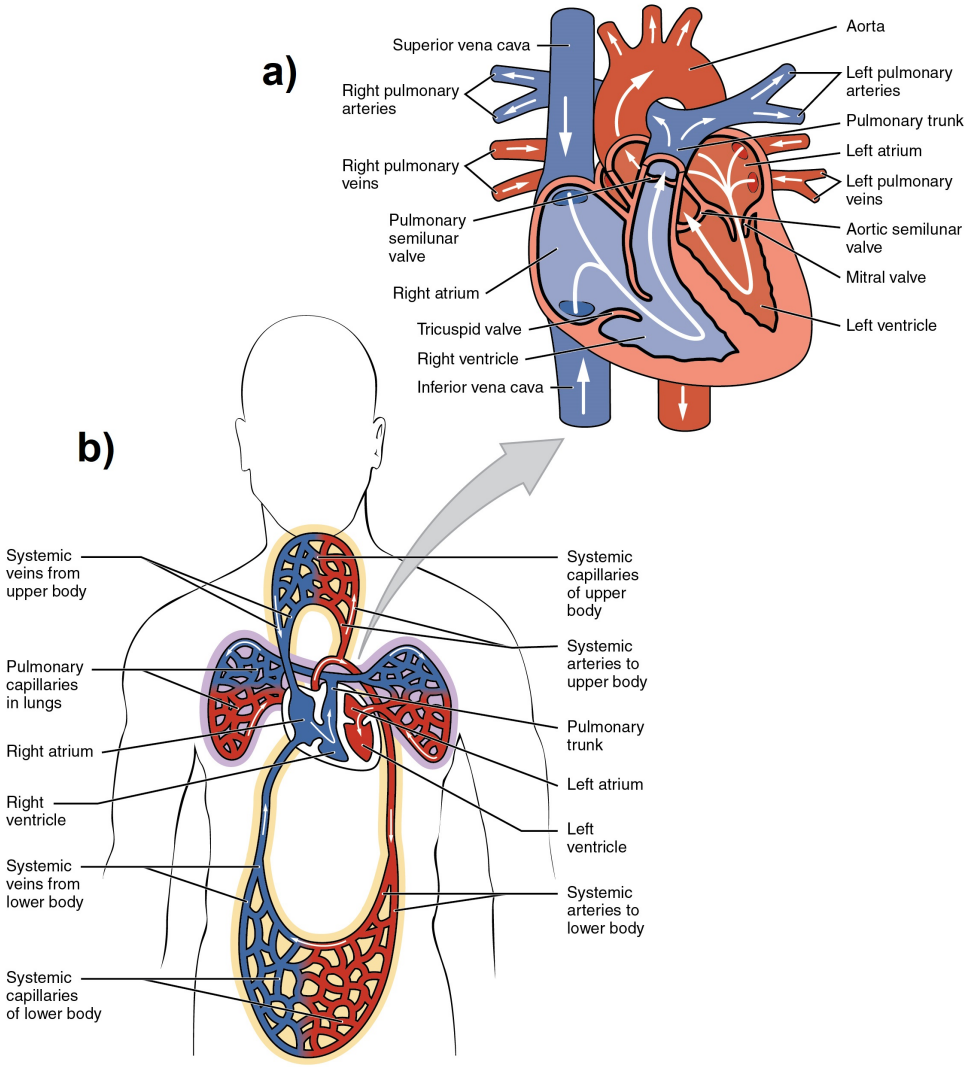

Figure 2.11: **a)** Detail of blood flow within the chambers of the heart. **b)** Closed-circuit of blood: the systemic circuit is highlighted in yellow and the pulmonary is highlighted in purple.

However, it is important to note that the right and the left part of the body work simultaneously, i.e., left and right atria receive blood at the same time and so do both ventricles contract sending blood respectively to the body and lungs.

2.2.4 The respiratory system

So far, this document has explained how blood travels around the body to deliver oxygen to tissues and to remove carbon dioxide from the organism. If tissues are one limit of the circuit, the lungs can be said to be the other terminal part. Nevertheless, although it is in the lungs that the gas transfer is done, both oxygen and carbon dioxide have respectively to get into and get out of the body cyclically. The organ system responsible for collecting and expelling these gases is known as the respiratory system. However, as it also happens for other organ systems, the respiratory system does cover some other functions such as sensing (i.e., smelling), coughing or speech production. This subsection will give a brief explanation about how this organ system works concerning the gas exchange mentioned in the previous Subsection 2.2.3.

The respiratory system is composed of two major parts: the conducting zone and the respiratory zone (see Figure 2.12). As it can be deduced from its name, the role of the conducting zone is to provide a path for the gases to get into and to escape from the lungs. Also, the conducting zone is responsible for filtering the incoming gases so that no external threats enter the organism as well as conditioning them for their arrival at the lungs at certain pressure, temperature and humidity conditions. On the other side, the respiratory is the part in charge of producing the gas exchange with blood.

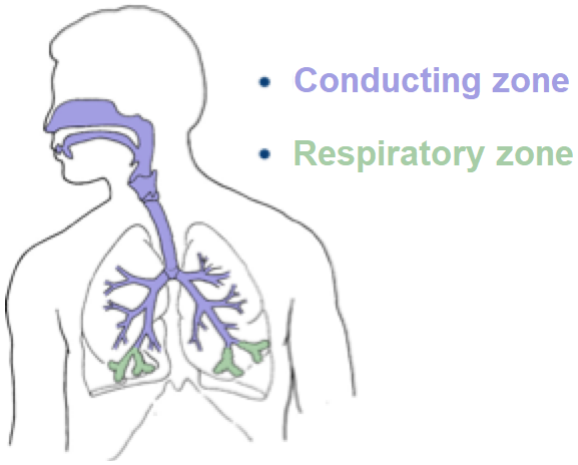

Figure 2.12: The two main zones of the respiratory system.

The conducting zone starts in the nose, where air gets into the body through the nostrils. As air passes through the nose cavities, it gets filtered by both nose

hair and mucus avoiding external threats from entering the body. Moreover, it is in the nose where air gets the temperature and humidity conditions needed for the body.

After the nose, air gets into the pharynx, which is a muscle that enables air to get lower into the body. It also contains several lymphocytes (a type of white blood cell) that help to eliminate remaining pathogens in the air. Besides, as it is not only connected to the nose but also the mouth, the pharynx is also the place where food gets into the body. However, at the lower part of the pharynx, the path for food and air get separated where food is sent to the digestive system through the oesophagus and air goes into the next part of the respiratory system: the larynx.

The larynx is the part of the conducting zone that connects the pharynx with the trachea and is the one in charge of controlling the amount of air entering and leaving the respiratory system. When swallowing, the muscular movement forces the larynx to close; it is this way how food and drinks are prevented from going further into the respiratory system. Also, it is in the larynx where the vocal cords are located, so it is in the larynx where different pitch sounds are produced.

Following the airflow, the next part of the conducting zone is the trachea, a flexible muscle that allows air to pass through and that connects to the primary bronchi, one for each lung. Later, air flows through the primary bronchi into the bronchi tree, which, as its name explains, it is a ramification of the primary bronchi into smaller bronchi. Then, each branch of the tree gets thinner until bronchioles are reached. It is at this point where the conducting zone of the respiratory system is said to finish and where the respiratory zone begins. The path that gases follow along the conducting zone can be seen in Figure 2.13 (USA public domain image).

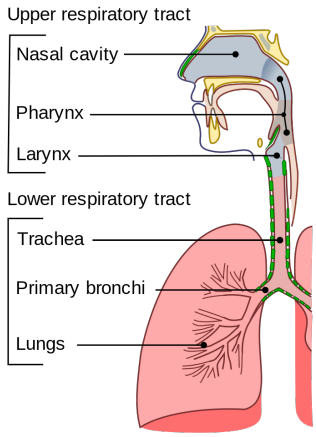

Figure 2.13: The conducting zone starts in the nose and ends in bronchiole.

The respiratory zone begins at the smallest size bronchioles, the respiratory bronchioles. Respiratory bronchioles connect to alveoli. Alveoli are the grape looking smooth muscles via which the gas transfer with blood is done. Therefore, to enable the gas transfer alveoli are surrounded by several capillaries that, thanks to pressure differences, collect oxygen and release carbon dioxide. Figure 2.14 gives a detailed image of bronchioles derive into alveoli and how, at the same time, these are surrounded by capillaries.

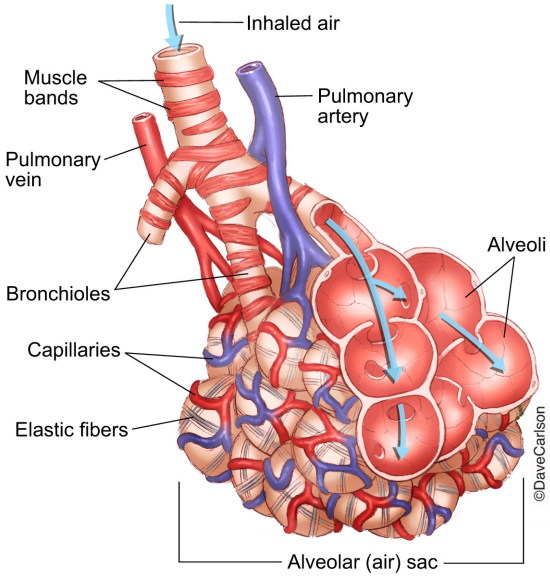

Figure 2.14: Gas transfer is done between alveoli and capillaries.

Chapter 3

Physiological signals, fuzzy logic, machine learning and BCI systems

3.1 Introduction

Whereas the previous chapter of this thesis was dedicated to giving the minimum background information concerning the human anatomy and physiology, this third chapter will focus on the technical aspects related to the signals and the computation techniques related to those organ systems presented in Chapter 2.

The chapter will be divided into three sections. The first, Section 3.2 will present the physiological signals that have been used for the contribution works of this thesis. Besides, this chapter will also explain some of the most commonly extracted features from those signals. Then, Section 3.3 and Section 3.4 will respectively present fuzzy logic and some machine learning techniques as both were used during this period of thesis. Finally, this chapter will conclude with Section 3.5, where three state of the art brain-computer interface (BCI) systems will be presented.

3.2 Physiological signals and feature extraction

This section will present certain signals that are representative of the functioning of the organ systems explained in Section 2.2. Although there are more signals that provide interesting information about certain organ systems, such as the photoplethysmography or the skin temperature, this section will only discuss about those signals that were used for the contributions presented in Chapters 4, 5 and 6 (Part III). First, Subsection 3.2.1 and 3.2.2 will respectively explain the electrocardiogram and the electro-dermal activity (signal related to sweating). Then, Subsection 3.2.3 will give information on the respiration. Finally,

Subsection 3.2.4 will analyse the electrical signals that can be collected from brain activity (electroencephalograms). Throughout each subsection, this document will provide an explanation about the processing techniques to obtain some of the most typical features of each signal, including among those used for the contributions of Part III.

3.2.1 The electrocardiogram (ECG)

The electrocardiogram (ECG) is the electrical signal related to the electrical activity of the heart. This signal provides very interesting information concerning the functioning of the heart. Because of that, this signal is of common use in medical centres, sports and many other fields.

The signal is captured by placing electrodes on the skin, which reflect the electrical stimuli of the heart. Depending on the application, the amount of electrodes used for capturing the signal varies from 3 leads to 12 leads (see Figure 3.1 taken from [12] for two examples). It is possible to obtain different information depending on the configuration chosen: those settings using more leads are the ones that give more information about the state of the heart. For example, paying attention to signal morphology of ECG registers, trained medical staff can detect anomalies in the functioning of the heart (such as blocks or tachycardia) that help in the detection of different heart diseases.

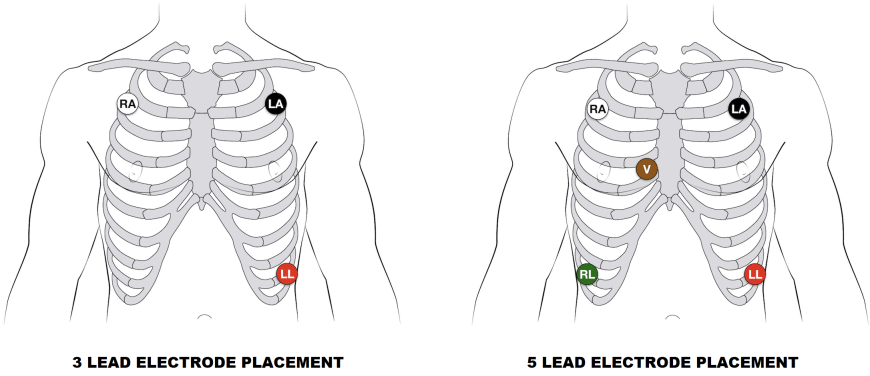

Figure 3.1: Electrodes are placed in different positions depending on the amount of leads used to collect an ECG.

Although it is not totally a periodical signal, because its frequency is variable, the morphology of the signal is repetitively producing the so-called PQRST waves. This name comes from the different waves or events that repeat cyclically: the P wave, the QRS complex and the T wave. The following Figure 3.2 provides an example of two ECG cycles in which the peaks of each wave has been highlighted:

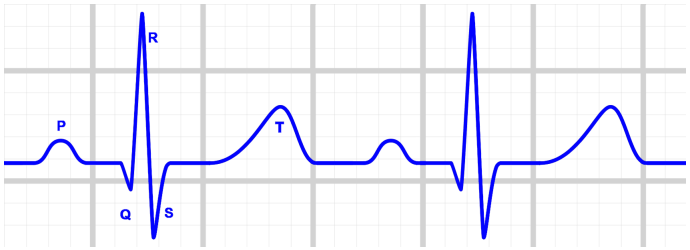

Figure 3.2: Two cycles of the ECG signal highlighting the different repeating events.

It has been previously said that the ECG is the signal representative for the heart's electrical activity. It represents how atria and ventricles polarise and depolarise. However, it is possible to establish a relationship between the events in the ECG and the mechanical movements of the heart. In each cardiac cycle, two main parts take place in atria and ventricles: the systole and diastole. Systoles are the part in which atria or ventricles contract and they eject blood. Diastoles correspond to when they relax and permits blood to enter the corresponding chamber.

At the beginning of a cardiac cycle, all the parts of the heart are relaxed. During this period, which finishes in the mid-point of the P wave, blood goes into the atria. At the same time, as the mitral and tricuspid valves are open (these are the atrioventricular valves), blood also gets into the ventricles filling up to 70% or 80% of their capacity. Then, during the second half of the P wave, atrial systole occurs and atria contract sending the remaining blood to the ventricles totally filling them. Atrial systole starts at the mid-point of the P wave and lasts until the peak of the R wave.

Then, at the time of the R peak, the first stage of ventricular systole takes place and ventricles start to contract. As they start contracting, pressure within ventricles increases and blood tries to scape to the body or back again into atria. Out of the heart the pressure is higher than inside the ventricles and so, pulmonary and aortic semilunar valves remain closed. However, the pressure produced in ventricular walls pushes atrioventricular valves and forces them to close. Hence, during this initial contraction, blood volume is kept constant and it is the pressure that grows up. Eventually, pressure in the ventricles gets so high that it forces the aortic and pulmonary semilunar valves to open and blood is pumped out of the heart. This is considered to be the second stage of ventricular systole.

Finally, the ventricular diastole starts after having pumped the blood out of the heart (T wave). As ventricles relax, pressure inside ventricles reduces and semilunar valves close. However, atrioventricular valves are still closed, so during this initial stage of ventricular relaxation volume is kept constant again. Later, as the ventricles relax, pressure continues decreasing and atrioventricular valves open, allowing blood getting back again into ventricles. It is in this stage

when a full cardiac cycle is completed and a new one starts.

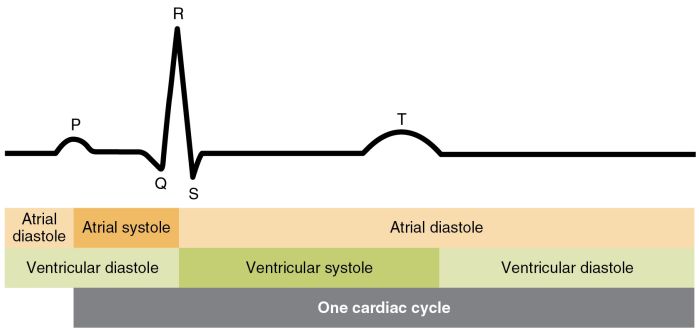

Figure 3.3: Relationship between mechanical events and the electrical signal of muscle polarizations and depolarizations (figure extracted from [4]).

ECG processing and feature extraction

Processing the ECG signal is a very common practise in physiological computing as it is a relatively easy signal to collect which provides very useful information. Depending on the intention of the analysis, different features are extracted.

- **Heart rate (HR):** this feature corresponds to the time difference between adjacent heartbeats. It is traditionally given in a minute basis (in beats per minute, [bpm]). It is possible to calculate it between two heartbeats that are sequentially one after the other, normally using R peaks as the reference points for calculating the differences. It is also common to calculate the average time difference between various adjacent beats and then transform the result to a minute basis so that a less dynamic signal is obtained.

  As explained, when the SNS activates it forces the heart to beat at a faster pace and so the heart rate increases. On the contrary, when the PNS activates the opposite effect takes place and the heart rate decreases. Thus, the heart rate is inversely proportional to the time differences between beats. It can be defined using equation 3.1, where $\Delta t_{n,n-1}$ stands for the time difference between to subsequent heartbeats in a set of N total beats.

$$HR = \frac{60}{\left(\sum_{n=1}^{N} \Delta t_{n,n-1}\right)/N} \tag{3.1}$$

- **Heart rate variability (HRV):** this feature does not give information about how the heart beats, but about how the beating pace varies. It

is known to provide significant information about aspects such as the capacity of the heart to cope with physical efforts [13] or events related to psychological changes as, for example, handling stress [14]. On the contrary to what common sense tells us, having a high HRV is significant for a healthy heart. Therefore, the HRV is a feature that can be useful to detect certain heart dysfunctions [15].

There are different ways to calculate this parameter (and even to look at it). Although there are other time-domain methods based on statistical calculations, a common technique is to calculate the HRV using the Root Main Square Successive Difference method (RMSSD). To calculate the RMSSD HRV first it is necessary to obtain the differences between various heartbeats and then calculate the square power of each of those differences. Then, the mean value of all the squared differences is calculated and, finally, the square root for this mean values has to be obtained. The whole calculating process gets summarised with 3.2.

$$HRV_{RMSSD} = \sqrt{\frac{\sum_{n=1}^{N} \Delta t_{n,n-1}^2}{N}} \tag{3.2}$$

- **The Poincare plot:** An interesting approach to visualize HRV is using the Poincare plots. The Poincare plot is a scattered plot that consists in plotting the n+1$^{\text{th}}$ RR peak intervals related to the n$^{\text{th}}$ RR peak interval. For example, if the time intervals between four peaks would be $RR_{21} = 0.6s, RR_{32} = 0.8s, RR_{43} = 0.7s$ and $RR_{54} = 0.4s$, the Poincare plot would be the depicted in Figure 3.4.

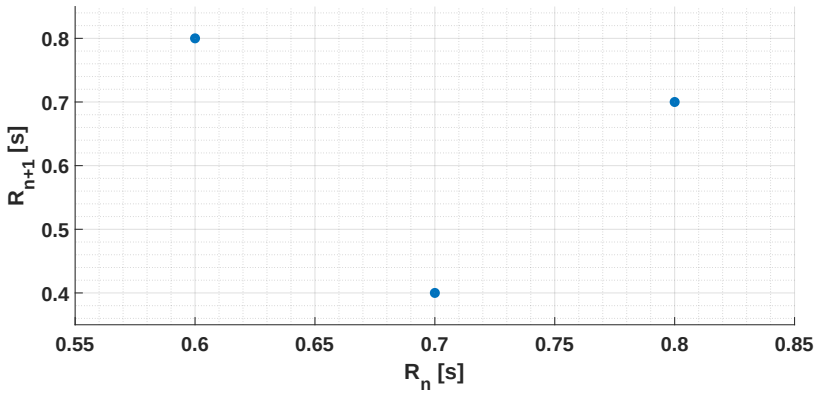

Figure 3.4: Poincare plot of the $RR_{21}, RR_{32}, RR_{43}$ and RR_{54} intervals.

Although in Figure 3.4 there were only three points, normally Poincare plots typically show an elliptical shape point clouds. Poincare plots are useful for describing the non-linearities of the HRV such as the instant

short term and continuous long term standard deviations of the HRV (SD1 and SD2 in Figure 3.5 respectively). Also, certain studies find Poincare plots to be useful tools for detecting certain cardiac affections [16].

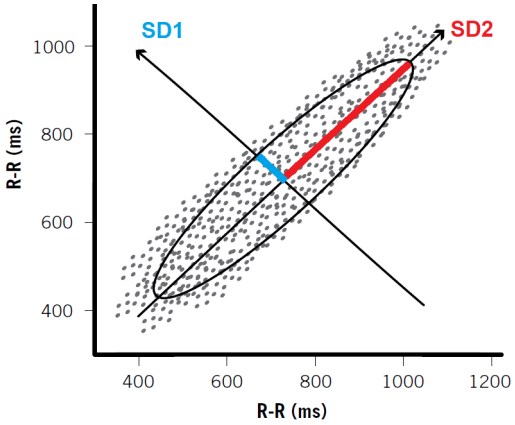

Figure 3.5: Typical aspect of a Poincare plot. There are some meaningful features in the plot, such as SD1 and SD2.

- **HRV frequential analysis and low/high-frequency ratio (LF/HF):** it is also possible to do a frequential analysis of the HRV using a Fourier transform to get information about its different frequency bands. Low frequencies (LF) are considered to be bound in the [0.04,0.15] Hz range and high frequencies (HF) in the [0.15,0.4] Hz range. The LF data is related to the heart beating pace (and so, it is an indicator of the activation level of the SNS) and the HF is be related to the HRV variability. There is another metric, which is the LF/HF ratio. This metric calculates the ratio between the HRV spectral power density in the LF and HF bands. However, there are doubts about the benefits of using these metrics [17].

3.2.2 The electro-dermal activity (EDA)

Recalling what explained in Subsection 2.2.1, it is in the skin where both eccrine and apocrine sweat glands are located. Whereas apocrine are related to hormone production, eccrine glands produce sweat for thermal regulation. Although they are almost all over the body, the eccrine gland concentration is bigger in the forehead, hand palms and feet soles. Curiously, the glands located in the hand palms and feet soles are more related to psychological changes and emotions rather than thermal regulation [18]. Thus, measuring the sweating of the skin in these areas is of great interest for psychologists and physiologist. Nevertheless, collecting sweat is a difficult task; it is easier to take electrical measures on the skin surface. The signal obtained from measuring the conductance of the skin is called electro-dermal activity (EDA).

There are two different ways to measure the EDA. The first, the endosomatic method, does not need any electrical current to be applied on the surface of the skin. The second method, the exosomatic method, consists in sending a small electrical current between two electrodes placed on the skin surface to measure the conductance of the skin. The skin's electrical conductance varies proportionally to sweat secretion: the greater the moisture the easier it conducts electricity. This is the most extended method and it is also the method that has been used for collecting data during the period concerning this thesis.

Although EDA can be collected from any point in the body, due to the reason mentioned above, it is of common use to collect it from the hands. The preferable position for placing electrodes is on the distal phalanges. However, other positions are also possible, as shown in Figure 3.6 extracted from [5]. Another interesting point to bear in mind is that the normal tendency is to place the electrodes in the non-dominant hand: it is less prompt to have cuts, callosities and it allows the monitored person to use the dominant hand if it was needed.

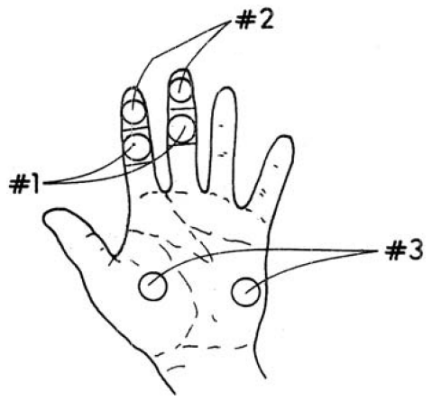

Figure 3.6: Different positions for placing EDA electrodes. The preferable position is number 2, on the distal phalanges.

Unlike it happened with ECG, EDA signals are not periodical. Instead, they have an accumulative nature as moisture needs time to evaporate. It is also a signal that is also affected by environmental conditions as it is strongly related to changes in humidity and temperature. There are two main components in EDA signals: a tonic component called Skin Conductance Level (SCL) and one phasic component named Skin Conductance Response (SCR). The SCL component corresponds to the low-speed variations of the moisture of the skin, i.e., corresponds to the baseline of the signal. On the contrary, SCRs correspond to the sudden sweat bursts as a consequence of SNS activations. Thus, it is common to see an EDA full of SCRs when catching someone's attention or putting a person in alert. This can be easily seen the EDA signal depicted in Figure 3.7: whereas the boxed left part the signal is free from SCRs (the person

is relaxing), the rest of the signal contains several SCRs meaning that the person was actively paying attention or getting into alert.

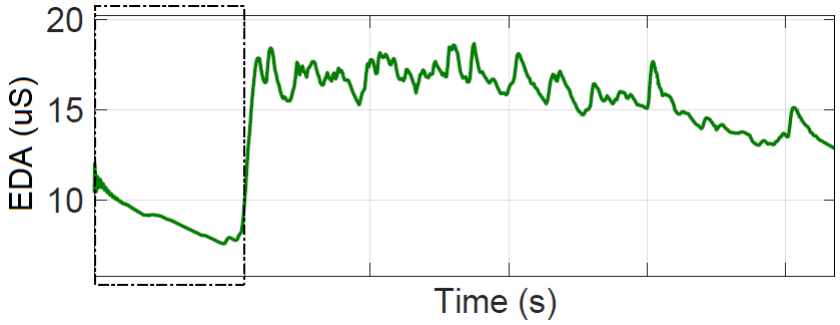

Figure 3.7: Unlike it happens in the rest of the signal, the boxed left part of the EDA lacks SCRs, meaning that the person is relaxing.

EDA processing and feature extraction

EDA has been broadly used for several purposes, not only for just medical monitoring but also for emotion recognition or for the automatic detection of patterns that lead to the detection of psychological phenomena [19]. As it has been mentioned, it is normal to decompose the EDA into the phasic and tonic components. Anyway, some more features and sub-features can be extracted from this physiological signal.

- **Tonic and phasic components (SCL and SCR):** the skin conductance level (SCL) corresponds to the slow dynamic component and the skin conductance responses (SCRs) correspond to faster variations of the signal or phasic component. These can be decoupled from each other and, mainly, it has been done to study the SCRs as they give direct information about the activation of the SNS. It is easy to decouple them using filtering techniques as they occupy different areas of the spectrum [20]: the SCL is found in the [0,0.05]Hz band whereas SCRs take place in the [0.05-2]Hz range.

- **Sub-features of the SCL and SCRs:** there are several sub-features that can be extracted from time domain analysis of the SCL and SCRs, as the listed in Table 3.1 and Table 3.2 that are proposed in [5] and [21]. These features belong both to morphological and statistical feature extraction techniques.

Table 3.1: Sub-features extracted from SCL.

Feature	Description
max value in a window	Peak value obtained by the SCL within the analysis window.
slope	Slope of the SCL within the analysis window.
mean value in a window	Mean value of the SCL within the analysis window.
standard deviation	Standard deviation of the SCL within the analysis window.

Table 3.2: Sub-features extracted from SCRs.

Feature	Description
number of SCRs	Number of SCRs taking place within the analysis window.
max value	Peak value obtained by the SCR within the analysis window.
mean value	Mean value of the SCR within the analysis window.
standard deviation	Standard deviation of the SCR within the analysis window.
slopes	Slopes of the rising and decreasing parts of the SCR.
latency	Time difference between the activating stimulus and the SCR.
rising time	Time the SCR needs for reaching the maximum point.
half recovery time	Time the SCR needs for decreasing from the maximum point to the 50% of the maximum.
total mean value	Mean value of all SCRs in the whole signal register.

As most of the works related to the content of this thesis have focused on stress and emotional detection, it makes sense that they are focused on SCRs as they give instant indication of SNS activations produced by those psychological stimuli. Hence, as shown in Table 3.1 and 3.2, most of the features are related to SCRs.

- **Power spectral density (PSD):** studies such as [22] propose to analyse the power spectral density to see the SNS activity that drives the EDA. Recall that a similar strategy was also presented for HRV analysis; the cardiac system is driven by both the SNS and the PNS and so, the usability of the method had been debated. However, as the EDA is only driven by the SNS, the application of the method to the EDA seems much more robust. Besides, unlike time-domain analysis features that vary vastly from subject to subject, the approach of calculating the PSD of the signal can be useful to handle those interpersonal differences.

- **Areas under the curve (AUC):** some researches propose to analyse the AUC to asses the intensity of SNS activations. To do so, there are

two main groups: who decouples the EDA and who do not do it. The first step for researchers of the first group is to decouple the EDA into the SCL and SCRs and then analyse the AUC for each component, especially focusing on the AUC related to SCRs ([23]).

On the other hand, the researchers of the second group ([24], [25]) take advantage of the shape of SCR bursts. Instead of calculating the whole AUC, they only do it for the rising part of the SCR. As the shape of the rising half of SCRs are almost triangular, they calculate the AUC of that half using the formula of 3.3.

$$AUC_{rising-half} = \frac{Amplitude_{SCR} \cdot \Delta t_{rising}}{2} \tag{3.3}$$

3.2.3 The respiration (RESP)

When talking about the respiration signal (RESP), this document refers to the electrical signal that is produced from the expansion and contraction of the chest during breathing. Many different measures can be extracted from the respiratory system, such as those given by spirometry, but those are more related to medical applications.

As previously explained, the RESP signal is the electrical signal related to the expansion and contraction of the chest. Therefore, it is necessary to place a band around the chest of the person's chest. This bands normally consist of a simple strain gauge that varies the output voltage according to the deformation it is suffering. Thus, the voltages given by the chest-band are greater when the lungs are full of air and smaller when air is being exhaled.

Looking at the signal, RESP signals tend to be relatively harmonic during relaxation as the pace of taking and exhaling air is relatively similar. However, during stressful events, the balance between inhaling and exhaling tends to get disrupted and the signal loses the harmonic shape: amplitudes vary, the breathing speed is increased, long breaths are taken eventually... Paying attention to these phenomena can be useful for determining certain psychological phenomena, such as stress. Figure 3.8 gives good examples for seeing the differences between the signals coming from calm and stressful situations.

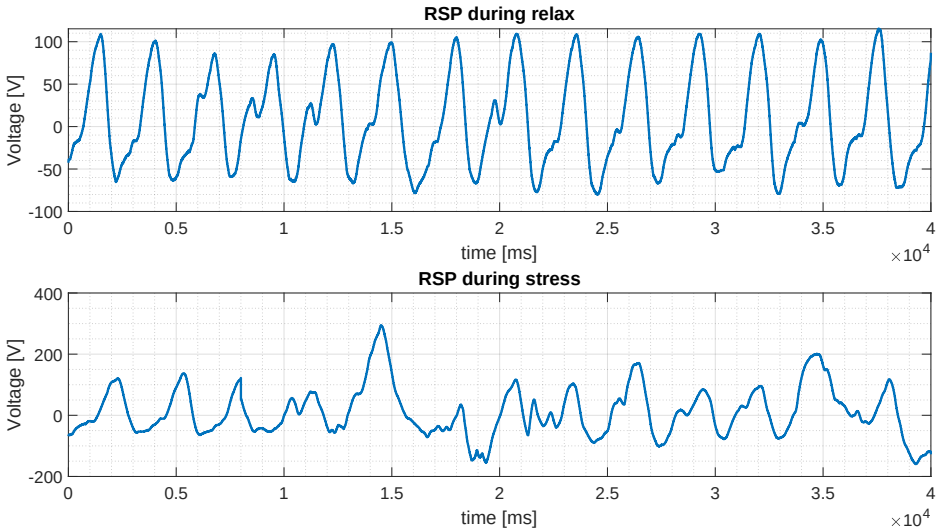

Figure 3.8: Whereas during relaxation it is close to being harmonic, the shape of the RESP signal gets completely distorted when getting stressed.

RESP processing and feature extraction

Taking into account that the respiratory system is very closely related to the cardiovascular system, it is normal that the variations produced in a system get mirrored in the other. However, unlike when processing ECGs, where it is more normal to pay attention to temporal features, it is typical to do it from a frequential perspective when processing RESP.

- **Power spectral density (PSD):** basing on the premise that when being calm the RESP is more harmonic, then it makes sense to look at the PSD to get a measure of how harmonic the signal is. Thus, knowing the normal breathing pace of a person during inactivity is useful to extract what could be considered to be the frequential pattern for a calm state and, so, any deviation from that pattern can be used to asses activation. Figure 3.9 depicts the spectra of the signals of Figure 3.8: when stressed, the RESP spectrum has more frequential components.

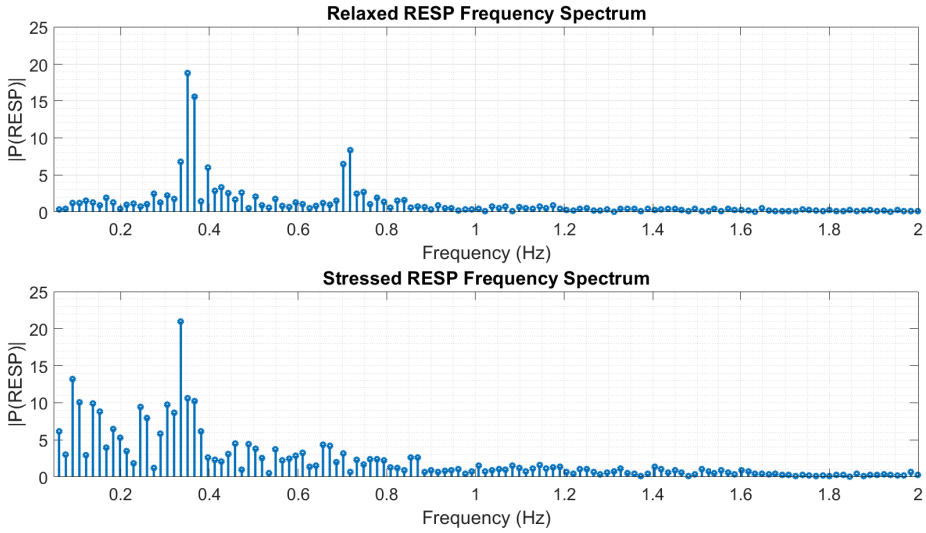

Figure 3.9: The spectrum of the RESP signal during stress is relatively powerful in many frequency components.

3.2.4 The electroencephalogram (EEG)

There are several different methods to study the phenomena that take place in the brain: positron emission tomography (PET), functional magnetic resonance imaging (fMRI), magnetoencephalography... one of those methods is the electroencephalography (EEG), which consists in collecting the electrical activity happening in the brain cortex. Studying of EEG, as it happens with other brain-related research topics, is a relatively new discipline. At the beginning of the 20^{th} century, it was only possible to collect the electrical activity of the brain using highly invasive methods, such as inserting needles into the brain cortex. Although there are cases that the oldest and most invasive methods are still used, as it happens with basic research using animals, nowadays it is more common to collect the electrical activity by just placing electrodes along the scalp. Hence, surgery is no longer needed for data collection.

Although it is not the simplest methodology to collect EEG, the 10-20 electrode positioning configuration is considered to be an international standard and it is widely used in clinical situations. The 10-20 configuration serves also as a reference for naming electrodes [26]. Thus, it is worth using it for explaining where electrodes are traditionally positioned. In the 10-20 configuration, there are four main reference points: the nasion (the point where the nose and forehead meet), the inion (the protuberance located at the back of the cranium) and the preauricular points of each ear.

Then, taking those four points as a reference, two imaginary axes are created

surrounding the surface of the cranium (nasion-inion and left-right preauricular points). Each axis is marked at 10% of the distance from one of its beginnings. Then, moving along the axis, it is marked every 20% of its total length. This marking methodology is what it gives the name to the electrode placing methodology, as marks are placed every 10-20-20-20-10%. Reaching this point, the scalp would have been marked 9 times. The five marks of the nasion-inion axis would correspond to the Fz, Cz, Pz points (that would correspond to electrode positions) and to Fpz and Oz points (that are not electrode positions but will be useful for later positioning of other electrodes). On the other side, all the five marks of the ear-to-ear axis correspond to electrode positions, being, from left to right: T3, C3, Cz, C4, T4. So far, the marking of electrode positions would be as depicted in Figure 3.10:

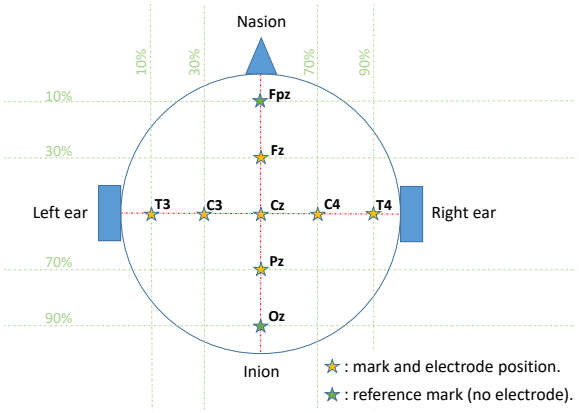

Figure 3.10: The first reference marks and electrodes are located by the 10-20-20-20-10% pattern along the axes.

The next step for locating the electrodes is to apply the 10-20-20-20-10% distancing pattern from Fpz to Oz surrounding the head from the sides in both directions. The 10 points resulting from this process are Fp1, F7, T3, T5 and Oz1 on the left side and Fp2, F8, T4, T5 and Oz2 on the right (see Figure 3.11). Note that not only these points, but all of them are numbered with odd numbers on the left part and using even numbers on the right part.

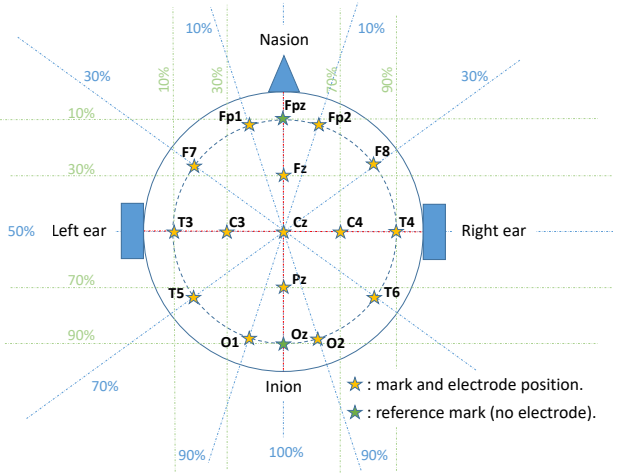

Figure 3.11: The new electrode locations are obtained by applying the 10-20-20-20-10% distancing process from Fpz to Oz.

Then, the location of the F3 electrode is calculated by finding the intersection points between the imaginary $\overline{Fp1, C3}$ and $\overline{F7, Fz}$ segments. F4, P3 and P4 electrodes can be positioned by applying the same criterion on the other three quadrants of the cranium, as shown in Figure 3.12.

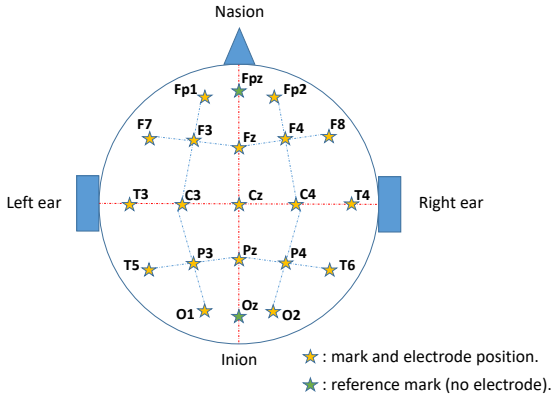

Figure 3.12: F3, F4, P3 and P4 are located using the segment intersections.

Finally, there are only two electrodes left, A1 and A2 reference electrodes. These are normally placed in the earlobes and they are used for removing electrical noises from the EEG channel measurements. They can also be placed in the mastoids (the bony prominences behind the ears) for the same purpose. If so, they are named M1 and M2 instead of A1 and A2.

Depending on the phenomenon to be studied and the resolution desired,

many equipments make possible to collect EEGs more easily. Whereas the simplest devices in the market only permit to record one or two EEG channels, the more advanced (and expensive) devices for clinical use permit the collection of over a hundred channels. However, the studies concerning physiological computing rarely go over 16 channels as real-life applications involving so many electrodes are rarely mobile or attractive for the final user. For example, some researches state that using only eight [27] or ten electrodes [28] should be enough to get good results in a P300 based brain-computer interface (more information on brain-computer interfaces will be given in Section 3.5). Moreover, for the sake of the ease of use, it is normally preferred to use wireless devices that are said to be "wearable" and that do not need using electrode gel.

EEG processing and feature extraction

- **Frequential band decomposition:** Concerning the electrical properties of EEG, most of the studies focus on the narrow [0-100]Hz band. Anyway, the slowest and fastest parts of the range, [0-7]Hz and [30-100]Hz respectively, are rarely relevant in adults and most of the relevant events and mental states take place within the [8-30]Hz range. Depending on the phenomena to be studied, the EEG can be divided into different frequency bands, being the Delta, Theta, Alpha, Mu, Beta and Gamma bands the most common [29].

  - Delta band: it corresponds to phenomena happening in the [0.5-3.5]Hz band. This band is normally related to the non-REM deep sleep states in which delta activity can get more than 50% of the total power spectral density. During wakefulness it is only predominant in newborn infants and it decreases as people get older. These signals are the ones with the highest amplitudes and are normally in the [20-200]μV range.
  - Theta band: it is the electrical activity of the [4-7.5]Hz band. In healthy awake adults (25-30 years old and above) the theta band is rarely powerful and when it is, it normally is a sign of some kind of disorder. Nevertheless, this band is normally present in children and also in adults when they are sleeping. The activity of this band in adults is related to drowsiness, meditative states free from thinking or with the less deep sleeping states. They can take place in any place of the scalp and so, they are not strongly related to any specific region. The amplitudes of the signals in this band are normally found in the [20-100]μV range.
  - Alpha band: it corresponds to activity in the [8-13]Hz band. Alpha rhythms take place while the subject is awake and take place in the posterior parts of the head, showing their highest intensity in the occipital areas. They are said to take place during relaxation or with

very low mental activity and they are seen best if eyes are closed (it gets strongly suppressed with the eyes open).

The amplitudes of this band are normally found in the [0-60]μV range. The study presented in [30] pointed out that 66% of the alpha rhythms are bounded in the [20-60]μV range and that their amplitude rarely goes above 60μV (only 6% of the cases). Alpha waves normally have sinusoidal or rounded shapes. However, there are cases in which the negative part of the wave can be sharp without meaning any clinical abnormality.

- Mu band: the mu band is located almost in the same frequential range as the alpha band. However, whereas the alpha band mostly takes place in the back part of the head, the mu band takes place in a band defined in from one ear to the other. This type of band is related to motor control of the body and the waves in these band get the most intense during physical rest. On the contrary, the mu band gets normally suppressed during motion or when actions performed by others are seen. Some researchers believe that the detection of the reductions of this band can be used for specific BCI systems [31].
- Beta band: it is [14-30]Hz band. The beta band was discovered by Hans Berger [32], the inventor of the EEG. The band is best seen in frontal and central regions of the head. The activity in this band is related to the active thinking and concentration and it hardly goes above 30μV (though normally the amplitude is much lower than that). These signals tend to be stronger in the central regions of the cortex.
- Gamma band: it corresponds to phenomena happening above 30Hz. The waves above 30Hz are said to be Gamma waves, but it is still unclear what can be they related to. Some researchers state that they are related to subjective awareness and to some sensory processing [33] (mostly related to visual perception). However, other researchers believe that gamma frequencies are due to mismeasurement related artefacts, as those produced by muscular electrical activity [34].

- **Common spatial patterns (CSP):** the extraction of CSP is a technique that, in spite of being originally designed for abnormal EEG detection, it is frequently used to obtain meaningful data from EEG signals. This method is based on linear algebra calculations to extract the covariance of several EEG channels through time and so, it is very suitable for BCI applications. For example, in the simplest motor imagery application (only two movement types), EEG data is collected with the intention to detect whether the patterns of the EEG belong to one type of movement or to the other [35]. In those situations, the covariance between the EEG patterns for each movement type can be of great help to determine the type of movement the EEG signals belong to.

For the explanation, consider that N EEG channels are being collected repeatedly every second. Then for every of those trials, the EEG channels can be given in a matrix A of size NxL, being L the total of samples collected for each channel. The covariance matrix CV of the samples of each imaginary movement trial i is obtained by applying equation 3.4.

$$CV_i = \frac{A \cdot A'}{trace(A \cdot A')} \tag{3.4}$$

Note that A' is the transposed matrix of A. Thus, $A * A'$ gives as result a square matrix from which it is possible to obtain the trace, i.e., the sum of all the values in the diagonal of a square matrix. If this operation is repeated for all the trials that have been collected, an overall covariance matrix $CV_{overall}$ can be obtained by aggregating all the CV_i and then dividing it by the number of trials. If this is done, it is clear that $CV_{overall}$ will include information related to all the different types of imaginary movements.

Continuing with linear algebra theory [36], if a matrix can be diagonalised, then it is possible to represent it factorised using eigenvector matrix V_{eig} and the eigenvalue diagonal matrix λ. Applying this, $CV_{overall}$ gets decomposed as done in equation 3.5.

$$CV_{overall} = V_{eig} \cdot \lambda \cdot V'_{eig} \tag{3.5}$$

After that, to get the CSPs properly, it is necessary to sort the eigenvalues of λ matrix in descending order (it will be further recalled). Then, from expression 3.5, it is possible to get the matrix whitening matrix W. Whitening a matrix is to transform it so that the covariance matrix is an identity matrix I (all elements in the diagonal equal to 1 and the rest of the elements equal to 0). Here, the whitening matrix is $W = \sqrt{\lambda^{-1}} \cdot V'_{eig}$. In such a way, it is possible to apply the whitening transformation to $CV_{overall}$ to make its eigenvalue matrix be an identity matrix I_{eig} (equation 3.6).

$$CV_{overall} = W \cdot I_{eig} \cdot W' \tag{3.6}$$

Knowing this, and, recalling what said that $CV_{overall}$ contains information from all types of movements (two types in our case), then it can be concluded that the eigenvalue identity matrix I_{eig} is a sum of the eigenvalue matrices of the two types of movements. Basing on this premise, the covariance matrices for the two types can be expressed as in equation system 3.7 using the same W whitening matrix as for $CV_{overall}$.

$$\begin{aligned} S_{type-1} &= W \cdot CV_{type-1} \cdot W' \\ S_{type-2} &= W \cdot CV_{type-2} \cdot W' \end{aligned} \tag{3.7}$$

Later, similarly to what done for $CV_{overall}$, it is possible to decompose each of those S_{type} matrices, getting equation system 3.8. Each transformation will give a new eigenvalue matrix λ_{type} different to an identity matrix (note that the whitening matrix W only does that for $CV_{overall}$). However, the eigenvector matrix B resulting from that transformation will be the same for the two types of movement.

$$\begin{aligned} S_{type-1} &= B \cdot \lambda_{type-1} \cdot B' \\ S_{type-2} &= B \cdot \lambda_{type-2} \cdot B' \end{aligned} \tag{3.8}$$

Doing this, finding the relationship between the transformations for each type of movement and the overall matrix becomes feasible, which is the one of equation 3.9.

$$I_{eig} = \lambda_{type-1} + \lambda_{type-2} \tag{3.9}$$

Finally, from equation system 3.8 and equation 3.9, it may be concluded that the eigenvector of matrix B that has the greatest eigenvalue λ_{type-1} will make the complementary eigenvalue λ_{type-2} be as opposite as possible. In the same way, the greatest eigenvalue λ_{type-2} produces the smallest λ_{type-1} eigenvalue. Now, it is important to remember that after equation 3.5, all the values of λ were sorted from bigger to smaller. Therefore, the eigenvectors of B will also be sorted. This means that the initial and final eigenvectors of B will be the most useful for discriminating between movement types.

Reaching this point, it is possible to take advantage of this property and transform each trial's matrix E to get a set of projections in matrix Z, in which all the projections are sorted according to their meaningfulness for discriminating one class form the other. This is done by using the projection matrix Q, as shown in equation 3.10.

$$Z = Q \cdot E \rightarrow \ being\ Q = (B' \cdot W)' \tag{3.10}$$

The columns of matrix Q' are what is called the common spatial patterns (CSP) and they are invariant through time, being useful to arrange the EEG inputs according to their usability for classifying distinguishing between the two movement types.

It is important to mention that the CSP extraction method here explained is only suitable for binary class problems. However, there are ways to apply CSP methods to multi-class situations, as done in [37] or [38].

- **Morphological and statistical features:** as it happens with many of the other physiological signals of this chapter, it is also common to extract features related to the morphology of the signal, such as signal maximum and minimum values, average voltage, signal power, RMS value, number of zero crossings, among others.

3.3 Fuzzy logic

Fuzzy logic is a control strategy belonging to Computational Intelligence that, as stated in [39], has been broadly used for identification problems. It was firstly introduced by [40] and it has been used for several identification problems. This section will review the parts of which a fuzzy logic system consists of. The first subsection, 3.3.1, will explain how input and output membership functions are designed. Later, Subsection 3.3.2 will cover the rule systems that are used to relate the inputs to the output membership functions. Finally, the defuzzification stage will be presented in Subsection 3.3.3. The defuzzification is where the system calculates the value of the output from the relationships established in the rule system.

As fuzzy logic is a control strategy that sometimes might be difficult to understand, for the sake of clarity, the explanation this section will be conducted using the example of a simple fuzzy logic control of a shower.

3.3.1 Membership functions

Membership functions (MF) are one of the most important parts of a fuzzy logic system as it is via the MFs that variable values are related to certain control states or estimation levels. MFs also permit to handle certain ambiguities in the definition of the input variables using linguistic terms for the definition. For example, to control the water temperature of the shower, a user might define 34°C to be "cold" and 40°C to be "hot" water respectively. But what about 37°C? It might be there in the middle, belonging in part to the term "warm" and in part to the term "hot". Fuzzy logic MFs work in a similar way to this example and, consequently, are very powerful in the cases in which the boundaries to define a variable are not clear.

Hence, MFs are used to convert a variable from the machine understood domain, crisp from now on, to the linguistic domain in which human thinking operates. To do this conversion it is necessary to define a range of crisp values in which a specific linguistic term is going to take place for a variable. For instance, in the example above, it could be possible to say that the [30,34]°C range is considered to belong totally to the term "cold", and, beyond 34°C until 36°C, it could be said to be in part "cold" and in part "warm". Different function shapes can be used for this conversion: straight, triangular, trapezoidal, Gaussian, etc. As shown in Figure 3.13, which depicts the previous example of the temperatures, it is possible to vary the abruptness of the transition between MFs by choosing different MF shapes.

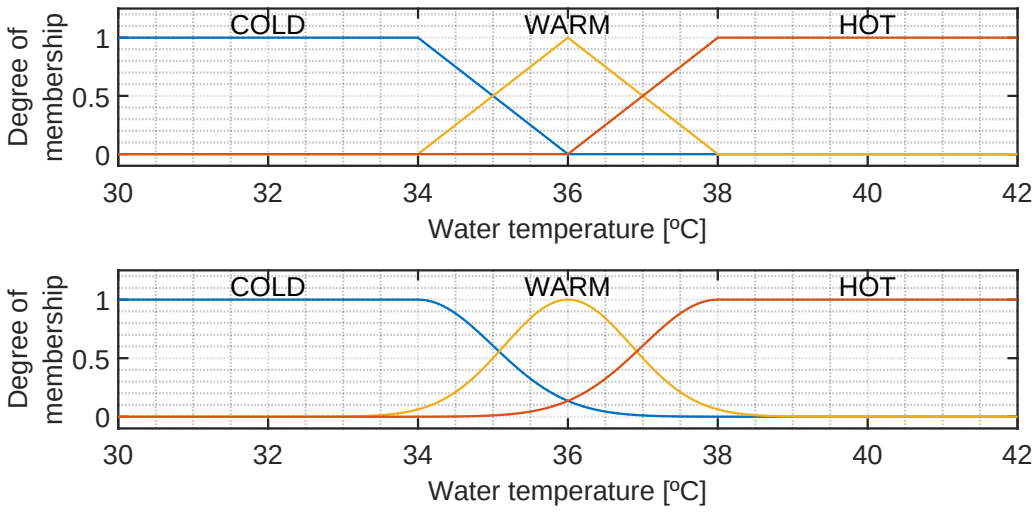

Figure 3.13: Whereas in the top part the MFs are trapezoidal and triangular, in the bottom part all the MFs have been defined to be Gaussian and sigmoidal.

This process of transforming the variables from crisp to the fuzzy or linguistic domain has to be done for all the inputs of the system that is going to be controlled or estimated. It is important to note that the membership values are normalised and, so, the highest membership value that any variable can have for a given linguistic term is 1. Nevertheless, as fuzzy logic permits to handle ambiguities, it is possible that a variable belongs totally to a term and simultaneously in a partial way to another (as it happens around 34°C or 38°C in Figure 3.13, for example). Also, bear in mind that it is not only for inputs that MFs have to be defined but also for the outputs of the system.

3.3.2 Rule system

Sometimes, it is complex to define the relationship between system inputs and outputs. In those cases, defining a set of rules based on human knowledge becomes very helpful [41]. So, once all variables have been transformed to the linguistic domain, it is time to establish the relationships between them and the control action or estimation output of the system. Relationships are established via "IF... THEN..." rules. However, there is no limit for combining input variable relationships; if a new input variable is added to the conditional chain it has to be done using either logical "OR" or "AND" operators. Also, sometimes "NOT" operators may be used.

Some showers only have a single handle, controlling both the temperature and the amount of water. Therefore, the position of the handle depends on both input variables. In this sense, the handle can open or close the valve and at the same time take it to a colder or hotter water position. Hence, for this situation,

Table 3.3: Rule system to control the shower.

IF... AND IF... THEN... rule system		
Water temperature	Amount of water	Handle movement
Cold	Too low	$\nearrow$
Cold	Ok	$\rightarrow$
Cold	Too much	$\searrow$
Warm	Too low	$\uparrow$
Warm	Ok	$=$
Warm	Too much	$\downarrow$
Hot	Too low	$\nwarrow$
Hot	Ok	$\leftarrow$
Hot	Too much	$\swarrow$

the rules system of Table 3.3 could be defined.

For the example of Table 3.3, the following consideration has been taken: water temperature increases from left to right and the amount of water increases moving the handle upwards. Hence, it is possible to make a similar control to what humans would do by relating variables with "IF... AND IF... THEN..." rules.

When these types of combinations are applied, as input variables have areas of ambiguity, more than one rule can be active in a certain degree. Therefore, it is necessary to calculate the surfaces resulting from each subset of conditional rules individually. Concerning this calculation, the resulting area is affected by the type of operators used in the rules. On the one hand, "AND" operators give as a result an area defined by the minimum membership value of all the inputs affecting the conditional chain. On the other hand, "OR" operators give as result an area defined by the maximum membership score.

To make it more clear, for the example of the shower, if the water temperature was 0.6 "hot" and the amount of water was 0.8 "too low", the result of the conditional rules would be different depending on the logical operator used. Whereas using an "AND" the area resulting would have its maximum membership value at 0.6, it would be 0.8 if an "OR" operator had been used (see Figure 3.14 for the graphical representation).

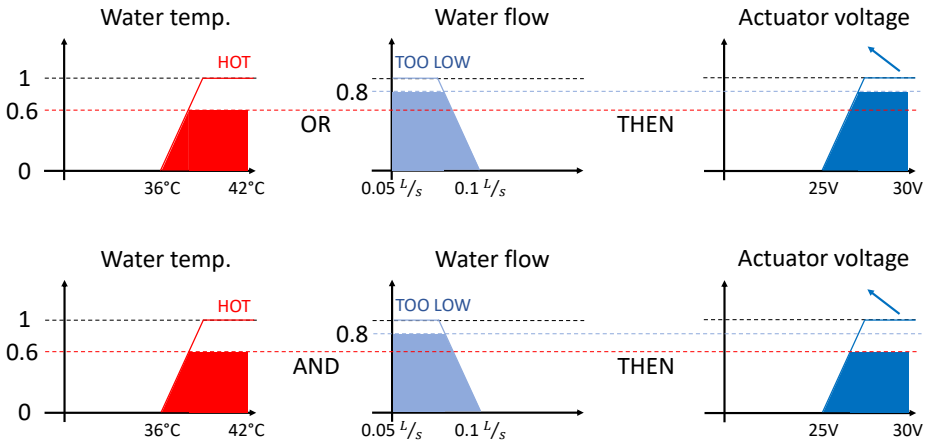

Figure 3.14: The membership of the output varies depending on chosen the logic operator.

As it can be seen in Figure 3.14, input variables not always have a membership of 100%. When this happens, likely, some other rules get also active with a lower degree of membership. Because of that, as different rules may give different resulting areas, it is vital to aggregate all the areas before going into the last step, defuzzification. These two last phases are going to be explained in the following Subsection 3.3.3.

3.3.3 Aggregation and Defuzzification

Before defuzzification, the fuzzy logic system needs to get the overall aggregated output. There are different methods to do so, as are, for example, the aggregation using the "max" operator or the summing method, among others. Both methods have the same starting point: they propose to get a general output by adding the independent areas resulting from each of the rule subsets. However, they perform differently when two independent result areas overlap. On one side, the "max" operator aggregation only takes into account the area scoring the highest values. On the other side, the summing method adds the overlapping areas and gets, as a result, an even bigger area. Figure 3.15 provides an example of how the general result varies when the partial results of the conditional sets are aggregated with different methods. Consequently, as the final area changes, the output voltage to control the shower handle would also vary in the defuzzification.

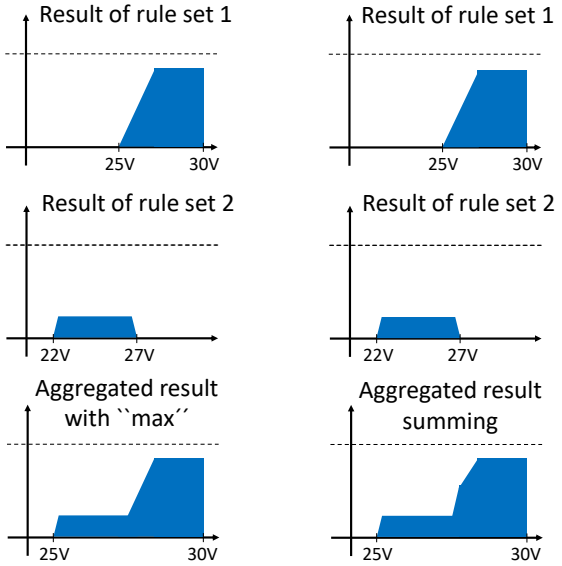

Figure 3.15: For the same partial results of each conditional set, different aggregation methods vary the general result of the system.

Finally, once the aggregated output has been calculated, the inference system needs to convert the overall output back to the original crisp domain. This step is called defuzzification. There are different approaches to do the defuzzification [41], such as calculating the centre of gravity (COG) or the mean of the maximums (MOM), among others [42].

The work presented later in Chapter 4 used the COG method. This method is the most popular defuzzification method and, as its name states, it is based on calculating the centre of gravity of the aggregated result. Equation 3.11 has to be applied to the aggregated result to obtain the COG, where μ_i stands for the maximum value of each of the N output MFs that create the aggregated result and χ_i stands for the centre value of the crisp value range covered by the i^{th} MF.

$$output_{crisp} = \frac{\sum_{i=1}^{N} \mu_i \cdot \chi_i}{\sum_{i=1}^{N} \mu_i} \tag{3.11}$$

For example, further expanding the example of the “max” operator aggregation of Figure 3.15 and giving it the values of Figure 3.16, the output crisp

value would be calculated as shown in the following equation 3.12.

$$output_{crisp} = \frac{\frac{30+25}{2} \cdot 0.8 + \frac{27+22}{2} \cdot 0.2}{0.8+0.2} = \\ = \frac{27.5 \cdot 0.8 + 24.5 \cdot 0.2}{0.8+0.2} = \frac{22+4.9}{1} = 26.9V \tag{3.12}$$

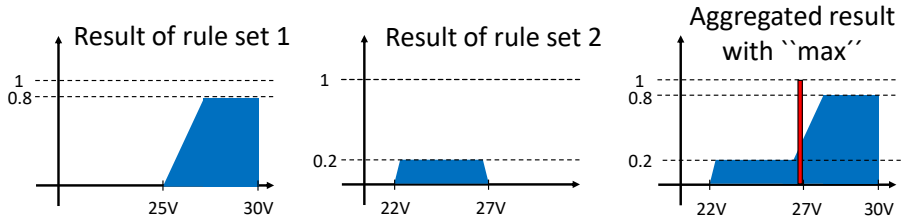

Figure 3.16: The COG defuzzification method gives a crisp value of 26.9 volts as result to control the movement of the shower handle (the centre of gravity has been marked in red).

3.4 Machine learning

When trying to interpret data for any type of purpose, sometimes humans cannot establish relationships patterns that make sense. However, despite they are not capable to see them, the analysts of these data often know (or at least suspect) that there exist underlying relationships and patterns that permit them to extract valuable information from the data. Luckily, machine learning makes possible to automatically find those patterns thanks to the higher degree of abstraction that algorithms can apply to data analysis. On the one side, if the algorithm is used for predicting a variable that is defined in a continuous range, the algorithm is said to operate a regression. On the other side, when the problem belongs to a finite discrete domain, the algorithm is considered to be a classifier.

Depending on how an algorithm learns from data, there are two major machine learning approaches: supervised learning algorithms and unsupervised learning algorithms. The following Subsections 3.4.1 and 3.4.2 will present some of the most relevant paradigms for both types of algorithms. Later, this document will give information about algorithm validation methods in Subsection 3.4.3. Finally, different classifier performance metrics will be presented in Subsection 3.4.4.

3.4.1 Supervised learning

The supervised learning algorithms are those algorithms that need external help to learn how to do a regression or to classify. In that sense, the external help is no other than telling the algorithm the class to which the learning instance belongs to. By knowing it, supervised learning algorithms are capable to establish functions that map inputs to the desired outputs.

Thus, all supervised learning algorithms require data to train themselves and to improve their performance. Training an algorithm is an iterative process in which for each iteration the algorithm is given a certain amount of data as inputs. These input data that are used to feed the algorithm is known as an instance. Once it has been fed with data, the algorithm tries to recognise patterns in data and it varies the mathematical model that it uses to recognise common data patterns in the instances. Hence, having a sufficient amount of data plays an important role when using machine learning; if there is not enough data then the training of the algorithm will be poor and so will also be the results. Nevertheless, overfeeding the algorithm is also negative and so, it is important to find a balance in the number of instances used for the training.

Later, once the algorithms have been trained, algorithms have to undergo another phase before being used in a real-life application: the testing. If all data available was used for training and a training instance was used to test the algorithm, then the algorithm would surely be good at handling the classification because it would already know all the instances. Therefore, to get trustworthy test, it is very important to save data only for the testing process, because, while keeping the safety of not being a real-life situation, it is where the algorithm is going to prove its suitability for future use in a real situation. A widely spread approach is to use 2/3 of the data to train the algorithm and 1/3 to test the model resulting from the training. Also, it is common to save a part of the training data to validate the parameters of the training just before going into the testing phase. However, there are other approaches for partitioning data for training and testing that will be further explained in Subsection 3.4.3. Figure 3.17 depicts a diagram of the basic approach for a data analysis problem using supervised learning.

There are several supervised learning algorithms ([43], [44]): Naive Bayes classifiers (NB), multi-layer perceptron (MLP), support vector machines (SVM), K nearest neighbours (K-NN), decision trees (DT) and ensembles of classifiers, among others. Each of these algorithms base on a different mathematical approach for achieving classification tasks.

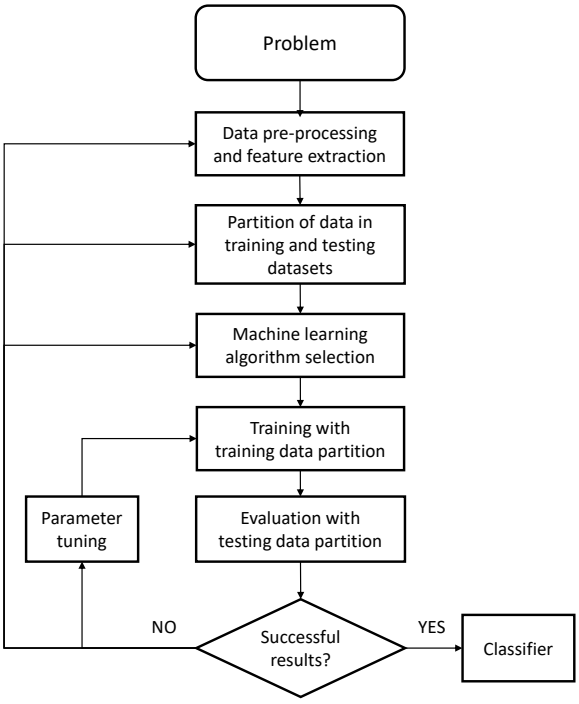

Figure 3.17: Basic diagram of how to approach a data analysis problem using supervised learning algorithms.

Naive Bayes (NB)

This classification method is based on Bayes' theorem for conditional probability. The mathematical expression of the theorem is presented equation 3.13, where $P(A \mid B)$ stands for the probability of A happening given B as a condition; $P(B \mid A)$ for the probability of B happening given A as condition and $P(A)$ and $P(B)$ stand for the probabilities of A and B happening respectively.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \tag{3.13}$$

Besides, these type of classifiers base on the assumption of the independence between the variables influencing the probability of a phenomenon happening [45]. Thus, if two independent conditional variables $F1$ and F_2 influenced in phenomenon A, it would be possible to assume that $P(A \mid F_1, F_2) = P(A \mid F_1) \cdot P(A \mid F_2)$.

Hence, taking the assumption of independence between variables and the expression of equation 3.13, the statistical model that calculates the probability of an instance for belonging to a certain class gets represented by the following

expression.

$$\begin{aligned} P(A \mid F_1, F_2, ..., F_n) &= \frac{P(A) \cdot P(F_1, F_2 ..., F_n \mid A)}{P(F_1, F_2, ..., F_n)} = \\ &= \frac{P(A) \cdot P(F_1 \mid A) \cdot P(F_2 \mid A) \cdot ... \cdot (F_n \mid A)}{P(F_1, F_2, ..., F_n)} \rightarrow \\ \rightarrow P(A \mid F_1, F_2, ..., F_n) &= \frac{P(A)}{P(F_1, F_2, ..., F_n)} \cdot \prod_{i=1}^{n} P(F_i \mid A) \end{aligned} \tag{3.14}$$

Knowing expression 3.14, it is possible to calculate the probabilities of an instance belonging to the different classes. Finally, having all those probabilities calculated, the instance is set as belonging to the class to which the belonging probability was higher.

Multi-layer perceptron (MLP)

MLPs are one type of artificial neural network (ANN) that is based on the backpropagation principle for learning. As biological neural networks, ANNs are composed of several processing units called neurons that are connected forming a network. Each connection between two single neurons will have a corresponding weight that will vary its value during the training of the ANN according to the importance of the connection for achieving a correct classification/regression task. Thus, needing to know the class of the training instances to vary the connection weights, MLPs are inside the supervised learning algorithm group.

Some parameters have to be decided when tailoring an MLP, such as the number of layers. However, the minimum amount of layer needed to build an MLP is three layers: the input layer through which the network is fed, the output layer that gives the classification/regression results and a hidden layer (or more) that interconnects the input layer to the output layer. Apart from this, MLPs correspond to the feedforward ANN family, that is, information only flows forwards from inputs to outputs and there is no feedback loop between units. The following Figure 3.18 shows an example of an MLP with one layer of each type (input, hidden and output).

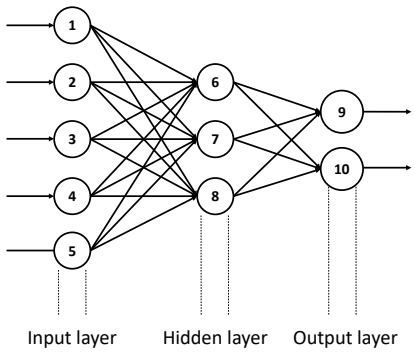

Figure 3.18: Multilayer perceptron with one input, hidden and output layer.

To know how backpropagation algorithms work, it is important to understand properly the inner structure of a unit or neuron. For example, looking at neuron 6 from Figure 3.18 it is easy to deduce that it gets its inputs from all the nodes in the input layers. However, connections between neurons are not simple. Instead, the connections are weighted, meaning that the output value of any preceding neuron will be multiplied by a certain connexion weight before entering the next layer's input. This can be seen in the zoomed Figure 3.19 of the already mentioned neuron 6. Also, there is another value being used as input to the neuron, Θ, which is known as neuron bias. Finally, after all the weighted inputs and the bias have been summed, the resulting value goes into the so-called activation function. The activation function is the one in charge of transforming the summed input into the output value of the neuron (that will subsequently be used as input for all the next layers' inputs). Traditionally, MLPs use the sigmoid function, which is continuously differentiable and non-linear. Thanks to that, MLPs can give solution to non-linear problems.

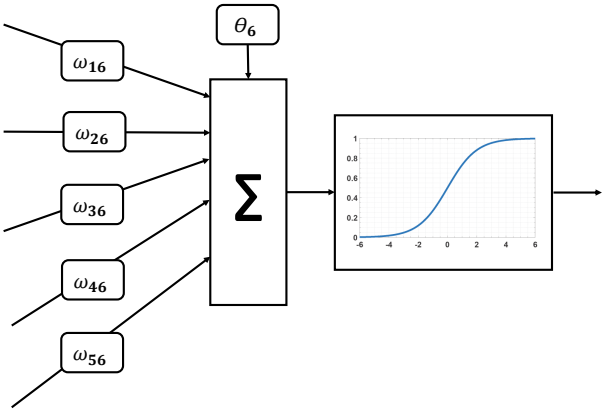

Figure 3.19: Inside structure of neuron 6, whose connections to its precedent layer neurons is weighted.

Having explained the inside structure of neurons, it is now possible to get deeper into the MLP training and backpropagation. First of all, the topology of the MLP has to be chosen. The second step is to initialise all ω_{ij} weight and Θ_j bias values. This is normally done by giving them a random value in the [-1,1] range. The learning rate l needs also to be fixed in the (0,1) range. Finally, as all other supervised algorithms, the MLP has to be trained iteratively.

For the sake of understandability, the training process will be divided into two sequential blocks: the input forwarding process will be given in algorithm 1, and, later, the backpropagation will be presented in algorithm 2. These algorithms focus on a specific layer, i and k respectively being the number for a neuron in the previous and posterior layer. I and K stand for the number of units/neurons in those layers. Lastly, j and J have the same meaning but for the layer in which the algorithm is focusing on. Despite these algorithms have a complex mathematical formulation, applying the algorithms is not as complicated as it seems at first sight. Along with a more detailed explanation, it is possible to find a very clarifying numerical example in [46].

It is important to mention that the MLP is a basic form of ANN. However, there are many variations that create new types of NNs allowing to tackle problems from different perspectives. For example, being very similar to MLPs, Radial Basis Function networks (RBF) use Radial Basis Functions for the activation function instead of using the sigmoid function [47].

Finally, although using them is not difficult, it is complex to understand how ANNs attempt classifications. Also, training these algorithms implies high computational load and it is also time-consuming. In fact, as convergence of the learning is not guaranteed, it is common practice to establish a limit of iterations for the training. Nevertheless, in the last years, new techniques have been discovered allowing to considerably speed up the training of networks, as using GPUs for the linear algebra calculations.

Algorithm 1 Input forwarding and neuron output calculation.

1: **for** each input unit **do**
2: $Out_i = In_i$;
3: **end for**
4: **for** every neuron in each hidden layer **do**
5: Get the summed input of the neuron:

$$In_j = \Theta_j + \sum_{i=1}^{I} Out_i \cdot \omega_{ij}$$

6: Get the output of the neuron using the sigmoid activation function:

$$Out_j = 1/(1 + e^{-In_j})$$

7: **end for**
8: **for** each output neuron **do**
9: Get the summed input of the neuron using the same formula of line 5.
10: Get the output of the neuron using the same formula of line 6.
11: **end for**

Algorithm 2 Backpropagation algorithm.

1: **for** each output neuron **do**
2: Calculate the error:

$$\mathcal{E}_j = Out_j \cdot (1 - Out_j) \cdot (Class_{instance} - Out_j)$$

3: Calculate the weight updating increment: $\Delta\omega_{ij} = l \cdot \mathcal{E}_j \cdot Out_i$
4: Update ω_{ij} weights: $\omega_{ij-new} = \omega_{ij} + \Delta\omega_{ij}$
5: Calculate the bias updating increment: $\Delta\Theta_j = l \cdot \mathcal{E}_j$
6: Update bias Θ_j: $\Theta_{j-new} = \Theta_j + \Delta\Theta_j$
7: **end for**
8: **for** each hidden layer **do**
9: Calculate the error:

$$\mathcal{E}_j = Out_j \cdot (1 - Out_j) \cdot \sum_{k=1}^{K} \mathcal{E}_k \cdot \omega_{jk}$$

10: Calculate the weight updating increment: $\Delta\omega_{ij} = l \cdot \mathcal{E}_j \cdot Out_i$
11: Update ω_{ij} weights: $\omega_{ij-new} = \omega_{ij} + \Delta\omega_{ij}$
12: Calculate the bias updating increment: $\Delta\Theta_j = l \cdot \mathcal{E}_j$
13: Update bias Θ_j: $\Theta_{j-new} = \Theta_j + \Delta\Theta_j$
14: **end for**

Support vector machines (SVM)

SVM algorithms base on an easy principle, maximising the marginal separation between two classes. For the explanation, an example will be given in Figure 3.20 for a two-dimensional space. Nevertheless, despite it is not possible to see it graphically, the same principles can be applied to multidimensional cases.

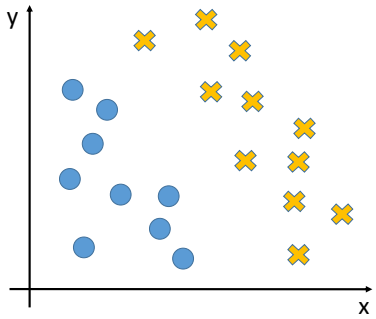

Figure 3.20: In this case, instances of either classes are linearly separable.

In the example of Figure 3.20, it is very clear that the instances of the two classes can be very easily separated. In cases like this, SVM algorithms calculate the hyperplane of dimension $n-1$ that best separates the classes in a dataset. In the case of a two-dimensional space like the one of the example, the dimension of the hyperplane would be $2-1=1$, i.e., a straight line (as seen in Figure 3.21).

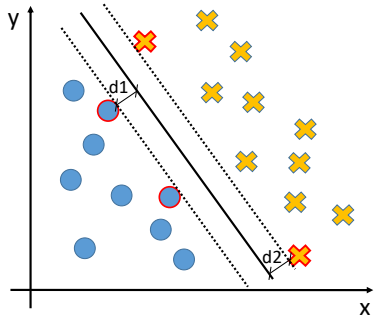

Figure 3.21: In a two dimensional space, instances can be linearly separated by a straight line. The optimum case is where the margin between classes is maximum, i.e., when d1=d2.

There are infinite hyperplanes that could be used to separate between classes but there is only one solution that maximises the distance between the closest instance of a class and the hyperplane. In a two dimensional space, the best

hyperplane is the line in the middle of the biggest separation possible between the two classes (in Figure 3.21 it is the line that makes $d1 = d2$). The instances that are closest to the hyperplane get the name of "support vector" (highlighted in red in Figure 3.21).

Analytically, it is possible to define the maximum margin hyperplane [48] that maximises the separation between support vectors using a similar approach to the Hesse normal form of a plane (equation 3.15).

$$\vec{\omega} \cdot \vec{x} - b = 0 \tag{3.15}$$

Recalling equation 3.15 from the field of geometry, $\vec{\omega}$ corresponds to the normal vector to the new hyperplane, b to a point in that hyperplane and $\vec{x}$ to the instance features. However, here it is not necessary to normalise $\vec{\omega}$. This hyperplane would correspond to the continuous line of Figure 3.21.

Using this analytical formula, it is possible to define two more hyperplanes, parallel to the original, which will be held by the support vector instances (the dashed lines of Figure 3.21). These hyperplanes are defined so that every $\vec{x}_i$ instance above one of them belongs to a class (namely $\vec{y}_i = 1$ in equation 3.16). Similarly, the other hyperplane is set in a way that any $\vec{x}_i$ instance below it belong to the other class ($\vec{y}_i = -1$, equation 3.17).

$$\vec{\omega} \cdot \vec{x}_i - b \geq 1 \rightarrow \vec{y}_i = 1 \tag{3.16}$$

$$\vec{\omega} \cdot \vec{x}_i - b \leq -1 \rightarrow \vec{y}_i = -1 \tag{3.17}$$

The distance between these two hyperplanes can be defined as $\frac{2}{||\vec{\omega}||}$. Hence, minimising $||\vec{\omega}||$ produces the margins to be maximum. Once solved that optimisation problem, when features of a new instance are fed into the classifier, it will be the sign of the result obtained from $\vec{\omega} \cdot \vec{x}_i - b$ that will determine if the new instance belongs to a class or the other. If the sign is positive then $\vec{y}_i = 1$ and, if not, $\vec{y}_i = -1$.

When the classification problem belongs to a higher-dimensional space, SVMs operate the same way: they try to linearly separate the instances of a group from the other. That being said, there are cases in which classes are not linearly separable. For those cases, there are different approaches as determining soft margins or applying kernel transformations. Applying soft margins is a simple technique that allows some tolerance when defining the hyperplane and the margins so that it allows a limited amount of instances to be misclassified (see Figure 3.22).

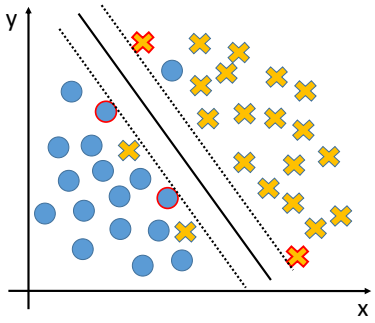

Figure 3.22: The application of the soft margins permits to separate classes easily at the cost of certain error.

On the other hand, the application of kernel transformations consists on transforming data using certain mathematical operations so that non linearly separable class instances are translated to another space in which they can be linearly separated (a visual example is presented in Figure 3.23). Depending on the distribution of the instances, different Kernel transformations permit better or worse solutions [49]: linear transformations, polynomial transformations, radial basis transformations...

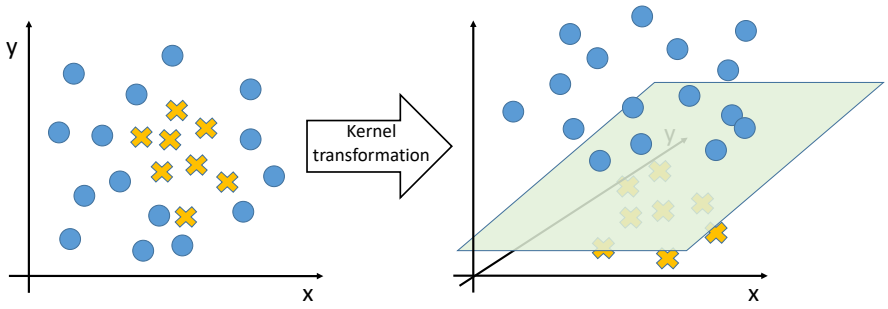

Figure 3.23: Thanks to the kernel transformation the instances from the two classes can be linearly separated.

K nearest neighbours (K-NN)

These algorithms are based on calculating the proximity of an instance to the previously classified instances [50]. Once the similarity distances have been calculated (normally, the Euclidean distance), the algorithms start a voting process in which the new instance will be classified as belonging to the class with a higher number of instances close to it.

However, not all instances are taken into account but only the ones closest to the new instance. K is the parameter that defines how many of the closest instances will be taken into account for the voting. This, choosing a proper K parameter is very important as it might affect the classification result. It is possible to see an example of this phenomenon in Figure 3.24, whereas choosing $K = 3$ results in a successful classification (left side of the figure), choosing $K = 5$ produces a wrong classification result (right side of the figure).

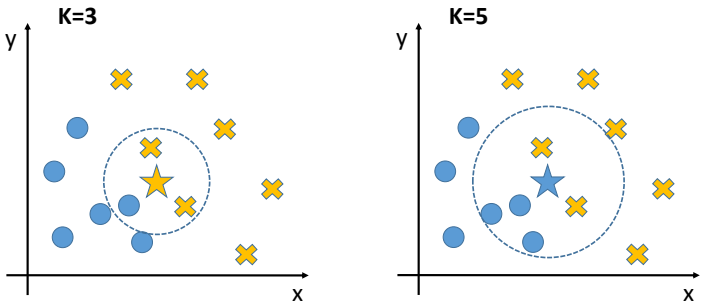

Figure 3.24: The class of the new instance (star) might vary depending on the setting of K.

For the sake of practicality, some researchers suggest using $K = \sqrt{n}$, where n is the total of instances in the training dataset. Other researchers suggest going on the "trial and error" process using cross-validation (it will be explained later in Subsection 3.4.3). However, there are modifications to the K-NN algorithm that cope with reactivity to noise due to a bad set of K, as giving higher importance weights to the closest instances so that the furthest have a lower influence in the classification [51], [52].

Decision trees (DT)

DTs are flowchart looking structures that permit the classification of data. From the initial node, named "root node", new branches derive to an evaluation point for a certain feature of the instances. Depending on the result of the evaluation of the feature, the node branches again into more nodes repeating this process until a final node is reached. These final nodes, where the class of the instance is given, get the name of "leaf nodes". It is because of this branching structure from the root to the leaves similar to a tree that these type of algorithms receive their name: decision trees (DT). To illustrate the concept, a simple DT is presented in Figure 3.25 for determining the relationship of a certain person in a family to a newborn infant of the same family.

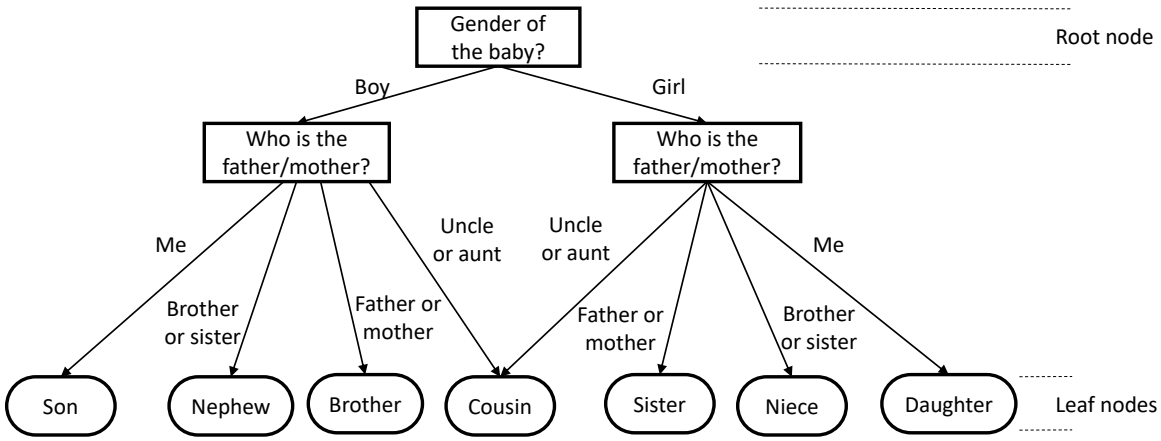

Figure 3.25: A DT structure to obtain the relationship to a newborn infant in the family.

The induction of a DT structure is an iterative process for which three inputs are needed [53]: the dataset D with the instances, the classes C of the training data instances and the branch splitting method. For the sake of intelligibility, the branch splitting method will remain unexplained for the moment. The iterative process to create a DT has the steps indicated in Algorithm 3.

The pseudo-code of Algorithm 3 is recurrently repeated branching the tree downwards until all the instances in a partition belong to the same class or if there are no attributes left to describe the instances. Also, when prepruning is not used (explained later), the branching stops if a splitting criterion creates a new partition with no instances in it, i.e., an empty partition. When this happens, the new branches are discarded and node N would be labelled as the class to which the majority of instances in the previous partition belonged to. When a branching process is stopped, then the algorithm goes to another branch and repeats the same thing with the pending data to be branched.

So far, the construction of DTs has been explained. Nevertheless, the branch splitting method still remains unexplained. What it does is to find the feature that permits to split the dataset in the most homogeneous manner. Once found, it chooses the best splitting point for the feature and divides the dataset into smaller subsets accordingly. These methods vary depending on the type of tree that is going to be used: ID3 trees use the "Information Gain" method, C4.5 trees use the "Gain Ratio" method, CART trees use the "Gini Index" and so on for other topologies.

Finally, once the tree induction is finished, sometimes the amount of branches needs to reduced as it makes the DT to be more sensitive to noises and abnormalities. Continuing with the plant-related analogy, this process is called postpruning. Pruning normally is started from the bottom branches of the tree

Algorithm 3 Decision Tree induction algorithm.

```
1: while stop_criterion == FALSE do
2:   Create a node N with the remaining D data partition.
3:   if all instances in D belong to the same class == TRUE then
4:     Label node N as a leaf node corresponding to that class;
5:     stop_criterion = TRUE;
6:   else
7:     if are there features left for the instances? == TRUE then
8:       Choose the best feature for splitting the branch and divide D accordingly;
9:       Remove the selected feature from the newly divided partitions of D;
10:      if any new partition is empty == TRUE then
11:        Label node N as a leaf of the majority class of the original D;
12:        stop_criterion = TRUE;
13:      end if
14:    else
15:      Label node N as a leaf of the majority class in the remaining data;
16:      stop_criterion = TRUE;
17:    end if
18:  end if
19: end while
```

and goes upwards. There are different methods to consider whether a branch should be pruned or not. For instance, cost complexity and pessimistic pruning methods compare the current tree error rate against the error rate that would be produced if branched node was substituted by a leaf node of the prevalent class. The tree would get pruned in the case that the error rate of the pruned branch was smaller. If not, the tree would be left as it is. It is also possible to prune a tree during its construction by just stopping the branching process when the number of instances going through the branches is not significant. This type of pruning is called prepruning.

To finish, an interesting property of DTs is that they are easy to understand for humans. Hence, DTs can be easily transformed to rule-based classifiers, such as, for example, the one rule algorithm (1R). Also, it is possible to create random forest (RF) classifiers, which are nothing but a set of uncorrelated DTs that give the definitive classifying result by voting with the outputs of each of the independent DTs. These uncorrelated DTs can be easily trained by modifying the training data with the bootstrap method (very similar approach to the bagging method explained in the following paragraphs). This way, it is possible to overcome one of the disadvantages of DTs: the high sensibility to noise in data.

Ensembles of classifiers

Although they are not a specific classifier type on their own, creating ensembles of classifiers is a general method to improve the accuracy of classifications and predictions. What ensembles of classifiers methods propose is no other than combining different mathematical models to attempt the classification or regression. Bagging and boosting methods are amongst the most popular ensemble methods [53].

Bagging consists on the similar principle as getting a second opinion from an expert, which is not something uncommon in real life. For example, in health-related problems, it is common that patients ask more than one doctor for their opinion and to take as valid the opinion that was repeated more times. Therefore, bagging methods consist in creating various classification models and then selecting the most frequent output (see algorithm 4). Bagging classifiers implies training several classifiers of the same algorithm type by using new datasets using the bootstrap method (this method is explained in Subsection 3.4.3). Once the desired amount of models have been trained, the only thing left is to feed the same instance to all those models and vote for the majority class.

Algorithm 4 Bagging algorithm.

```
i=1;
while i < The desired amount of models do
    Create a new dataset D using bootstrap sample and replace method;
    Train a M_i model using the new dataset;
    i=i+1;
end while
Use all M models to classify the new inputs and choose the most frequent class.
```

Boosting is another technique to improve the results of a problem using several models. However, unlike bagging, boosting methods do not give the same importance to all the classifiers but they weigh them depending on their performance: the decision of the best classifiers have greater weight when it comes to voting. Besides, boosting methods also vary the weight of the inputs according to their ease of classification. In other words, if an input is misclassified then its weight increases. On the contrary, if it was correctly classified then the weight is reduced. In this manner, boosting methods give more importance to the inputs that are more difficult to classify so that the performance for those inputs is improved.

One of the most famous boosting methods is the Adaboost method (short form for adaptive boosting) [54]. The pseudo-code of algorithm 5 summarises such method for a binary class classification.

Algorithm 5 Adaboost boosting algorithm.

1: i=1;
2: **while** i < The desired amount of models **do**
3: Initialise ω_i instance weights so that $\omega_i = 1/D$, being D the size of the database;
4: Train a M_i model using a new dataset created with sample and replace method;
5: Compute $\mathcal{E}$ error rate: summation of the weights of the misclassified instances;
6: **if** $\mathcal{E} > 0.5$ **then**
7: Discard M_i and restart the model training from step 1;
8: **else**
9: Update weights: $\omega_{i-new} = \omega_i \cdot (\mathcal{E}/(1-\mathcal{E}))$
10: **end if**
11: Normalize the weight of the instances;
12: i=i+1;
13: **end while**
14: Use all M models to classify the new inputs giving a ω_i weight to each model for the later voting, being $\omega_i = \log \frac{1-\mathcal{E}_{M_i}}{\mathcal{E}_{M_i}}$.

3.4.2 Unsupervised learning

Unlike supervised learning algorithms, unsupervised learning algorithms do not need to know the class of the instances to learn a model. Instead, they learn on their own from the data characteristics and they group data accordingly without any outer help. Thus, they are especially useful when labelling data is costly or very time-consuming. In this thesis, only the K-means clustering algorithm is going to be explained, as it has been the only the unsupervised learning method that has been used in this work.

The objective of clustering methods is to find similarities inside data so that instances are gathered in groups sharing similar properties, and, at the same time having different properties to the instances in other groups. However, depending on the algorithm, similarity can be measured using different approaches. More precisely, this document will focus on the K-means clustering method.

K-means

The K-means algorithm is a clustering method that uses distances to decide how instances will be grouped into different clusters. In the case of K-means, parameter K stands for the number of clusters in which data will be grouped; whereas if $K = 3$ all data will be gathered in three big clusters, setting $K = 5$ will produce the creation of two clusters/groups more. Hence, K is a parameter that has to be defined by the user.

The operation of K-means is simple: it moves the centroid of the K clusters

looking to minimising the distances between those centroids and all the instances belonging to the cluster of the centroid. To do so, first of all, the algorithm initialises the position of the centroids by randomly choosing the coordinates of K instances in the dataset. After that, the iterative process starts (see algorithm 6 for a n dimensional case): first, all instances are considered as belonging to the closest cluster centroid; then, centroids are recalculated to minimise the sum of the squared errors for the euclidean distances of all the instances belonging to it. It is possible to find a numerical example of the algorithm in [55]. This process is repeated until no instance change from a cluster to another or when the algorithm has gone over N_{lim} iterations.

It is interesting to note that K-means clustering depends considerably on the initial position of the cluster centroids. This way, it is probable to obtain different clustering result if the initial centroid positions are modified. Thus, it is common to run the K-means algorithm several times and to pick the best performing after comparing the results for all the runs. However, this increases the computational and time costs. Also, there are techniques that try to deal with this problem by selecting the best initialisation parameters [56].

Algorithm 6 K-means clustering algorithm.

1: Given K, initialise K centroids with the coordinates of K randomly chosen instances;
2: $n_{iterations} = 1$;
3: **while** stop_criterion == FALSE **do**
4: Compute the distances for each i instance of n attributes to all k cluster centroids:

$$distance_{ik} = \sqrt{(x_{1k} - x_{1i})^2 + (x_{2k} - x_{2i})^2 + \cdots + (x_{nk} - x_{ni})^2}$$

5: Assign instance i to the cluster for which $distance_{ik}$ was minimum;
6: **if** (instances_changed_cluster == FALSE) **or** ($n_{iterations} > N_{lim}$) **then**
7: stop_criterion = TRUE;
8: break;
9: **end if**
10: Compute the new coordinates of the k cluster centroids (it depends on the number q of instances into cluster k):

$$(x_{1k}, x_{2k}, \cdots, x_{nk}) = \left(\sum_{i=1}^{q} \frac{x_{1i}}{q}, \sum_{i=1}^{q} \frac{x_{2i}}{q}, \cdots, \sum_{i=1}^{q} \frac{x_{ni}}{q} \right)$$

11: $n_{iterations} = n_{iterations} + 1$;
12: **end while**

3.4.3 Data partitioning for validation

When the performance of the classifier has to be evaluated, as it has already been said, it is important to leave all the training data apart and to only use testing data that has remained unused. Thus, it is the testing dataset the one that enables to determine the error and success rates of the algorithms. This subsection will present some of the methods that are used to partition data for calculating the performance of the algorithms: hold-out, bootstrap and K-fold cross-validation.

Hold-out method

The hold-out method is the simplest of all the data partitioning methods. It consists in randomly dividing data into two parts, training and testing datasets. Normally it follows the approach of using 2/3 for training a model and the remaining 1/3 for testing its performance.

There is also a slight variation of the hold-out method called random subsampling which repeats the hold-out process a k number of times. Then, the k different datasets are used to calculate k models and subsequently test them with the k testing datasets to get the performance metrics. Finally, the average performance metric is calculated using those previous k values.

Bootstrap selection

The bootstrap method is based on the "sampling and replace" approach. The principle is simple, iteratively an instance is selected from the whole database to create the training set. However, the selected instance is not removed from the database but it is kept so that it is possible to reselect it in the next iteration. Repeating the process as many times as the number of instances contained by the database is a common approach named 0.632 bootstrap selection (the following two paragraphs will explain it).

Clearly, some instances will get repeated in the training set and some others will not be used for such a manner. All instances have always the same probability of being chosen for the training, which is $P_{training} = 1/n$, being n the size of the database. On the contrary, the probability for not being chosen for the training (or said in another way, being chosen for the test) is $P_{testing} = 1 - 1/n$. If, as said before, the process is repeated for n iterations then the probability of an instance for being in the testing dataset is $P_{testing} = (1 - 1/n)^n$ which approximates to 0.3678. And, subsequently $P_{training} = 1 - P_{testing} = 0.632$.

Finally, to asses the error rate it is necessary to compensate for those biasing towards the training set probabilities by multiplying the training and testing error rates by the probability of the non-corresponding set [57]. Numerically, it would be represented as $\mathcal{E} = \mathcal{E}_{training} \cdot 0.3678 + \mathcal{E}_{testing} \cdot 0.632$.

K-fold cross-validation

The last data partitioning method that is going to be explained in this thesis is the K-fold cross-validation method. This method divides the database in K equally sized parts, all of them containing instances that are exclusive to their specific partition. Then, $K-1$ partitions are used to train the data and the remaining partition is used for testing. This process is repeated K times, sequentially changing the partition used for the test: in the first run is partition 1, in the second is partition number 2, and so on. Finally, as done for random subsampling, the performance is calculated by averaging the results obtained from the K repetitions.

Normally, K is set so that it is 10 or another number close to that, such as 5 or 20. When using this method, apart from setting K, it is also important to stratify data. This means that all the partitions should have a similar amount of instances for each of the classes in the database. Otherwise, the models can be biased towards one class. This approach enables the use of 90% of the data for the training but it has the counterpart that it is more computationally costly compared to the other methods.

Finally, the most extreme variation of the K-fold cross-validation is known as "leave-one-out cross-validation", which is when just a single instance is used for the test in each fold. It is very computationally costly as almost all instances are used for training. Because of this, its use may not be feasible for very large datasets. Nevertheless, it can be very effective smaller datasets.

3.4.4 Performance metrics

So far, some partition methods have been explained. During their explanations, it has been mentioned that the way of calculating the performance metrics has to be modified. However, these performance metrics have not been explained yet. This subsection will focus on presenting the most commonly used performance metrics which can be useful for comparing how well different classifiers work for a given problem. Most of the performance metrics are calculated by operating the following indexes of the classification, taking a class a the reference class:

- **True positives (TP):** the instances that have been correctly classified for the reference class.
- **False positives (FP):** the instances that have been classified as belonging to the reference class but that did not belong to it.
- **True negatives (TN):** the instances that have been correctly classified as not belonging to the reference class.
- **False positives (FN):** the instances that have been classified as not belonging to the reference class but belonged to it.

Accuracy and error rate

These two performance metrics are complementary to each other. Whereas the accuracy stands for the percentage of correctly classified instances, the error rate stands for the percentage of misclassified instances. Hence, both indicator range from [0-1]. Accuracy and error rates are respectively described by the expressions 3.18 and 3.19.

$$acc = \frac{TP + TN}{TP + TN + FP + FN} = \frac{T}{T + F} \tag{3.18}$$

$$err\ or\ \mathcal{E} = \frac{FP + FN}{TP + TN + FP + FN} = \frac{F}{T + F} = 1 - acc \tag{3.19}$$

Confusion matrices

The confusion matrix is a table in which all the classes are placed as row and column headers. There, the class in each row is considered to be a reference class and the columns stand for the class stated by the classifier. Thus, cells provide the number of instances of the reference class that have classified as belonging to each of the classes. Looking at the example of Table 3.4 can be helpful to understand how confusion matrices work.

Table 3.4: Confusion matrix of a 330 instance database. There are 110 instances for each of the 3 classes.

	Class 1	Class 2	Class 3
Class 1	100	10	0
Class 2	5	85	20
Class 3	10	20	80

Looking at the confusion matrix of Table 3.4 it is possible to see that the classifier that was used for the task obtained the best results for the first class, where 100 instances were correctly classified and only 10 were misclassified. On the other hand, having the highest number of incorrect classifications, the classifier found the third class to be the most difficult to identify.

Apart from giving a good view of how was the overall performance of the classifier, confusion matrices are also useful to compare the performance of various classifiers related to the costs of misclassification. Sometimes, missing a class is much worse than missing the classification of another class [43]. A clear example of this is found in cancer diagnosis, where it is much better to have FP rather than FN. A FN would mean not detecting a cancer, which may have fatal consequences for the patient as quick detection is critical for cancer treatment. On the other hand, a FP would not have such a critical impact as it would mean that the patient would be treated as having a cancer and, for sure, later diagnostic test would discard the false hypothesis. Thus, although both

diagnostic errors produce very undesirable effects, FNs have an infinitely higher personal cost.

For sure, the cost of a single human life is too high to account for. However, when the most costly option is not that clear, it can be somehow unclear to know which classifier is better for an application. In those cases, the total of misclassified instances are multiplied by the cost of each type of classification failure to choose the classifier whose errors have a smaller cost. This can be seen in the example of the diagram of Figure 3.26, where using the first classifier is the best option despite it has more errors than the second classifier.

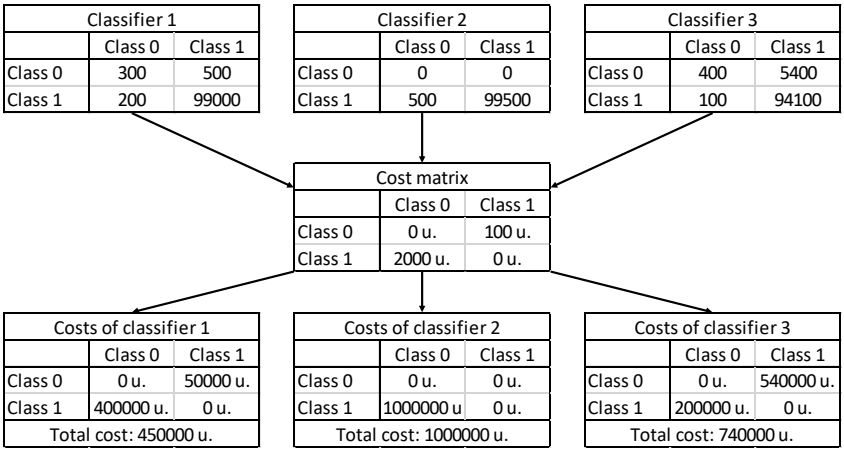

Figure 3.26: Although classifier 2 has less errors, the error associated costs reveal that the best option is using classifier 1.

Precision, recall, specificity and F-measures

As it has just been explained, sometimes it is important to look deeper into the type of errors that are being done. In this sense, accuracy and error indexes can be misleading as it happened in the example of Figure 3.26. Because of this, a new set of performance indicators were designed each of which focus on a specific part of the classification.

- **Precision (pr):** it stands for the percentage of true positives among all the instances classified as being positive.

$$pr = \frac{TP}{TP + FP} \tag{3.20}$$

- **Recall (re):** it stands for the percentage of true positives among all the instances that have a positive class in the dataset. It indicates the ability

of the classifier for detecting positive cases.

$$re = \frac{TP}{TP + FN} \tag{3.21}$$

- **Specificity (sp):** it stands for the percentage of true negatives among all the instances that have a negative class in the dataset. It gives an indication of the ability of the classifier for avoiding false alarms.

$$sp = \frac{TN}{TN + FP} \tag{3.22}$$

- **F_β-measures:** it is the index that gives a balance measure between precision and recall. If one of the two is high but the other is not, then the F_β-measure will be low. Typically, precision and recall are given the same importance by setting $\beta = 1$, which results in the F_1 index. This can be modified depending on the needs: if β is increased then precision gains importance and the opposite happens when it is decreased.

$$F_\beta = (1 + \beta^2) \cdot \frac{pr \cdot re}{(pr \cdot \beta^2) + re} \tag{3.23}$$

  F-measures are especially useful when precision and recall might give tricky information. For instance, if $pr = 1$ and $re = 0.7$ it might seem that the algorithm is very good, but $F_1 = 0.8235$, which indicates that the algorithm might not be as good as thought when looking at the precision.

3.4.5 Other tools frequently used

This subsection will present two tools more that have been used for tasks related to machine learning but that they may not totally fit in any of the other subsections. The first tool corresponds to the Wilcoxon test, which is used for analysing the statistical significance of the results. The second tool presented in this subsection is the correlation based feature selection method. This second tool is not a machine learning algorithm, but it is often used to simplify the classification task helping classifiers do their work.

The Wilcoxon test

The Wilcoxon test [58] is a non-parametric statistical test that is useful to study the relationship between matched samples. Therefore, it can be used to determine whether there are statistically significant differences in the results obtained by two different classifiers. One of the advantages of this test is that, being non-parametric, the samples that are compared do not have to belong to any specific statistical distribution. Although it is going to be explained in the following paragraphs, many mathematical computer applications have built-in

functions that automatically compute the test given the samples to be compared, such as "signrank(x,y)" in Matlab ® or "wilcox.test(x,y, paired=TRUE)" in R.

To compute the Wilcoxon test between datasets D_1 and D_2, first, the samples of both dataset are arranged in pairs. Then, the absolute difference between the samples in each pair is calculated with the following subtraction: $|X_{D_2 i} - X_{D_1 i}|$. In this step, if the absolute difference equals 0 then the pair giving that result should be discarded, subsequently reducing the size of the data to be compared.

The next step it to rank increasingly all the absolute differences that have not been removed. If there is any repeating absolute difference then all the pairs giving the same value are ranked with the same position, which corresponds to the average of all the positions they would occupy. For example, if there were three repeating values getting the 4^{th}, 5^{th} and 6^{th} positions, then all of them would be ranked as being in the 5^{th} position. Namely, let the position of each pair in the rank be identified as R_i.

Finally, knowing the ranking, it is possible to calculate the test statistic using equation 3.24, and, then, to calculate the p-values. In this sense, p-values below 0.05 indicate that there is less than a 5% of probability for the results to be random and, so, they can be considered to be statistically significant.

$$W = \sum_{i=1}^{N} sign(X_{D_2 i} - X_{D_1 i}) \cdot R_i \tag{3.24}$$

Correlation based feature selection

It is not uncommon that datasets contain redundant or noisy information. When these situations take place it is more complicated for machine learning algorithms to extract the most valuable information. Moreover, the problem of having redundant and noisy information can have a great impact on the classification performance. DTs are a good example of this phenomenon as they try to split data using the most meaningful attributes of data. Thus, it is very important for some algorithms to clean data before classification, i.e., to select the most meaningful features. Even more, algorithms that are relatively robust to noises and redundancies also benefit from feature selection as thanks to it they can perform faster. Hence, it can be concluded that applying feature selection to reduce the dimensionality of the datasets is most of the times beneficial.

While developing the work of this thesis, when needed, the feature selection was done with the Correlation-based Feature Selection (CFS) method presented in [59]. The method calculates the $Merit_s$ value of a S subset of features as an indicator of how good they predict the class and, at the same time, giving indication of how redundant those features are. Following this approach, the most significant feature subset is the one getting the highest $Merit_s$ value. The $Merit_s$ index is calculated by equation 3.25, where k is the number of features in subset S and r_{cf} and r_{ff} stand for the average feature-class and inter-feature

correlations respectively.

$$Merit_s = \frac{k \cdot r_{cf}}{\sqrt{k + k \cdot (k-1) \cdot r_{ff}}} \tag{3.25}$$

As stated in [59], the application of this approach follows the heuristic best-first searching strategy. First it starts with an empty S subset and it goes on adding new features and evaluating them one by one. In this manner, it is important to prevent the algorithm from analysing all the possible feature combinations. To do so, the algorithm is told to stop when no improvement is obtained after getting five consecutive fully expanded subsets.

3.5 Brain-Computer Interfaces (BCI)

Brain-computer interface (BCI) systems are those technological systems that sample and process the electrical activity happening in the brain so that final users can interact with their environment (for example, controlling a computer or a robotic mechanism). Thus, thanks to BCI applications users can avoid using their muscles to control those end-effectors, which is something of special interest for people with certain special needs. Apart from a headset to collect the EEG data, all BCI systems use some kind of machine learning algorithm that is used to determine whether the EEG patterns taking place in the brain correspond to the user wanting to command a control order or not. Hence, the discipline of BCI is strongly related to the techniques presented in Section 3.4.

Throughout this section the dissertation will explain some of the most common principles in which different types of BCI are based on: event-related potentials (ERPs), steady-state visually evoked potentials (SSVEP) and event-related desynchronisation/synchronisation (ERD/ERS) motor imagery systems will be respectively explained in Subsections 3.5.1, 3.5.2 and 3.5.3.

3.5.1 Event related potentials (ERPS)

These type of BCI systems base on having the user focused on detecting a specific event. While the user is waiting for the event to happen, the brain activity of the user remains relatively stable. However, when the target event happens there is a significant alteration on the EEG signals recorded by the electrodes placed in the area related to that given event. For instance, if the event to happen is an audio stimulus, whenever the target sound is sensed it will produce an alteration of the EEGs collected from the auditory cortex (located at the lateral part in both sides of the brain). In this sense, various types of stimulation can be used for evoking ERPs: visual, tactile, auditory, electrical...

Concerning ERPs, the EEG alterations produced by these stimulations differ depending on the properties of the stimulation used. However, they share a common property and it is that there is always a latency between the stimulus sensing and the EEG alteration. This means that there is a difference in time between the moment in which stimulus takes place and the highest (or

lowest) signal peak that is produced by the EEG alteration. Hence, knowing the morphology of the EEG alteration is a key point for designing a machine learning algorithm that determines if the target stimulus has been sensed, and, subsequently, trigger a control command of a given end-effector.

In this sense, there is certain consensus in the naming of ERPs. First, "P" or "N" letters are used to describe the direction of voltage deviation produced standing for positive or negative deviation. Additionally, a number is used referring to the amount of time (in milliseconds) that the voltage peak needs to appear on the EEG channel. For example, if the deviation goes negative in voltage and needs around 200ms for the peak to show up, then ERP would get the name of N200. However, these times are just to be used as a reference as they are not deterministic and ERPs might vary their timing [60]. Among the different types of ERPs, P300s are one of the most used, as it happens in the speller application.

The speller BCI application is, as its name indicates, a system that allows the user to write letter by letter following a process similar to spelling [61]. Normally, letters and numbers are presented in a screen using a matrix format so each letter or number occupies a single cell [62]. Then, while the application is running, the BCI system highlights the letters and numbers in a controlled manner (see Figure 3.27, extracted from [63]). At the same time, the user will pay attention only in the letter or number that needs to be written. Thanks to this, the BCI system can detect a P300 ERP and associate it to the highlighting of the desired letter, subsequently writing it.

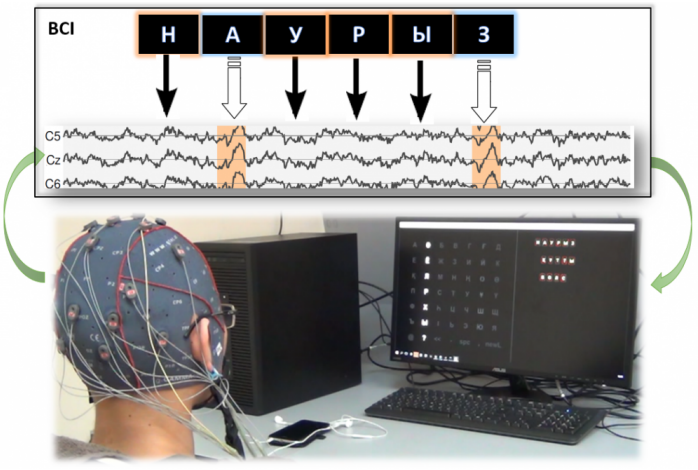

Figure 3.27: It is possible to see a P300 potential after the target letter has been highlighted.

3.5.2 Steady state visually evoked potentials (SSVEP)

Another way to induce electrical potentials in the brain is by making the user look continuously at a light or spot flashing at a specific frequency. When this is done, it is possible to detect how a resonance of the same frequency builds up in the cortical areas related to visual processing, i.e., occipital and back part of the parietal cortex [64]. This resonance will be maintained steadily (as the name of the techniques states) until the user stops looking at the flickering stimulus. Hence, if different parts of a screen are set to flicker at different frequencies, then it is possible to trigger user-determined control commands by arbitrarily associating the EEG resonances of those frequencies to the desired user commands. Figure 3.28 shows an example of how the flickering frequency builds up with different powers in the different EEG channels (figure extracted from [65]).

One of the main benefits of this type of systems is that it is relatively easy to detect the frequency resonances using Fourier transforms and calculating the power spectral density. Also, it permits the user to have many different commands as it is possible to detect many different frequencies, such as the 12 different classes were detected in [66]. On the contrary, despite wearing an EEG headset is considered not to be invasive, looking at flashing points is a relatively aggressive method and it produces user fatigue easily.

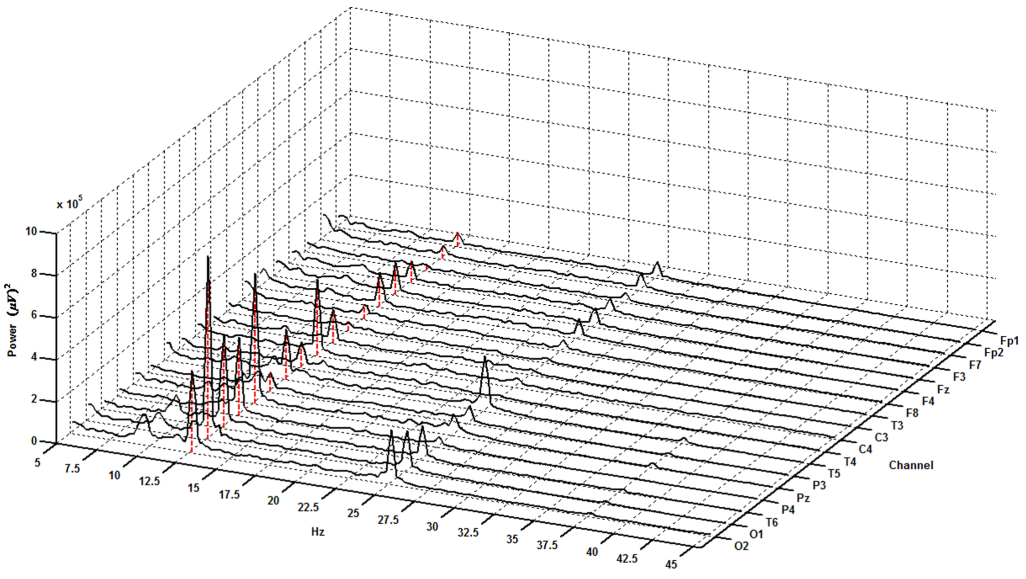

Figure 3.28: The 12.5Hz frequency marked in red corresponds to the stimulus flashing frequency spotted by the user.

3.5.3 ERD/ERS motor imagery

The last BCI method/application explained in this thesis corresponds to the event-related desynchronisation (ERD) and event-related synchronisation (ERS) motor imagery. The ERD/ERS parameter stands for the measure of the decrease or increase of the power in a given frequency band during the period of given event [60], [67]. In the case of ERD/ERS based motor imagery BCI, the affected frequency bands are β and μ. As mentioned in Subsection 3.2.4, these two bands get suppressed when moving a limb [68]. However, it does not only happen when moving a part of the body but also when the user imagines to move it. This is why this type of BCI is called like that.

Taking advantage of this phenomenon, machine learning algorithms make it possible to detect patterns in the EEG channels that are more strongly related to the movement of the target limb. Hence, knowing which part of the brain takes control of the movement of each of the parts of the body is very important. For example, it is the C3 channel (left hemisphere) that gets more strongly suppressed when thinking of moving the right arm. On the contrary, whereas C3 remains unaffected, it is C4 in the right hemisphere that gets suppressed while thinking of moving the left arm. Therefore, the limbs of one side of the body are controlled by the motor cortex of the contralateral hemisphere. However, although it is relatively easy to detect the EEG patterns related to the arms and hands, it is important to note that doing the same for the legs is rather complicated. This is because the area of the motor cortex that controls one leg is very close to the area that controls the other, and so, the collected EEG signals are very similar and difficult to differentiate.

Using this BCI methodology is very interesting because unlikely it happened for ERPs and SSVEP, it does not require any outer stimulus as it is the user himself who produces the EEG pattern using the imagination. On the contrary, it requires long training times, not only for the machine learning algorithms but also for the user (imagining moving a limb is rather unnatural). Hence, the performance of the system varies considerably between users. Currently, these BCI systems are already being used in medical applications, as shown in the neuro-rehabilitation application of Figure 3.29 (extracted from [1]), where a stroke patients is being helped to regain control of a limb.

This dissertation will further expand the information about this type of BCI systems in Chapter 6, where one of the contribution works of this thesis is presented, being focused on motor imagery classification.

Figure 3.29: Neuro-rehabilitation BCI systems based on ERD/ERS can be used for regaining control of limbs during.

Part III

Contributions

Chapter 4

Contributions to the continuous estimation of the stress level

4.1 Introduction

These days, the society is gradually getting more concerned about feelings, emotions and psychological well-being in general. Accordingly, the number of works focusing on detecting, classifying and assessing emotions and certain mental phenomena has considerably grown in the recent decades. Related to it, this chapter will present the contributions done to the estimation of the stress level by processing physiological signals.

As a starting point, the research group had already worked on the detection of physiological changes to assess sudden physiological drifts towards stress [69]. However, this first work, that was done in collaboration between Aldapa and GICI research teams of the University of the Basque Country (UPV/EHU), only focused on sudden changes towards stress. Thus, the team considered interesting to design a tool that could assess stress in a continuous manner using soft-computing techniques. The work that is going to be presented in this chapter was done under the supervision of Dr. Irigoyen from the GICI as it corresponds to the initial phase of this thesis work. It is as a consequence of the collaboration between GICI and Aldapa that the submitter of this thesis ended up belonging to Aldapa research team.

Concerning the structure of the chapter, Section 4.2 will give the reader a review of the research context and the work in the literature related to this contribution. Then, Section 4.3 will present the experimental protocol that was used to collect the physiological signals used to assess stress. After that, Section 4.4 will explain the signal processing techniques used for extracting the features used as inputs of the fuzzy logic system and, consecutively, Section 4.5

will explain the fuzzy-logic system for assessing stress. Finally, the results of the work will be presented in Section 4.6 and a brief summary will be given in Section 4.7.

4.2 Context and related work

Affective computing is an interdisciplinary field that merges techniques and studies from computer science, psychology and cognitive science. Although psychologists have worked on similar studies, the modern conception of affective computing derived from [70]. The span covered by affective computing is considerably wide, being the part related to emotion and feeling measuring the one of interest for this chapter.

The study of emotions and feelings is not a new a branch of research. Nevertheless, it keeps on drawing the attention of researchers as the human mind is something about which there are still many thing to discover. Accordingly, there are many works that study the influence of the emotions in daily life. Some research focus on the study of human behaviour [71]. Some others pay more attention on how emotions can be elicited [72]. Nevertheless, there is also a branch that, deriving from the postulation of the "fight or flight" theory [73], has focused its attention on the relationships between physiological changes and the feeling of psychological stress [74, 75]. It is to that last branch to which the contribution presented in this chapter belongs to.

As already explained in Chapter 2, "fight or flight" situations produce an activation of the Sympathetic Nervous System (SNS) that makes to the body react: the cardiac and respiration paces accelerate to provide more oxygen to muscles, sweat secretion increases, eye pupils dilate [76, 77]... Consequently, as these changes can be measured by sensors, several research teams have focused attention on measuring these physiological variations looking to analyse their relationship to emotions and/or feelings, such as stress. Whereas some use intrusive sensory such as cameras for eye diameter or facial expressions [78], others try to use less intrusive (or even non-intrusive) devices integrated in clothes or accessories [79, 80].

Although it seems obvious that physical, physiological and psychological well-being are related, how this linkage works is not totally clear [81]. Subsequently, many researchers have used different types of soft computing techniques such as machine learning or fuzzy logic to help to relate physiological changes and stress [82–85]. Following this line, the work explained in this chapter will present a enhanced fuzzy logic system to continuously assess stress that is based on the one presented in [86], which used the heart-rate (HR) and electro-dermal activity (EDA) as inputs signals. The enhancements of the proposed system mostly derive from improving the preprocessing of physiological signals used by [86], such as the ECG R peak detection [87–90] or decomposing the EDA, and from the inclusion a new input signals to the system, the respiration (RESP).

4.3 Experimental protocol

Creating an experimental protocol in which humans will be taking part requires to take several aspects into account from legal, ethical and technical perspectives. Concerning the legal and ethical aspects, it is of major importance to preserve all the privacy rights of the participants. Therefore, saving data in a way that the identity of the participants remains anonymous has become a common practice in human-related research [91]. Besides, all participants have to be informed about the experimental procedure and then sign a consent form in which they declare that they have accepted to take part in the experiment voluntarily. 166 people participated in the experiment, all coming from the University of the Basque Country UPV/EHU (125 male and 41 female). The population was aged between 19 and 45 years old, with an average age of 22.8 years and a standard deviation of 3.1 years.

Apart from the ethical and legal issues, the technical aspects that a research team has to bear in mind concern other types of matters, such as choosing the proper position of the sensors, designing the phases of the experiment in the correct order or choosing the best tasks to produce the desired reactions in the participant. There is a lot of literature covering experimental designs for producing certain psychological or emotional states, such as [92] and [93].

The protocol presented in this section was already used by the team in [69] and it is intended to produce stress by asking participants to mount a 3D wooden puzzle in 10 minutes. Also, participants were shown a 2 minute relaxing video both before and after the puzzle assembling phase. The intention of showing this video was to get participants to a basal state so that their physiological variables would remain in normal ranges and could be used as control information. Hereinafter, the following short names will be used those three phases: RS (Relaxing State) will be used for the relaxing video displaying phase and the puzzle assembling part will be named SS (Stressful State).

Throughout the whole experiment the participants' EDA, ECG and RESP signals were collected using Biopac MP36® sampling at 1000Hz. As completing the puzzle requires hand dexterity, the EDA sensors were placed on the middle and index fingers of the non-dominant hand of the participants. This way, the EDA sensor would not affect the performance of the participants in a great manner while at the same time created a little disturbance that could be useful for producing a bit of stress. On the other hand, ECG electrodes were placed on the chest of the participants and the RESP was collected with a chest-band. Figure 4.1 shows both a diagram of the positioning of the sensors (left side) and an example of the collected signals giving marks for the beginning and ending of each phase of the experiment (right side).

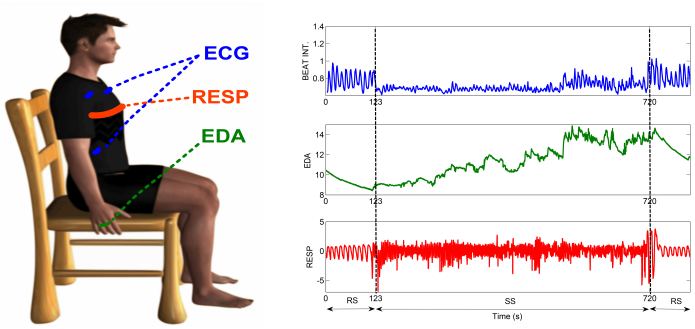

Figure 4.1: Sensor positioning scheme and collected data time series.

Another technical problem that needs to be overcome is the apparition of artefacts in the signals. As assembling the 3D puzzle requires the participant to move, the collected signals likely have artefacts as sensors might eventually lose contact with the skin. Also, the electrodes' gel may get dry subsequently reducing the sensor-skin conductivity. As a result of this, sometimes the amplitude of the ECGs may get reduced and, consequently, the detection of the R peaks becomes more complicated due to the signal to noise ratio reduction. Figure 4.2 presents some examples of these types of perturbations in the signals collected during the experiments. The processing techniques proposed in Section 4.4 were designed to help overcome this type of problems.

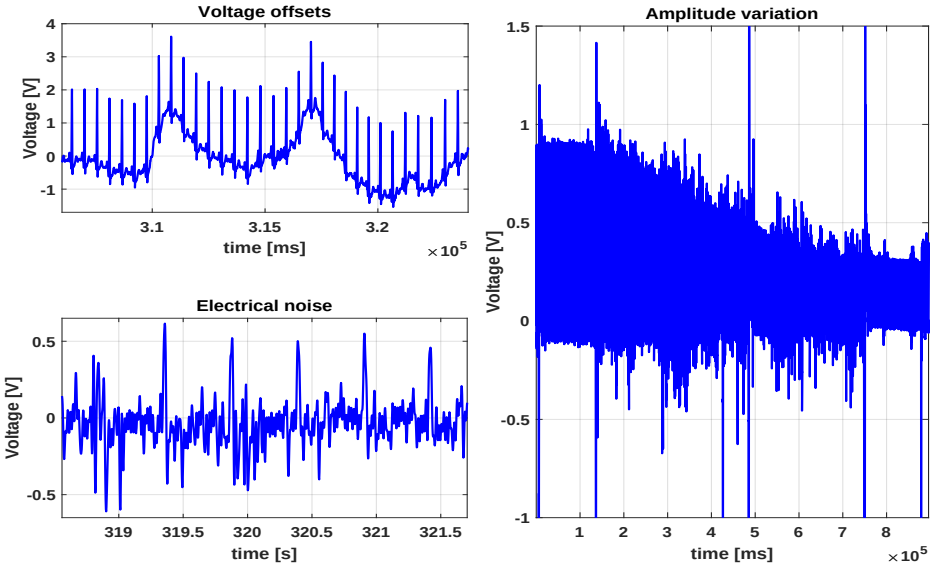

Figure 4.2: Different noises and artefacts in the ECG signal.

4.4 Processing physiological signals

As already mentioned, the proposed continuous stress level estimation method uses features extracted from the ECG, EDA and RESP signals. Therefore, this section will present the processing techniques that have been used to extract the desired features from those signals.

First, Subsection 4.4.1 will present the processing of the ECG signal, presenting the contribution of an enhanced R peak detection method that deals with artefacts as the ones of Figure 4.2. Later, the treatment of the RESP signal will be presented in Subsection 4.4.2, where the combination of frequential and statistical analysis techniques is proposed. Finally, the extraction of the features related to EDA will be explained in 4.4.3.

4.4.1 Processing ECGs: enhanced R peak detection

Median Filtering

Getting a signal whose baseline voltage is moving is relatively common when collecting ECGs. There are different approaches for coping with this problem such as high-pass filtering. However, designing powerful frequency filters may require higher computational loads or can cause phases shifts in signal. That is why, being a very easy non-linear filter to apply, [87] proposed to use median filtering for removing baseline offsets.

After doing an empirical analysis of the signals collected in the experiment, it was decided to use a 100ms median filter. This filter size has a proper size to remove baseline drifts produced by body movements as these do not normally last longer than that. An example of this can be seen in Figure 4.3, where the original signal and the filtered signal have been plotted one on top of the other.

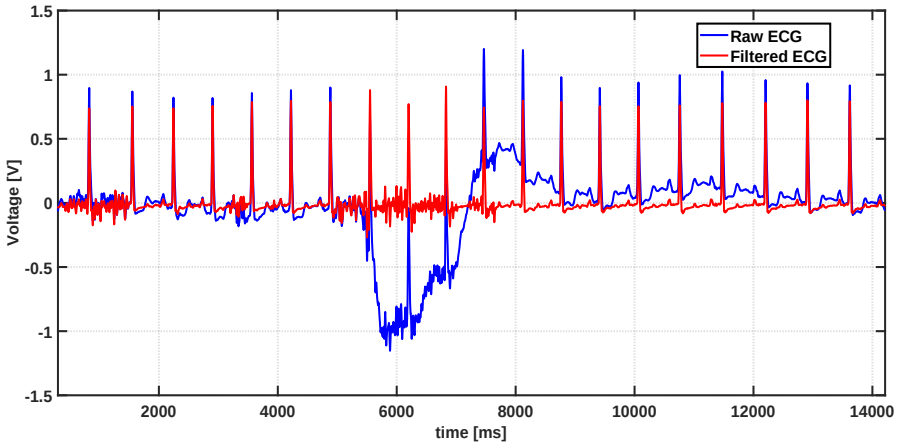

Figure 4.3: Baseline offsets are successfully removed thanks to the 100ms median filter.

Wavelet filtering

Apart from baseline offsets, it is not uncommon to pick up electrical noise while collecting biosignals. For instance, commercial devices usually offer the possibility to engage a notch filter at 50Hz to remove any noise coming from the power grid. However, if the contact between electrodes and the skin is not very good the noise collected by the sensors may not be concentrated in a specific frequency band. In those cases when traditional filtering fails, discrete wavelet analysis appears to be a good candidate to remove noise [94]. Figure 4.4 shows an explanatory diagram of how wavelet-based decomposition and reconstruction is done.

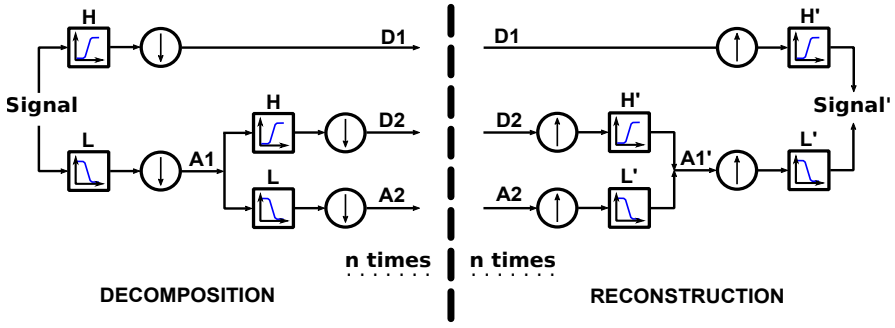

Figure 4.4: Wavelet decomposition and reconstruction scheme.

Wavelet decomposition is shown in the left part of Figure 4.4 and as it can be seen it consists on dividing the signal into two frequency bands one of them corresponding to the high-frequency information and the other to the low-frequency information using correspondingly H and L (H for high-frequencies, L for Low-frequencies) decomposition filters. After that, the resulting signals have to be subsampled to half of their original sampling frequency. It is from this subsampling stage where A and D coefficients are obtained: whereas A coefficients carry the signal information in the [0,Fs/2] range and D coefficients get the information bounded in [Fs/2,Fs]. Wavelet decomposition can be repeated as many times as desired, enabling to get all the information of the original signal in compressed size data chunks.

On the other hand, discrete wavelet reconstruction is just as simple as reverting the decomposition by going in the opposite direction. In this sense, instead of getting subsampled, A and D coefficients have to be upsampled by zero-padding the newly created samples. Afterwards, the resulting signals have to go through the reconstruction H' and L' filters, which deal with those zero-padded new samples. Finally, the A coefficient of the next decomposition level is obtained by adding the outputs of H' and L'.

Thus, discrete wavelet decomposition is a very powerful signal processing tool as it permits researchers to only focus on the relevant information band.

This can be done very easily by just making all the values of a given A or D coefficient arrays equal to zero. If this is done, then when attempting to reconstruct the signal all the information contained in those coefficients will be ignored, giving, as a result, a signal much cleaner of noise. Also, thanks to this property, wavelets have been widely used as a compression solution for storing large data.

However, reached this point, what it is still unclear is how to implement H, L, H' and L' filters. The answer to this issue is not trivial as the implementation of these filters depend on the wavelet that has been chosen for the analysis. There exist several wavelet families and it is not uncommon that different wavelets can handle the work properly. In this sense, if a signal gets decomposed and then it is possible to perfectly reconstruct it using a wavelet, then that wavelet can be considered to be appropriate for the task. Also, another good rule of thumb is to choose a wavelet that has a visual similarity to the relevant information.

In this sense, Coiflet wavelets [95] look relatively similar to QRS complexes and so, after checking that the reconstruction can be completed without any loss, it was selected for noise filtering problem of this subsection. Hence, the baseline offset free ECGs were decomposed using Coiflet wavelets. Later, it was decided to ignore the D coefficients for the reconstruction getting as a result and approximation of the ECG with less noise. It is important to bear in mind that this process produces some loss of information. Nevertheless, the morphology of the ECG is maintained relatively good and the signal to noise ratio is better than in the original signal. An example of this is shown in Figure 4.5, where the signal was decomposed six times and reconstructed only using A coefficients.

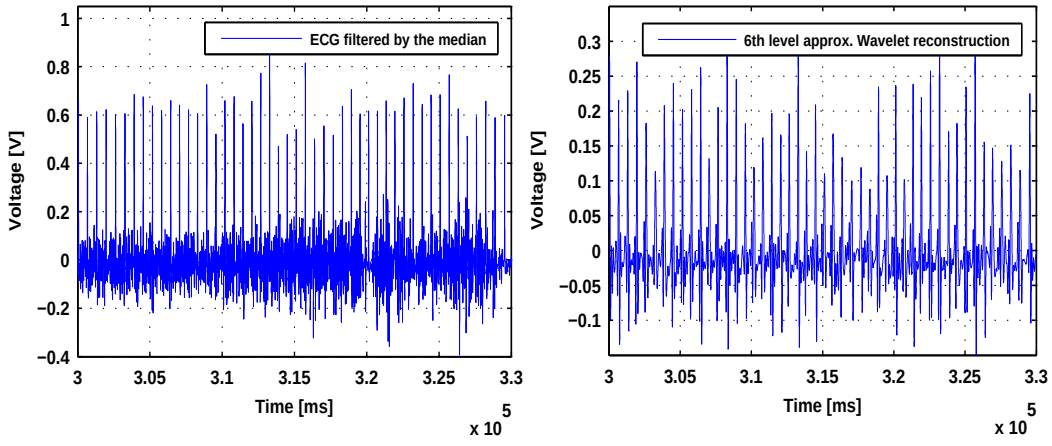

Figure 4.5: Noise filtering by the 6th wavelet approximation.

Peak detection

Once the noise has been removed, the next step to be taken to calculate the HR is to detect the R peaks. To do so, it is proposed to remove all the negative values from the ECG wavelet approximation (or only the negative part in those abnormal cases that R peaks are negative).

After that, it is only necessary to sweep the signal to detect the most prominent peaks, i.e., the R peaks. As the ECG is not a highly dynamic signal, this work proposes average the previous three RR intervals and to use that average value to estimate the position of the next R peak. Then, detecting the peak is as easy as sweeping the signal backwards and forward a 20% of the averaged RR interval. Once this is done it is convenient to check that the process was completed correctly by plotting the detected R peak positions on top of both the wavelet approximation and original ECG signals to check that the process was completed correctly (see Figure 4.6).

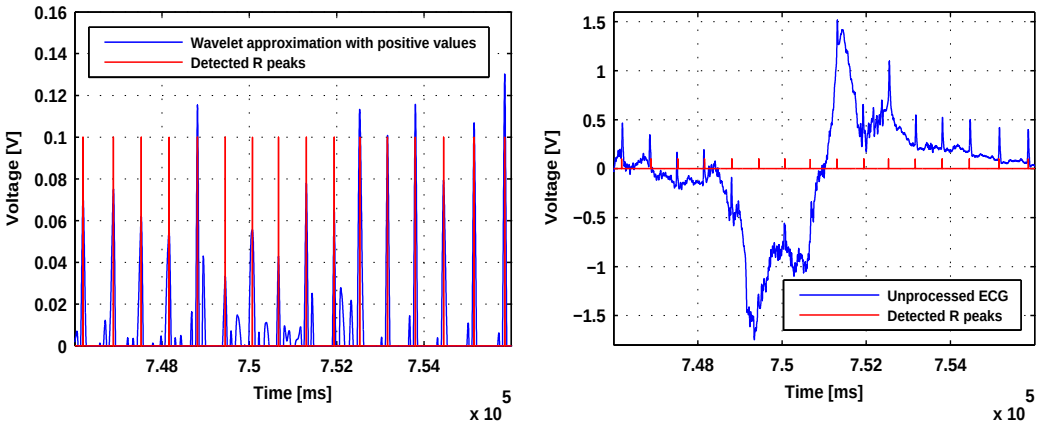

Figure 4.6: The detected R peak positions match the R peaks of both the wavelet approximation and original ECG signals.

Finally, if the R peak detection was done correctly, it is very easy to calculate the HR by just applying equation 3.1. This HR signal will be more robust than those calculated with more traditional methods. For example, for the signal of Figure 4.2 the proposed algorithm gives a more consistent HR calculation compared if it is compared against the one given by the Biopac MP36® device used to collect the biosignals during the experiment (see Figure 4.7).

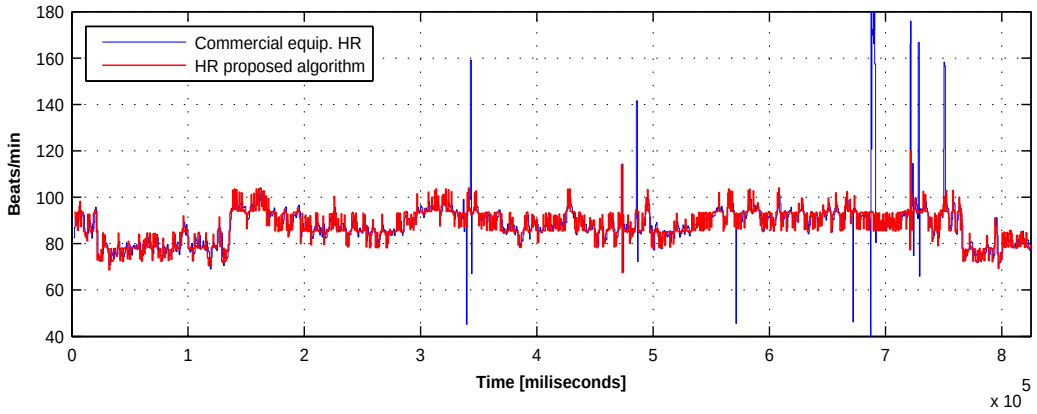

Figure 4.7: Whereas the commercial equipment can be highly affected by artefacts, the proposed algorithm gives a more robust HR calculation.

4.4.2 Processing RESP signals

As stated in Subsection 3.2.3, it can be expected that the RESP of a relaxed person is more harmonic compared to when feeling stressed. When the SNS gets active it forces the lungs to move faster so that the transfer of oxygen to blood can meet the required quantities by the organism. Hence, it is possible to obtain information for assessing the level of stress of a person by applying the proper processing techniques to the RESP signal. This work proposes to combine both frequential analysis techniques with statistical calculations so that this information can be used as an input feature for the fuzzy logic system presented later in Section 4.5.

Frequential analysis of RESP signals

Normally the Fourier transforms are used to check the frequential spectrum of a signal. This way, it is possible to check how harmonic a signal is. However, getting a big resolution in frequency requires many samples to be taken and that can be relatively computationally expensive. Moreover, traditional Fourier transforms provide information in the [0,Fs/2]Hz frequency range and, in the case of physiological signals, normally Fs/2 is a frequency from which little information can be extracted.

Thus, this work proposes to analyse the frequential concentration of the RESP, i.e., how harmonic the signal is, by correlating it with purely sinusoidal waves of different frequencies. By doing this it is possible to focus all the attention only in the interesting frequency band. Hence, for each subject, it is only necessary to calculate the spectrum of the signal once to check the boundaries of the frequency band in which most of the physiological phenomena are located. An example of it can be seen in Figure 4.8.

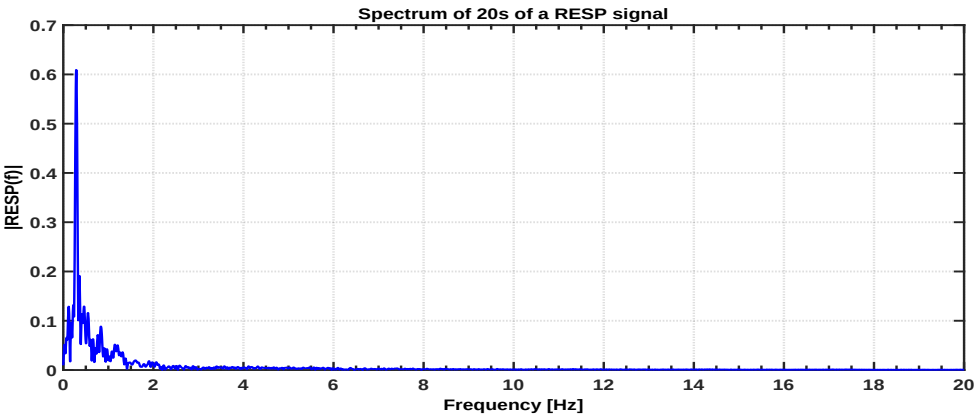

Figure 4.8: Frequency spectrum of a RESP signal.

After checking the spectra of the RESP signals of different participants, it could be concluded that the [0-0.5]Hz frequency range contains the information of both relaxed and stressed RESP signals. That is why it was decided to get the correlation between RESP signals and purely sinusoidal waves going from 0.01Hz to 0.5Hz using 0.01Hz as the step size. These correlations were calculated using different window sizes for seeing which window size would be best for extracting the information concerning stress. The sizes of the windows that were analysed were 20s, 40s and 60s, all of them using a sliding window step of 10s. The result of these correlations can be seen in Figure 4.9, where an example RESP signal has been plotted on the top and the correlations for the different window sizes have been given below it using a colour map.

Paying attention to Figure 4.9 it can be concluded that during the puzzle-solving part of the experiment the RESP signal loses frequential concentration and becomes less harmonic (more green spots appear in the graph while the amount of purple areas is considerably reduced). On the other side, in the parts intended to be relaxing, that is, in the beginning and end of the experiment, it can be seen how the RESP is highly concentrated in a given frequency (purple colour) and that all the other frequencies show low correlation values (red areas).

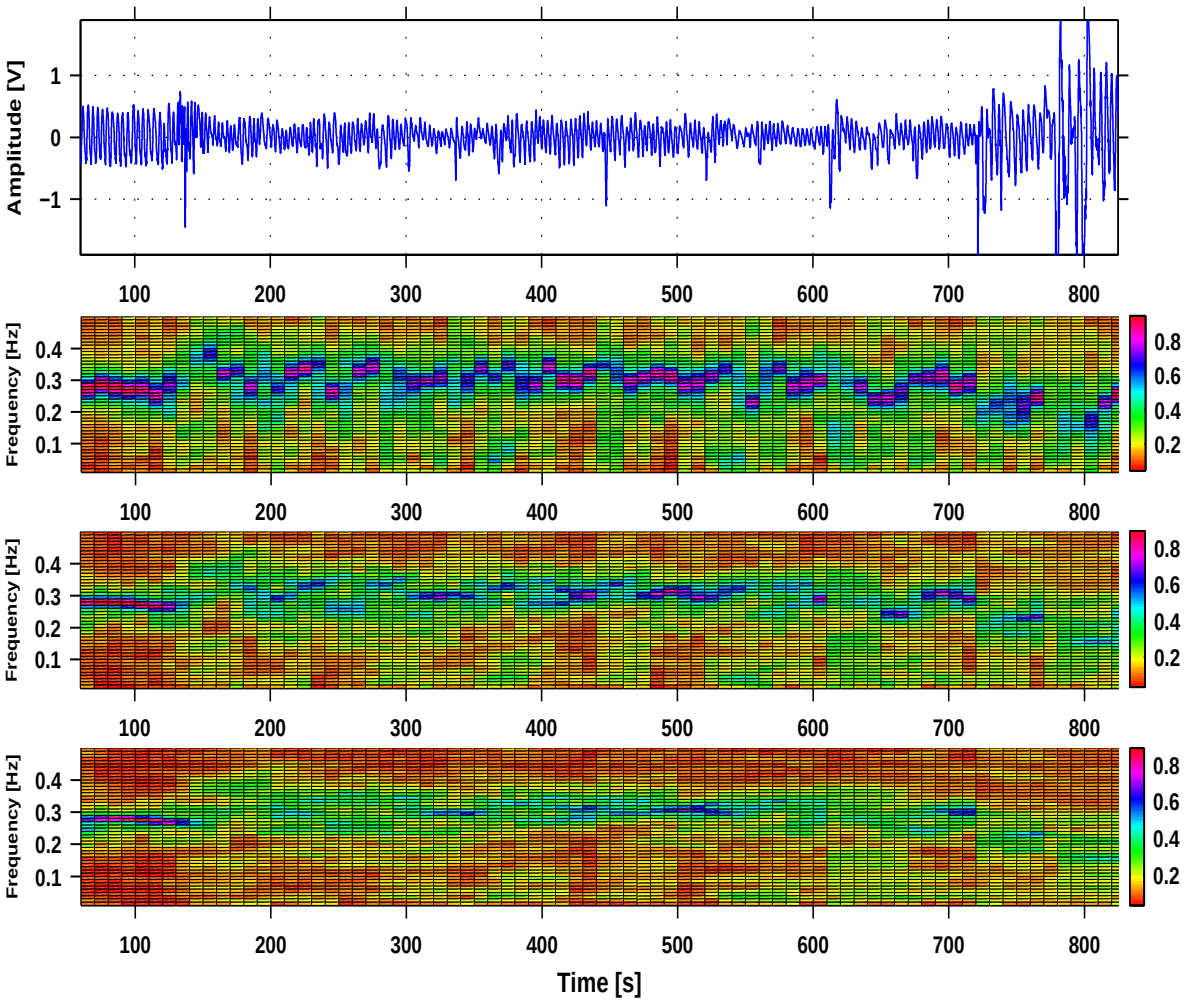

Figure 4.9: Correlation between the RESP signal and purely sinusoidal waves using different window sizes: first 20s, second 40s and then 60s.

Statistical analysis and signal softening

As seen in Figure 4.9, the frequential concentration of the RESP varies when going form relaxing to stressful situations. The work presented in this chapter proposed to use the standard deviation of the frequency correlation values as an input feature for continuous stress assessment fuzzy logic system.

Following this approach, the calculation of the standard deviation was proposed to be done following the sliding windows strategy. When this was done, the standard deviation of the correlation values was higher when the participant

was feeling relaxed. This makes sense because when feeling relaxed the correlation values for all the checked frequencies are very small, with the only exception of the main frequency which is highly correlated. Thus, the average value will be relatively distanced from most of those points. On the other hand, during stressful situations the correlation values are more evenly distributed and, so, all of them are closer to the mean value. Subsequently, the standard deviation is smaller compared to that of the relaxing situations. Bearing in mind that it sounds a bit tricky, Figure 4.10 depicts graphically one example of the two cases for the same person.

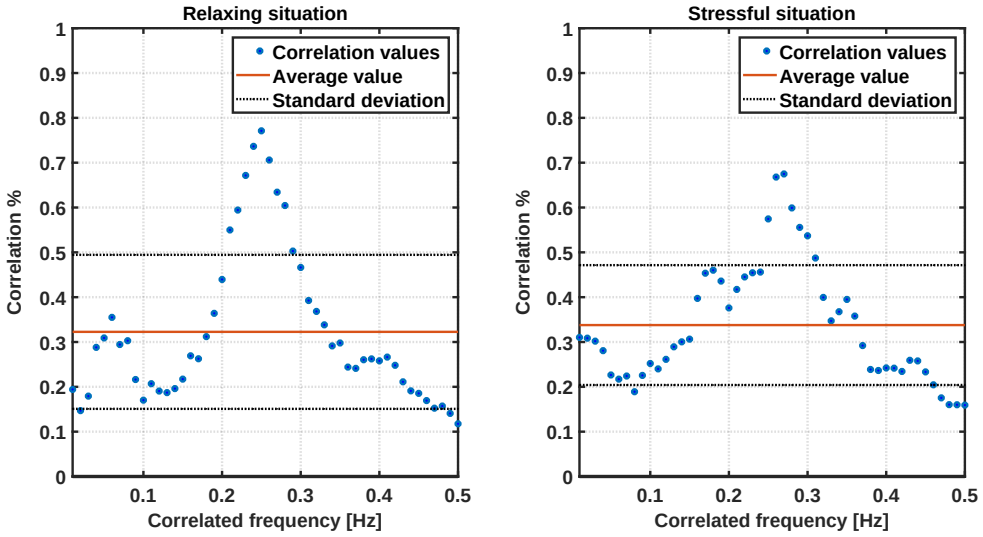

Figure 4.10: Correlation values between the RESP signal and purely sinusoidal waves and the statistic parameters.

Having understood this concept, it is also possible to plot the evolution of the standard deviation of the correlation values of the RESP with the pure sine waves. As shown in Figure 4.11, it can be seen how the standard deviation gets higher values in the relaxing parts of the experiment. Also, it can be seen that the standard deviation is smaller in the middle of the experiment (the stressful part).

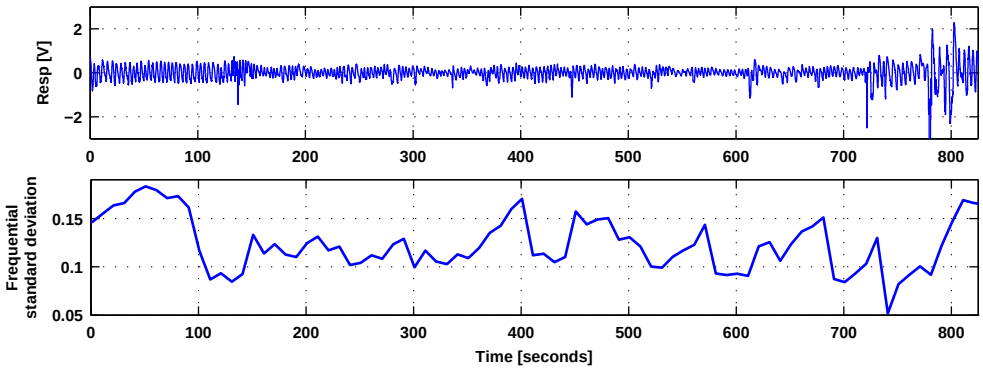

Figure 4.11: Top: A participant's RESP signal. Botton: the standard deviation of the correlation values.

Despite the difference between relaxing and stressful parts seems clear to the eye, this signal would lead the fuzzy logic system to some false detection if it was used as an input as there are some spots during the stressful part that have values relatively high (e.g., around 400s in Figure 4.11). Hence, it is necessary to soften this signal so that the differences between stress and relax are increased. This work proposes to do it by multiplying the signal by the RMS value of the RR peak time interval signal, whose formula was already presented in Equation 3.2. The result of this multiplication gives a signal in which the differences between relaxing and stressful situations have been increased (see Figure 4.12), and, so, it is possible to use it as an input to the fuzzy logic system.

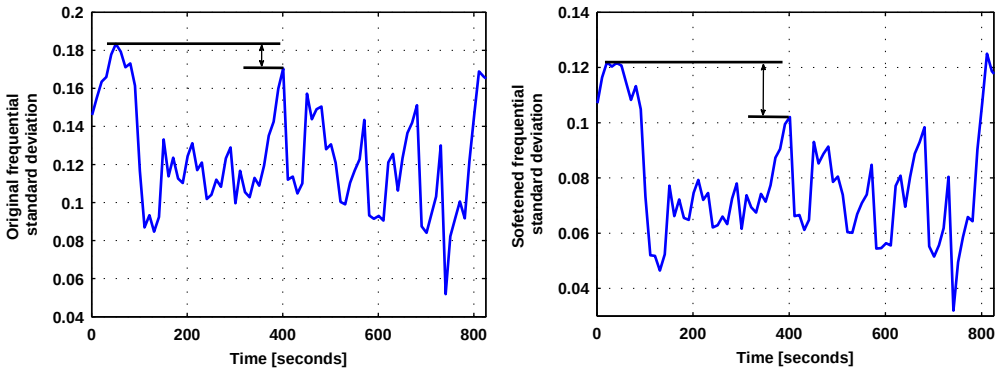

Figure 4.12: After the multiplication, the resulting softened signal shows bigger differences between the peak values of the relaxing and stressful stages.

4.4.3 Processing EDA signals

Concerning the EDA, its processing is simpler. It was decomposed into two parameters: the average value and the variations within a 20s analysis window. This way the average value is given less importance in favour of the variations of the signal within the window. This is a common approach in the literature because being representative of the sweat on the skin, the EDA is a cumulative signal, and so, it is difficult for it to reduce its baseline. On the other hand, the variations do not depend on the average quantity and it can be considered to be more representative of SNS activations compared to the average EDA value.

4.5 Stress detection fuzzy logic system

This section presents the fuzzy logic system that was designed to continuously assess the stress level. To do so, it uses the previously presented four input features: the enhanced HR signal, the softened standard deviation of the correlations between the RESP and pure sine waves and the average and variation values of the EDA. The design of the membership functions (MF) followed a similar approach to the template extraction proposal of [86], but adding an extra function for intermediate stress levels. The work presented in this section was done using the Matlab® fuzzy logic toolbox.

First, Subsection 4.5.1 will present the membership functions and the reason to do decompose the EDA signal and then, the strategy followed for the rule system will be explained in Subsection 4.5.2.

4.5.1 Design of the fuzzy logic membership functions

Initially, when the team started to design the fuzzy logic system of [96], the design of the input MFs was based on the template extraction approach from [86]. This approach proposed to take physiological signals from the RS and the SS phases separately. Then, for each signal and phase of the experiment, a Gaussian MF has to be defined using the average and standard deviation values of the signal in each period. Nevertheless, in some situations in the HR signal suffers from great variability and it produces the standard deviation of the HR to be very high. This phenomenon can be observed in the HR signal of Figure 4.13, where the HR of the participant varied a lot during the initial part of the experiment.

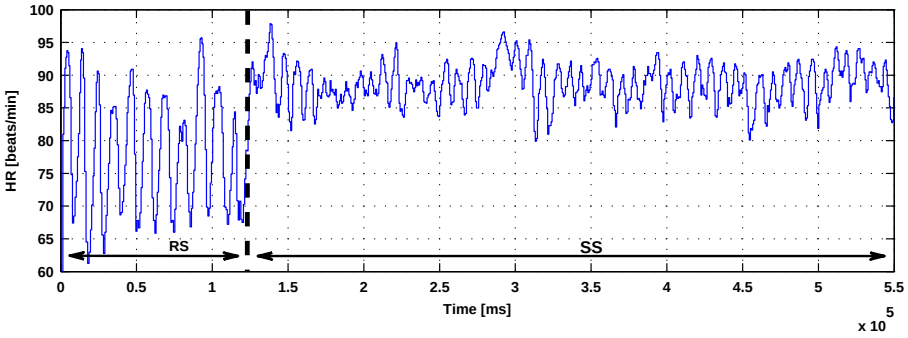

Figure 4.13: The variations of the HR were high during the RS phase of the experiment.

As a result of this phenomenon, the input MFs might get overlapped. This is a problem, as it leaves many situations of uncertainty in which an input HR value could belong both to the RS and SS situations in a very similar manner. To handle these situations, this work proposes to use an intermediate MF for the transitions between relaxation and stress. The term of the intermediate function was "medium stress" (MS). Also, it was proposed to use trapezoidal MF shapes for RS and SS and to only use a Gaussian shape for MS. As it can be seen in Figure 4.14, this approach prevents the overlapping problem from happening (right part of the figure), whereas using only Gaussian shapes might result in MF overlapping. Moreover, the proposed three MF approach deals with the handling extremely low and high values that would otherwise be taken as an uncertainty.

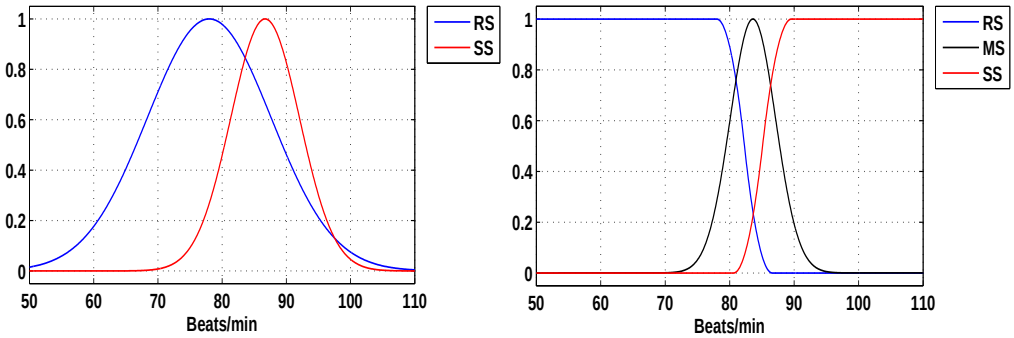

Figure 4.14: Overlapping of the HR MFs can be avoided using a third MF for transitions between relaxation and stress.

For the sake of coherence, the same strategy has been followed for the rest of the input MFs, defining RS, MS and SS as the MFs for all the inputs. On the one hand, all RS and SS have been defined as trapezoidal functions. On

Table 4.1: Definition of the membership functions.

Variable	Definition	States	Shape	Shape edges
Input:		RS	Trapezoidal	Variable
Heart-Rate	Variable	MS	Gaussian	Variable
		SS	Trapezoidal	Variable
Input:		RS	Trapezoidal	Variable
Avg.EDA	Variable	MS	Gaussian	Variable
		SS	Trapezoidal	Variable
Input:		RS	Trapezoidal	[-2,-2,-0.75,0]
ΔEDA	[-2.2]	MS	Triangular	[-0.5,0,0.5]
		SS	Trapezoidal	[0,0.75,2,2]
Input:		RS	Trapezoidal	Variable
RMS_{RR}	Variable	MS	Gaussian	Variable
$\cdot CORR_{RESP}$		SS	Trapezoidal	Variable
Output:		RS	Trapezoidal	[0,0,0.275,0.475]
Stress level	[0,1]	MS	Triangular	[0.25,0.5,0.75]
		SS	Trapezoidal	[0.525,0.725,1,1]

the other hand, not all variables have the same shape for MS: whereas Gaussian shapes were used for the HR and the average EDA, triangular functions were defined for the variations of the EDA and the multiplication of the RMS of the RR intervals and the correlations of the RESP. With regards to the output of the system, these three MFs had been defined since the beginning. The output was designed to be bounded in the [0,1] crisp vale range for having a normalised stress estimation. A summary of the definition of all MFs is presented in Table 4.1. Some values of the table can be predefined and standard for all subjects, but it cannot be done for the others because they depend on the voltage range of the collected physiological signal.

4.5.2 The rule system

The design of the rule system was done using "IF... AND IF... THEN..." rules using the "max" aggregation method. Hence, the input variables were related to each other by pairs. Also, as mentioned in Subsection 3.3.3, the defuzzification was done with the COG method. However, the relationships defined with the rules varied along with the development of the physiological signal processing techniques presented before in this Subsection 4.4.2.

In the initial stages of the rule design, two MFs had been defined for input variables (RS and SS) and three for the stress estimation output (RS, MS and SS). In this situation, the rules were defined so that the MS output would get active only when the first and the second inputs of the rules were not coincident in class, i.e., when input 1 belonged to the RS state and input 2 belonged to SS

and vice-versa. Otherwise, when the membership of both inputs was coincident the conclusion of the rule would be to activate the output of coincident type. This approach is represented in a summarised manner in Table 4.2.

Table 4.2: The initial rule relationships.

State of input 1	State of input 2	Conclusion
SS	SS	SS
SS	RS	MS
RS	SS	MS
RS	RS	RS

Later, after doing some tests, it was decided to modify the strategy as the system's output suffered very dynamic shifts very easily. Therefore, it was decided to add the MS state as the third MF of the input variables and, this way, take advantage of those intermediate states to get smoother transitions for the stress estimation. Then, the rules system had to be modified as a consequence of adding a new MF to all inputs. In this sense, the rules were modified get a much simpler system: the output would be RS if inputs 1 and 2 were RS and the same criterion would be followed for the MS and SS cases. This new approach is reflected in Table 4.3.

Table 4.3: Input variable relationships.

State of input 1	State of input 2	Conclusion
SS	SS	SS
MS	MS	MS
RS	RS	RS

To finish with the rule design, the set of rules of Table 4.3 were given different weights depending on the physiological variables that were feeding each rules. This was done because not all the inputs to the system have the same importance for assessing stress. For example, the relationship between the HR and the variation of the EDA is more reliable than the relationship of the HR with the average value of the EDA (due to the accumulative nature of the signal). All the rules of the system have been represented in Table 4.4, where the weight of each combination has also been indicated. Besides, the relationship between the two inputs related to the EDA has not been considered as it would be as entering the whole EDA as an input.

4.6 Experimental results

The last step of the design of the stress assessment system was to take it a testing stage. The intention was to do a comparative analysis between the proposal of [86], the system that was obtained from the firsts stages of the

Table 4.4: Proposed rule system.

Input 1: variable-state	Input 2: variable-state	Conclusion	Weight
HR-RS	ΔEDA-RS	RS	1
HR-MS	ΔEDA-MS	MS	1
HR-SS	ΔEDA-SS	SS	1
HR-RS	RMS_{RR}·$CORR_{RESP}$-RS	RS	0.75
HR-MS	RMS_{RR}·$CORR_{RESP}$-RS	MS	0.75
HR-SS	RMS_{RR}·$CORR_{RESP}$-RS	SS	0.75
ΔEDA-RS	RMS_{RR}·$CORR_{RESP}$-RS	RS	0.6
ΔEDA-MS	RMS_{RR}·$CORR_{RESP}$-RS	MS	0.6
ΔEDA-SS	RMS_{RR}·$CORR_{RESP}$-RS	SS	0.6
HR-RS	Avg.EDA-RS	RS	0.5
HR-MS	Avg.EDA-MS	MS	0.5
HR-SS	Avg.EDA-SS	SS	0.5
Avg.EDA-RS	RMS_{RR}·$CORR_{RESP}$-RS	RS	0.3
Avg.EDA-MS	RMS_{RR}·$CORR_{RESP}$-RS	MS	0.3
Avg.EDA-SS	RMS_{RR}·$CORR_{RESP}$-RS	SS	0.3

development [96] and the final one [97]. To do the comparison, the three systems mentioned above were fed with the same input signals (the ones collected in the experiment). For making the comparison easier, along with stress estimation of the three systems (bottom graph, all the input signals have been depicted in Figure 4.15 (the four first charts) in the following descending order: HR, average EDA value, variations of the EDA and softened standard deviation of the correlations between the RESP and pure sine waves. As stress responses do not have very fast dynamics, the values of the input signals were fed into the systems using a 20s window size with a sliding step of 10s.

Around the beginning and ending of the puzzle-solving phase (t=120s and t=720s), Figure 4.15 shows how the strategy of adding a new MF for the MS state enables the proposed system to react more softly compared to the system proposed by [86] and to the system of the initial stages of this work [96]. Furthermore, from t=250s to t=300s, the softened standard deviation of the correlations of the RESP signal act as a brake for the system not to overestimate the level of stress when the EDA is increasing.

Therefore, it can be said that the proposed new system is more stable across the whole span of the stress levels, being less prompt to react to sudden drifts in just one of the input signals. Also, it can be seen that the overall stress estimation levels given by the proposal are relatively similar to the ones given by the two other systems. Anyway, it is difficult to say how accurately these values represent reality as the feeling of stress is something subjective and the only way to quantify it is with the surveys fulfilled by the participants at the end of the experiment (the "Self-Assessment Manikin" proposed by Lang in [98]).

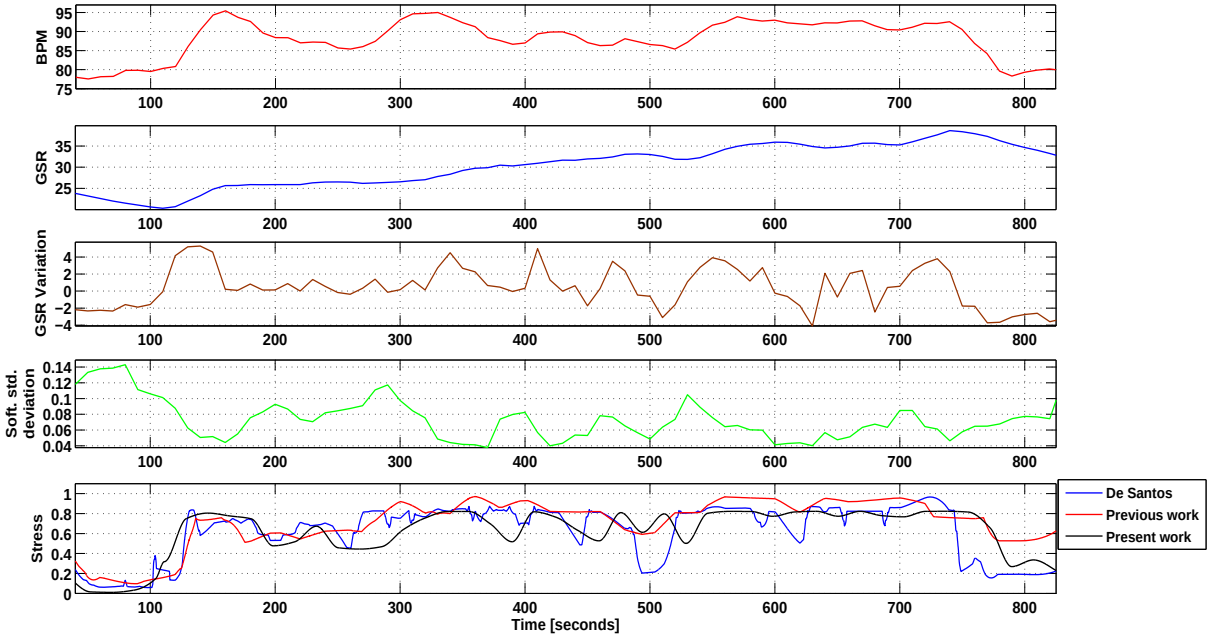

Figure 4.15: HR, EDA, EDA variations and softened RESP Standard Deviation inputs and estimated stress level outputs for the three methods.

The main problem in this sense is that the surveys do not provide a continuous indicator of the stress felt but rather an overall impression of how subjects felt during the experiment.

4.7 Summary

There is an increasing interest regarding the processing of physiological signals in the search of the well-being of the population. The automatic detection of certain physiological events can be a useful tool to detect certain psychological diseases such as stress and to enable a quick treatment. Nevertheless, physiological processing techniques have to deal with highly varying time series which likely have many artefacts.

The contribution of this chapter presented processing techniques that make it easier to cope some of the problems that arise when collecting physiological signals (the improved R peak detection) and proposed new features that can be useful for stress detection (the softened standard deviation of the correlations of the RESP). This way it has remarked the importance of having robust processing techniques for not getting false trips in a stress detection system.

Furthermore, it presented a modification of a fuzzy logic system to improve the detection of the psychological stress level by using four input features to

the system: HR, average value of the EDA, the variations of the EDA and the softened standard deviation of the correlations of the RESP. Besides, the way of defining the relationships between input variables has been made easier by adding an intermediate stress level to the system input membership functions. Finally, a comparison of the proposed system against two different systems showed how the proposal achieved softer transitions between different stress levels.

Limitations and future lines

Although the presented contributions present improvements to different domains related to stress detection, some weak points need to be mentioned in this regard. First, this work has not given a method to initialise the R peak detection sweeping algorithm. In this sense, the initialisation was done manually, which is one of the major flaws of the method. However, this weak point opened a new research line, leading the team to improve the proposed techniques. As a consequence, this type of issues have already been solved in [99].

Continuing with the limitations, there is a weak point that is relatively common in the world of the detection of events related to the psychological domain: labelling and validation of labels. When researching in these fields, it is very complex to label events and to quantify the strength in which psychological phenomena are taking place. The information related to these aspects can only be collected employing questionnaires and interviews, which normally give an overall indicator of how subjects had felt in the whole experiment. Anyway, these interviews hardly give information about eventual phenomena and, despite researchers can take labels during the experiments, these labels are difficult to validate. Undoubtedly, video recording the performance of the participants during the experiments can help to come over these limitations but it adds extra difficulty to the preparation of the experiments as the ethical and legal limitations become quite more restrictive.

Finally, concerning the proposed fuzzy logic system, the major weakness is that it requires to have previous knowledge about the normal user's physiological parameter. As the membership functions are based on template extraction, it is impossible to fine-tune the system if the ranges within which the physiological variables of the user are taking place are unknown. Some research teams as [86] propose the use of neural networks for creating adaptive membership functions to overcome this problem. Although the proposal sounds feasible, difficulties for getting a good labelling of data make this proposal relatively complicated.

Chapter 5

Contributions to detection of responses towards relaxation

5.1 Introduction

In the recent years, there has been a change in both medicine and psychology towards considering mental and physical well-being as a key tool for preventing diseases rather than focusing on treating their symptoms [100–103]. Following this tendency, psychologists and clinicians have started to use biofeedback techniques to determine the mental state of patients by monitoring their physiological signals. These techniques, for instance, can be used to determine how deeply a patient has gone into relaxation [104–106]. This is an interesting approach as it helps determine which are the best working relaxation techniques for each patient. Consequently, it helps them to prevent psychological disorders such as stress or anxiety, or else, recover from them. However, collecting and interpreting physiological signals is far from easy and requires expertise. Therefore, only professionals can use these techniques. That is why developing techniques for the automatic detection of psycho-physiological phenomena can be especially useful.

In spite of many studies focusing on stress, very few pay attention to the physiological phenomena linked to relaxation (most of them belonging to the field of psychology). In this sense, the team considered interesting to research around the automatic detection of relaxation for two main reasons. First, because the technical literature concerning relaxation is almost non-existent and, second, because it is a research line highly connected to the previous work done by the team around stress. Accordingly, the contributions presented in this chapter will be centred in the design of a strategy to enable the automatic detection of physiological drifts towards relaxation.

The work developed for this contribution covered areas of different disciplines of medicine and psychology, apart from computer sciences. Therefore, as most of the steps of this work required verification from experts of both medical and psychological disciplines, this work was carried out with the collaboration of the Department of Neurology of the Cruces University Hospital [107] and of the Instituto Burmuin centre of psychology [108].

Regarding the structure of the chapter, first, Section 5.2 will review the context from which this study derived and will present relevant literature related to the study of relaxation. Second, Section 5.3 will present the experimental setup that enabled to collect, analyse and prepare the data for a later study using machine learning techniques. Although more signals had been collected, only the EDA was finally used for the detection of changes towards relaxation. Accordingly, this section will also cover the extraction of the two features of the EDA, which contribute making it easier to detect physiological changes towards relaxation. After that, Section 5.4, will present the application of machine learning done for automatically detecting those physiological changes indicative of a change towards relaxation. This section also present the results obtained from applying these techniques, especially focusing on decision trees as they were the selected algorithm for taking the development some steps further. Finally, Section 5.5 will summarise the contributions done and will discuss the limitations of the work as well as the future lines derived from this study.

5.2 Context and related work

The World Health Organization states that health is beyond the absence of diseases or infirmity: health implies physical, psychical and social well-being [3]. Currently, evidence has pointed out the importance of how having good mental health acts as an efficient preventive measure for avoiding some everyday diseases from appearing [109]. In this sense, achieving relaxation can be very useful for preventing and recovering from some common diseases of the field of clinical psychology, such as anxiety disorders or depression [110–112]. Related to that, using relaxation techniques is a simple but effective way to get to that mental well-being [113–115].

This way, relaxing using relaxation techniques is a good practise that can be used in a dairy manner, even when no psychologically bothersome symptoms are felt. Focusing on relaxation techniques, it is evident that the effects they produce on the organism do not only stop in the psychological domain. Instead, relaxation techniques have also a beneficial impact on the operation of the body [116], [117]. These beneficial effects get reflected in the physiological changes that occur in the organism; this type of reactions were named "Relaxation Responses" (RResps hereinafter) by Benson in [118]. These reactions of the body are due to the parasympathetic nervous system (PNS) getting active. As already mentioned in Section 2.2, the sympathetic nervous system (SNS) and the parasympathetic nervous system (PNS) react complementarily to each other. Whereas the SNS activates while stressed, it is the PNS that operates

when stress stops or when feeling relaxed [5, 74, 119–121].

As said, these activations of the PNS have impact in many physiological variables, such as the reduction of the heart-rate and breathing pace or the reduction of the opening of eye pupils. Nevertheless, unlike most of those variables that are linked to both the SNS and PNS, the activation of the eccrine sweat glands is only regulated by the PNS. This phenomenon postulates the EDA signal as a good indicator for detecting relaxation: eccrine sweating only occurs when the SNS is active, or, in other words, if sweating is not taking place it means that the parasympathetic part of the nervous system is active.

Finally, concerning technical literature, some studies analyse relaxation from a perspective of emotional recognition. Some others study the states of relaxation using biological signals [80], [122]. Apart from those, some studies focus on stress and compare the patterns that take place during stress against those considered to be basal that can be seen when a person is relaxed [78, 123–127]. However, the team had been unable to find any study focusing on the automatic detection of the physiological shifts indicative of RReps taking place and, so, decided that it would be interesting to research on it and fill that gap in the literature.

5.3 Experimental setup and data analysis

As it normally happens in most of the studies implying the analysis of physiological signals, data is collected by means of an experimental stage. In this study, the signals were collected from two experiments, both involving relaxation events. Whereas the first experiment focused on producing both stress and relaxation, the second targeted only relaxation. After collecting the signals from both of them, these data underwent a qualitative and a quantitative analysis so that the features that were more indicative for relaxation could be extracted. This whole process from collecting the data until it was taken into the final classification stage is summarised by the diagram of Figure 5.1.

This section explains all the work done regarding data collection and preparation: Subsection 5.3.1 will present the populations and experimental protocols of the two experiments, as well as how data was partitioned for the later analysis; Subsection 5.3.2 will explain the qualitative analysis and labelling process of the data and Subsection 5.3.3 will explain the quantitative analysis that was done to extract the most relevant features of the EDA.

5.3.1 Data collection and preparation

As already explained, the data used in this study were collected from two different experiments. The main objective of collecting the signals from different experiments was to get information meaningful of the different types of situations in which relaxation occurs. For example, relaxation may take place in the middle of a stressful situation when a person tries not to get overwhelmed. In other cases, relaxation happens under less aggressive environments, as it

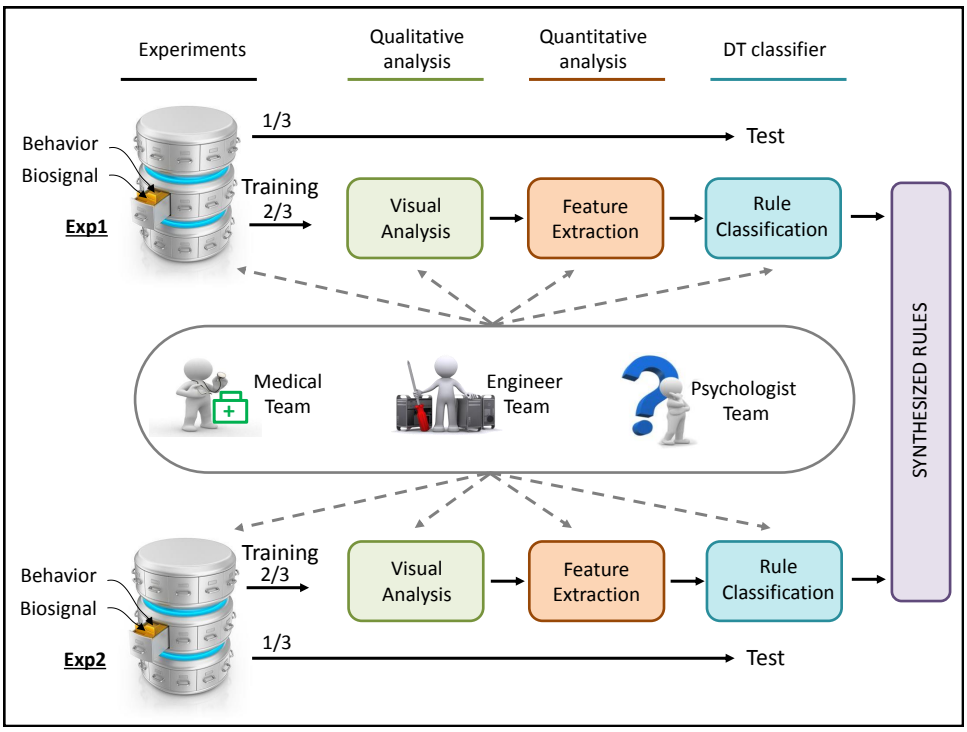

Figure 5.1: The data of the two experiments underwent through different analyses before applying machine learning techniques to detect the RResps.

happens listening to relaxing music laying down in the sofa. Based on these concepts, the first experiment involved RReps inside stressful situations and the second experiment aimed to produce deeper relaxation with the application of relaxation techniques in calm environments.

Experiment 1

The first experiment corresponds to the experiment of the 3D puzzle presented in Section 4.3 and it will be named Exp 1 from now on. As explained in the previous chapter, this experiment aimed to produce stress on the participants. Anyway, during these experiments there were many events of relaxation, as, for example, when participants took deep breaths to get calm. Therefore, Experiment 1 provides data full of information regarding RResps taking place in the middle of stressful situations.

Apart from the physiological signals, the researchers also took record of the feelings and impressions of the participants during the experiment. This was done by asking participants to fill the SAM questionnaire presented in [128]. Furthermore, marks were taken during the experiment, noting down those

moments in which the researchers saw that the participants were taking breaks or long deep breaths to relax. Other important events were also marked, such as the finishing of the puzzle or a piece of the puzzle falling from their hands. Later, participants would be asked about these marked events for corroborating that the notes were correct and subsequently build a behavioural database for labelling the data for the application of machine learning of Section 5.4.

Experiment 2

The second experiment (named Exp 2 from this point) from which data was collected was intended to produce deeper and longer RResps. To do so, in collaboration with Instituto Burmuin, the team designed an experiment in which participants achieved relaxation using different relaxation techniques. The literature covers many ways to induce relaxation: listening to music [129, 130], watching images and videos [131, 132], taking deep and controlled breath [133], meditating [106, 116] and using mind-fullness techniques [105]...

For this experiment, attentional breathing and muscular relaxation techniques were selected. First of all, as done in Exp 1, participants were taken to a basal state for having it as a reference. Then, the experiment was divided into three stages more. In the first stage, participants had to breath in a controlled manner at a rhythm that was indicated in a screen. Then, during the second stage, they would go through an attentional breathing period. To end the experiment, during the third stage they would close their eyes and practise muscular relaxation. The sessions had an average length of 12 minutes and, both before and after the experiment, the psychologist took track of the emotional states of the participants for building the behavioural database of Exp 2.

All the volunteers participated individually in the experiment and all of those sessions took place in a room inside Instituto Burmuin. The total population of the experiment consisted of 18 patients of Instituto Burmuin (4 men and 14 women) participating voluntarily. Therefore, compared to Exp 1, the population of this experiment covers a different age span: they were aged between 32 and 56, being 40.22 years old the average age with a standard of 9.14 years. As done in Exp 1, to fulfil all the legal and ethical requirements, all participants were explained the experiment and asked to fill their respective consent form (CEISH-UPV/EHU, BOPV 32 (M10_2016_189)).

Concerning the collection of the biosignals, for the sake of consistency, the same signals of Exp 1 were in the sessions of Exp 2: the EDA and ECG. Nevertheless, the equipment used to collect those signals was different. In this case, the team collected the signals using a sampling rate of 256Hz with the ProComp Infiniti System hardware and the BioGraph Software - T7500M equipment. Also, as the equipment also provides the possibility to obtain it, the HRV signal was collected during these experiments.

Data preparation

Before starting the analysis of the data and applying machine learning techniques, it is necessary to arrange it so that every data chunk fulfils certain criteria so that how data is presented does not affect any of the later processes. For instance, it is vital to ensure that training and testing data partitions are independent. But sometimes it is necessary to satisfy some other conditions as explained in the following paragraphs.

First of all, the data chunks or instances that are going to be used in the machine learning algorithms must have the same size. This may also affect the data labelling process. In the case of this study, the data collected does not have any label. However, this work was intended for classifying physiological signal chunks to determine whether an RResp is taking place or not. Accordingly, when the team of experts were asked to label the signals, they would have to do it by looking at signal chunks of consistent size throughout the whole duration of the experiments. Thus, following the line of previous work done related to stress, the team decided to divide the signals in 20s instance using a 20s sliding window moving every 5s. However, as the signals had been sampled at 1000Hz in Exp 1 and at 256Hz in Exp, the team had to subsample all of them down to 1Hz before doing the 20s windowing. The labelling process and window selection will be further explained in the following Subsection 5.3.2.

Apart from data chunk size, when machine learning algorithms are involved the datasets used for both training and testing the algorithms must be balanced. In the two experiments, there were different phases in which the subjects were doing different things. Therefore, to have balanced datasets, the researchers had to make sure that the training and testing partitions included data chunks or instances from all the stages of the experiments.

To do so, the team decided to create smaller data segments that would be independent of each other. For creating independence between segments, the team left unused those data windows that overlapped in contiguous segments, achieving this way segment disjunction. Finally, the team opted to employ 2/3 of each smaller segment for training the machine learning algorithms and the remaining 1/3 for testing. It is important to bear in mind that the smaller data segments had different durations depending on the experiment that was analysed. Whereas the duration of the longer experiment, Exp 1, lasted for 115s, the segments of Exp 2 had a duration of 95s.

5.3.2 Qualitative analysis

Accomplished the data preparation and having windowed the physiological signals, the team could then pay attention to the qualitative properties of the EDA signals. In this sense, before getting to numerical details, it was important to analyse the EDA qualitatively and to detect the RReps patterns so that they could be related to specific psychological states.

As explained in Subsection 3.2.2, the EDA is a signal that has an accumulative nature and that it is common to decompose it into two main compo-

nents [134]: the Skin Conductance Level (SCL), which corresponds to the slow variations and the Skin Conductance Responses (SCRs), that correspond to the fast variations produced by sweat burst. It was also explained that eccrine sweating only happens when the SNS is active and stimulates the sweat glands. Therefore, as SCRs correspond to the sweat burst, the presence of SCRs in the EDA can be used as a descriptor of SNS getting active. Moreover, some works state that the size of the SCRs is proportional to the magnitude of the SNS activations [135, 136], i.e, the bigger the SNS activation the greater the SCRs.

Taking this phenomenon as a premise, the study presented in this contribution suggested looking at the problem from another perspective and to use the absence of SCRs as an indicator of the SNS not being active. At the same time, these absences also indicate that it is the parasympathetic part of the ANS that is active [118]. Subsequently, it can be concluded that an EDA signal whose SCL component is decreasing is meaningful for relaxation taking place. Nevertheless, unlike it happens with the HR value (increasing or decreasing with SNS and PNS activations), the EDA baseline value is not meaningful for the state of neither the SNS nor the PNS. Hence, a researcher should not be disguised by the baseline value of the EDA as it is not indicative for relaxation. Instead, it is the decreasing tendency of the EDA baseline who reflects that relaxation is taking place.

Accordingly, depending on the shape of the EDA signal, it is possible to identify two types of situations. The first situation is when the SCL component is decreasing in a very linear manner with no sudden bursts. In these situations, it can be said that very deep RResp are taking place. However, there is another type of situations in which although many SCRs occur, the SCL decreases slightly for shorter periods. These type of situations correspond to the second scenario in which RResps happen: when a person tries to relax in the middle of a stressful situation (see Figure 5.2).

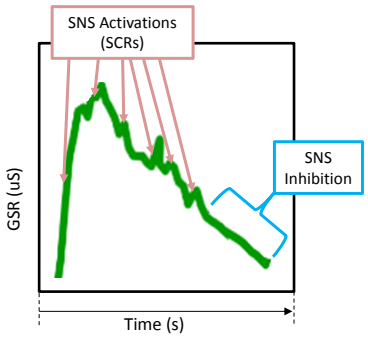

Figure 5.2: The presence and absences of SCRs in the EDA signal can be used as indicators of SNS and PNS activations, respectively.

Once these two patterns of the different types of RResps had been identified, the team could analyse the collected databases and see if those patterns had

been mirrored in the EDA signals collected in the experiments. By looking at the signals, the researchers could see that indeed a majority of the collected signals showed some of these patterns. For example, Figure 5.3 shows that different phenomena took place in two of the registers collected during Exp 1. Subject A showed very deep RResps at both the beginning and ending of Exp 1 (upper graph of Figure 5.3). On the contrary, subject B could not relax at the beginning and, despite he managed to do it at the end of the experiment, the presence of SCRs evidence that the relaxation achieved was not as deep as the subject A's. Seeing these phenomena in the signals, the team checked what each participant's replies to the interviews they had done at the end of the sessions. All the events in the EDA strongly correlated to their answer to the interview, in which subject A had found the initial and final videos to be very relaxing. Concerning subject B, he answered that he was very nervous at the beginning of the experiment and that he had hardly been able to concentrate on the initial video. However, once the puzzle-solving phase had finished, he said that it had been possible for him to relax a bit during the final video.

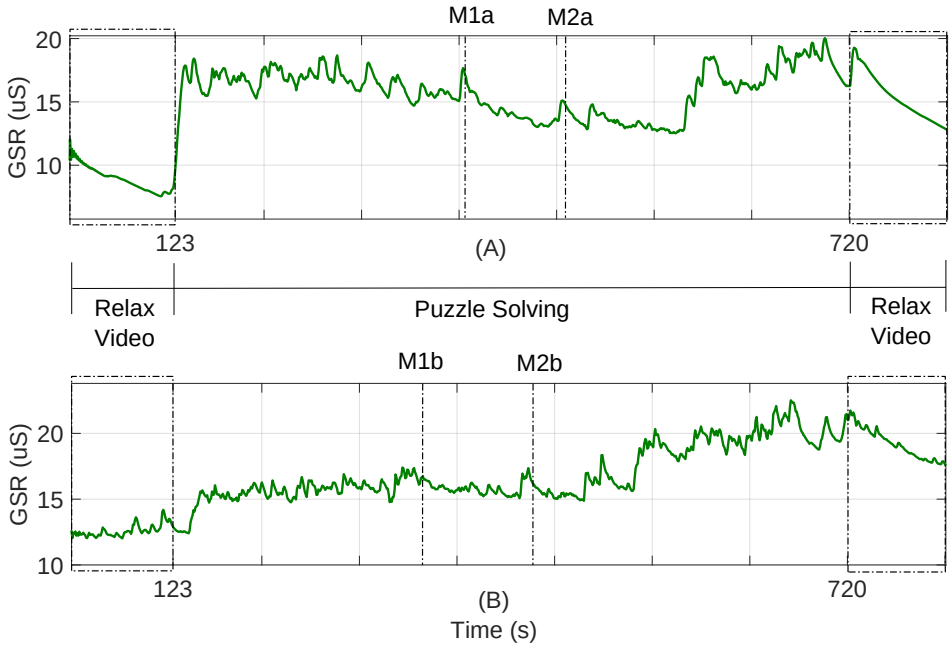

Figure 5.3: During the videos, subject A could relax better than subject B. The EDA of subject B was affected by several SCRs during the videos.

Apart from those overall impressions, the team also analysed the events that had been noted down by the researchers. Following with the example of Figure 5.3, during the experiment of subject A, the researchers had marked down events M1a and M2a as events related to relaxation. For instance, an RResp had been

marked at M1a, when subject A stopped, took a deep breath and tried to relax. Something similar happened at M2a, when the participant stopped for a while and started to consider giving up solving the puzzle. Hence, it is evident that there is lots of relevant information in the behavioural database built from the interviews and the notes taken by the researchers during the experiment. Consequently, the importance of collecting such a database is confirmed as there is a strong link between stress/relaxation and how people feel.

Regarding the registers of Exp 2, the collected signals also showed various types of patterns. Two registers have been plotted in Figure 5.4, corresponding to subject C (top graph) and D (bottom graph). In this case, subject C experienced a very linear and unaltered EDA decrease and, subsequently, it can be concluded that the volunteer achieved great relaxation. This got corroborated by the interview done to subject C at the end of Exp 2, who found the session to be very relaxing. On the other hand, subject D stated that it had been difficult to relax at the beginning and that it was not until the second half of the session that she could get relaxed. Analysing subject D's EDA, her declarations matched closely what happened in the EDA: in the first quarter the EDA suffered many SNS activations due to the nervousness; then, in the second quarter the SCL started to reduce but with many smaller SCRs happening; finally, after 90s, the EDA of subject D started to decrease in a very linear manner.

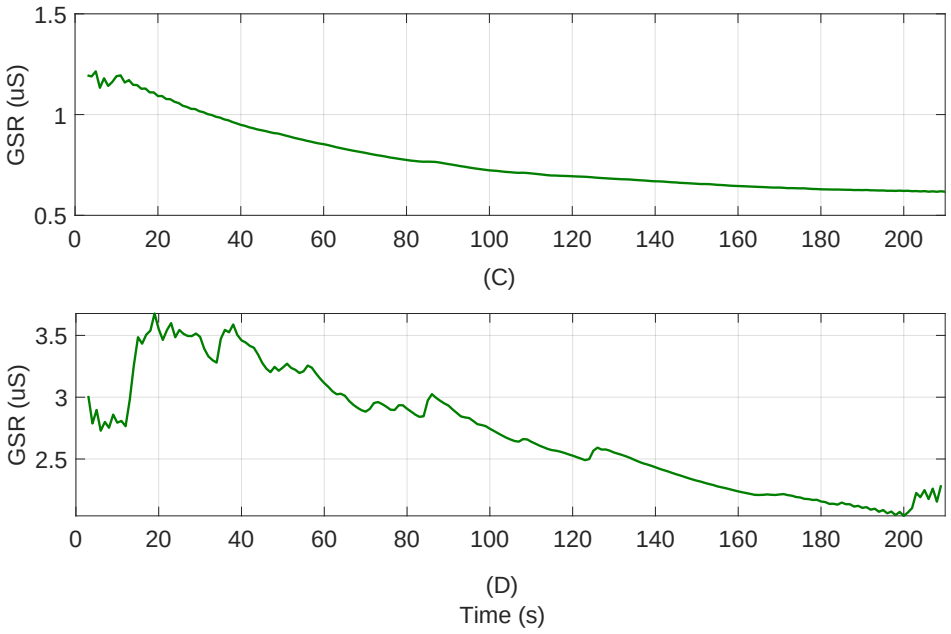

Figure 5.4: Although subject C relaxed deeply, subject D could not relax until the end of the session and, accordingly, the EDA only decreases linearly in the second half of the session.

In the next stage of the qualitative analysis, with the help of the psychologists and clinicians, the team created the labelling system to define the intensities of the RResps. The labelling system consisted of four different labels. The first label, LRResp (Low RResp), corresponds to those cases in which the RResp is not very deep but still there is a physiological change towards relaxation. Then, medium intensity physiological changes were given the MRResp label (Medium RResp). Accordingly, the deepest relaxation responses were labelled as HRResps (High RResp). Last, label NRResp (No RResp) was created for the situations in which no RResps were happening (when the subject is in neutral psychological states or stressing). This approach of using three intensity levels for the RResps is also consistent with the labelling used for assessing stress (recall Chapter 4).

Finally, the last step of qualitative analysis was to label the registers for the application of the supervised learning algorithms of Section 5.4. To do so, a member of the team who had knowledge on both physiology and psychology analysed all the windows and labelled them according to her opinion on the intensity of the RResp happening in the window (if there was any). To avoid biasing her opinion, she was given all the windows in a random order. Furthermore, although only the windows of the EDA signals would be labelled, the member of the team was also given the HRV signal for having a clearer idea of what happening in each window. Later, after she had finished, the rest of the team gathered with the expert neurophysiologists of Cruces Hospital and psychologists of Instituto Burmuin and analysed that window labels. As they had already collaborated with the research team in [69], those experts were accustomed to window approach analyses. Thanks to the experts, the team could get a second opinion on the labelling and corrected the labels in which the experts disagreed with the team member.

5.3.3 Quantitative analysis

The qualitative analysis and the data labelling led the team to the next step of the EDA analysis, the quantitative analysis. This analysis helped the team decide which were the best two features to extract from the EDA for identifying RResps, being one of the features one of the main contributions of this work. Besides, it is in the quantitative analysis that the values of those features were related to the different intensity RResp levels defined in the labels.

The analysis of the signals was done using a 20s sliding window approach, moving every 5s. Other types of signals, such as the periodical signals, can be easily analysed using frequency domain analysis. However, the EDA is a non-stationary and non-periodical signal, which makes it complicated to use frequency-based strategies. On the contrary, time-domain strategies as the sliding window are very suitable for signals with these properties and so, it is common to find its use in the literature [80, 86].

The operation principles of the sliding windows have already been explained in Chapter 4. Anyway, as the contribution presented in this chapter only used the EDA, it is interesting to determine which windows size is the best for the

recognition of RResps. Checking the literature, the team could find that the time windows vary widely from 3s short window to the longest 300s windows [80, 86, 137, 138]. Therefore, after reading in [21] that "*features extracted from the tonic component express the sympathetic tone and are often computed within time windows of 20s, since the upper cut-off frequency of the tonic component is about 0.05 Hz*", the team decided to use 20s windows and to be consistent with what done in [69]. This way, the team could study both phasic and tonic components (SCRs and the SCL) with a window size big enough to contain information and that is not very computationally costly.

Later, having chosen a proper window size, the team extracted different features from the EDA and ended up only using two of them: the slope of the EDA within an analysis window (sEDA) and, the innovative feature, the area enclosed between the instant EDA and the linear regression of the EDA within the window (aEDA). As the EDA is a largely varying signal, these features were normalised inside the boundaries of the analysis window. Thanks to it, the same threshold values could be used in all the analysed windows. As the descriptive name of aEDA might lead the reader to certain confusion, the graphical representation of the two has been depicted in the below Figure 5.5.

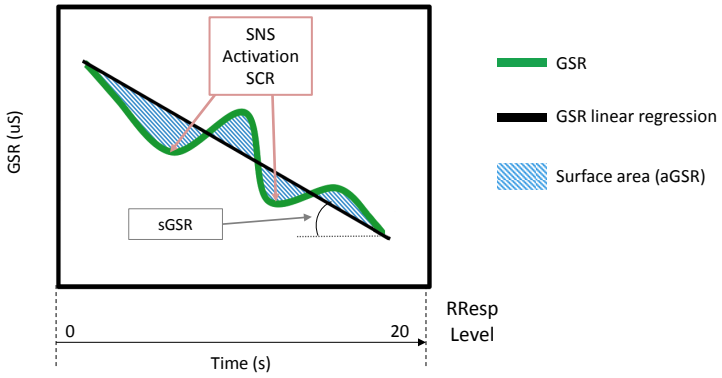

Figure 5.5: The selected features are the slope of the EDA within the analysis window and the surface enclosed between the linear regression of the EDA and its instant value.

Paying attention to the first feature, sEDA, it seems evident that a positive slope of the EDA means that the person is getting stressed and so, that no relax is taking place. Therefore, sEDA necessarily has to be negative within the analysis window if an RResp is happening. Concerning aEDA, taking into account the physiological relationship between the absence of SCRs and PNS activations, it seems clear that the EDA signal will decrease in a very linear manner, without SCRs, when the PNS activation is maintained in time. Subsequently, if the EDA decreases linearly then (due to the absence of SCRs), then the shape of the EDA will be very close to its linear regression within that

analysis window. Consequently, the aEDA feature results in very small values when the PNS activation is intense and long in time. On the contrary, if sudden SCRs altered a PNS activation, they would make the EDA signal to bounce around the linear regression, producing higher aEDA values. Thus, the smaller aEDA the deeper the RResp.

Many works in the literature extract features from the EDA for both emotional recognition and stress detection [21, 139, 140]. sEDA is a clear example of it being used since the middle of the 20^{th} century. Apart from the slope, it is possible to find features belonging to other types of analysis: statistical parameters, increments, nonnegative convolutions, areas, etc.([80,123–125,135]). If the proposed aEDA feature should be classified in one of these categories, it would fit in the ones related to extraction of areas. However, most of the area type features that are extracted from the EDA require the signal to be decomposed into the tonic and the phasic components. Later, most studies working with areas extract the areas under the curve and other area related features from the separated phasic and tonic components [23, 141]. Also, some research avoid decomposing them and focus on the morphological properties of SCRs, such as rising times and amplitudes of the first half of SCRs [24,25]. Nevertheless, to the extent of the authors' knowledge, the proposed aEDA feature is innovative and merges the strategies of these works only taking the most beneficial parts. Not implying component decoupling, the aEDA has information from both tonic and phasic components and, simultaneously, it is not computationally costly because the linear regression calculation is an easy procedure.

5.4 Application of machine learning

This section is divided into two subsections. The first subsection, Subsection 5.4.1 will explain the data classification done with machine learning techniques. In this process, the team used different supervised learning algorithms to classify the windows as belonging to one of the four aforementioned labels: NRResp, LRResp, MRResp and HRResp. Subsection 5.4.1 will also present the results of those classifications, which were used to select the best performing type of algorithm for classifying RReps. For the simulations and performance calculations of the two experiments, the team used the same partitioning strategy: 2/3 of the data were used for training and 1/3 for testing the algorithms.

After doing this process for the two experiments' datasets (Exp 1 and Exp 2), the team saw that decision trees (DT) obtained good results in both of them. Accordingly, they decided to merge both DTs and create a synthesized rule system for general use in any of the experiments. The process via which the team designed this rule system will be explained in Subsection 5.4.2.

5.4.1 Selection of the best classifying algorithm

There are many machine learning algorithms and depending on the application and on the databases to classify, their performance might vary considerably.

Therefore, when looking for an algorithm to classify data it is always interesting to do a comparison between various algorithms so that the best performing one can be selected. For the case of this study, the team decided to use only supervised learning algorithms; as the signal windows had already been labelled there was no point in using clustering algorithms.

To do this comparison between algorithms the team opted for using Weka's implementation of 12 algorithms with their default algorithm settings [142]. The team decided to use the default parameters because fine-tuning an algorithm requires time and sometimes optimising the classifier performance for a class may worsen its performance for the others. Therefore, the team thought that using default would be good as, instead favouring any algorithm in detriment of the others, it would rather indicate the suitability of each algorithm under fair-play conditions.

The team selected 12 algorithms considering them to be state of the art algorithms within the algorithm families that base on the same type of paradigm (refer to Section 3.4 for more information). The 12 supervised learning algorithms that were chosen are 1R rule, Decision Tree (DT), k-NN (1-NN and 5-NN), Naive Bayes (NB), Radial-Basis Network (RBF), Support Vector Machine (SVM), Logistic Regression (LR), Ada Boost (AdaB, combining 10 decision trees), Bagging (Bag, combining 10 decision trees), Random Forest (RF) and Multi-Layer Perceptron (MLP). The results of this comparison for the data of the two experiments can be checked in Table 5.1, where the given performance indicators correspond to the presented in Subsection 3.4.4.

Table 5.1: Results of the performance of the 12 classifiers.

Classifier	**Exp 1**			**Exp 2**		
	P	**R**	**F1**	**P**	**R**	**F1**
1R	0.649	0.682	0.665	0.773	0.797	0.785
DT	0.990	0.990	0.990	0.992	0.992	0.992
1-NN	0.978	0.978	0.978	0.969	0.969	0.969
5-NN	0.978	0.977	0.978	0.966	0.965	0.965
NB	0.822	0.797	0.809	0.859	0.823	0.840
RBF	0.955	0.955	0.955	0.953	0.953	0.953
SVM	0.913	0.912	0.913	0.814	0.739	0.740
LR	0.888	0.888	0.888	0.952	0.951	0.951
AdaB	0.989	0.989	0.989	0.992	0.992	0.922
Bag	0.990	0.989	0.989	0.992	0.992	0.992
RF	0.991	0.992	0.991	0.992	0.992	0.992
MLP	0.952	0.951	0.952	0.892	0.864	0.878

The results of Table 5.1 show that DTs, RF and the ensembles of classifiers (AdaB and Bag using 10 DTs) were the best classifiers, obtaining very similar results. Thus, seeing those results, the team chose to continue working on the

DTs because the other algorithms do not have the same explanatory capacity as DTs. Being very easy to understand makes DTs to be perfect for users without any computer science background. Moreover, DTs are very simple and the branch splitting criteria can be easily modified. This was very useful for the work presented in the next subsection, where the DTs specifically designed for each of the experiments were merged into a single synthesized rule system useful for both databases. The DTs produced during the comparison of the algorithms corresponds to Weka's J48 pruned tree algorithm (Weka's default DT setting), which corresponds to Quinlan's C4.5 algorithm [143].

5.4.2 The synthesized rule system

Being selected as the best algorithm for having one of the best performances and being easily understandable, the team considered that it was worth taking the DTs to a higher level of analysis. To do so, the team wanted to see and compare the rules of the DTs generated for Exp 1 and Exp 2, which can be seen in Figures 5.6 and 5.7 respectively. There, the main splitting points have been marked with colour boxes (those marks will be used later on this subsection).

```
J48 pruned tree Experiment 1
------------------------------------------------------
aEDA <= 0.069998
|   sEDA <= 0.000021
|   |   sEDA <= -0.04861: -3 (1494.0)
|   |   sEDA > -0.04861
|   |   |   aEDA <= 0.039991: -3 (361.0/45.0)
|   |   |   aEDA > 0.039991: 0 (92.0)
|   sEDA > 0.000021: 0 (295.0)
aEDA > 0.069998
|   sEDA <= -0.050193
|   |   aEDA <= 0.199712: -2 (1846.0/16.0)
|   |   aEDA > 0.199712
|   |   |   sEDA <= -0.300083: -1 (689.0/13.0)
|   |   |   sEDA > -0.300083
|   |   |   |   sEDA <= -0.289222
|   |   |   |   |   sEDA <= -0.294407: 0 (11.0/2.0)
|   |   |   |   |   sEDA > -0.294407
|   |   |   |   |   |   sEDA <= -0.292119: -1 (4.0)
|   |   |   |   |   |   sEDA > -0.292119
|   |   |   |   |   |   |   sEDA <= -0.291568: 0 (2.0)
|   |   |   |   |   |   |   sEDA > -0.291568: -1 (3.0/1.0)
|   |   |   |   sEDA > -0.289222: 0 (539.0/6.0)
|   sEDA > -0.050193: 0 (3374.0)
```

Figure 5.6: The rules of the DT for Exp 1.

```
J48 pruned tree Experiment 2
------------------------------------
aEDA <= 0.069816
|   sEDA <= -0.071428: -3 (247.0)
|   sEDA > -0.071428
|   |   sEDA <= -0.017014: -3 (4.0)
|   |   sEDA > -0.017014: 0 (3.0)
aEDA > 0.069816
|   sEDA <= -0.2889
|   |   aEDA <= 0.203202: -2 (128.0/3.0)
|   |   aEDA > 0.203202
|   |   |   sEDA <= -0.3014: -1 (84.0)
|   |   |   sEDA > -0.3014
|   |   |   |   sEDA <= -0.2939: 0 (2.0)
|   |   |   |   sEDA > -0.2939: -1 (3.0)
|   sEDA > -0.2889
|   |   aEDA <= 0.196464
|   |   |   sEDA <= -0.049283: -2 (49.0)
|   |   |   sEDA > -0.049283: 0 (50.0)
|   |   aEDA > 0.196464: 0 (457.0)
```

Figure 5.7: The rules of the DT for Exp 2.

Knowing how these DTs operate, the team wanted to study their performance with a higher degree of detail by seeing how good they were, not at a general level, but a class-specific level. The classification results of each DT for each class have been presented in Table 5.2. The first two columns of Table 5.2 indicate the labels for the different RReps levels and the number of instances that had been catalogued by the experts for each RResps level. Then, the performance metrics of Subsection 3.4.4 are presented from the third to the eighth columns. Please note that there are two rows for which only the precision (P), recall (R) and F-1 score have been given. These two rows correspond to the average results for the data of the whole datasets of Exp 1 and Exp 2. Accordingly, the labels of those rows have a total of instances equal to the sum of the quantities belonging to the four RResp levels in each experiment.

Additionally, as the main rules shown in Figures 5.6 and 5.7 had similar thresholds, the team also decided to cross the DTs. This means that they used the DT created for Exp 1 with the data of Exp 2 and, consequently, they did the same in the opposite direction. The results of the classification of the crossed systems are also given below in Table 5.2, from the ninth column to the fourteenth.

Table 5.2: The performance indicators of each classifier using them with the original dataset with which they were designed and with other experiment's dataset.

State	Manual label	DT						Crossed DT					
		TP	FN	FP	P	R	F1	TP	FN	FP	P	R	F1
Exp1	4354				0.990	0.990	0.990				0.965	0.965	0.965
NRResp	2177	2153	24	6	0.997	0.989	0.993	2140	36	95	0.957	0.983	0.970
LRResp	349	342	7	15	0.958	0.980	0.969	335	14	11	0.968	0.960	0.964
MRResp	923	909	14	4	0.996	0.985	0.990	901	22	14	0.985	0.976	0.980
HRResp	905	905	0	20	0.978	1.000	0.989	824	81	34	0.960	0.910	0.935
Exp2	513				0.992	0.992	0.992				0.992	0.992	0.992
NRResp	257	256	1	1	0.996	0.996	0.996	256	1	1	0.996	0.996	0.996
LRResp	44	43	1	1	0.977	0.977	0.977	43	1	2	0.956	0.977	0.966
MRResp	87	85	2	1	0.988	0.977	0.983	85	2	1	0.988	0.977	0.983
HRResp	125	125	0	1	0.992	1.000	0.996	125	0	0	1.000	1.000	1.000

The results given in Table 5.2 show that when crossed, the classifiers performed very good, almost at the same level as when they were used with their original dataset. This confirmed the suspicions of the researchers, who thought this might happen as a consequence of the rules marked in Figures 5.6 and 5.7 were very similar. Consequently, the team considered that merging both classifiers into a unified classifier would be interesting.

To calculate the new classifier the team took the rules of the original DTs and calculated new splitting criteria. These new points were obtained by calculating the average value of the splitting points of the two DTs. Then, for the avoiding rare numbers, the values were rounded keeping only two decimals. For instance, taking the splitting criteria of Figures 5.6 and 5.7 for the LRResp class (the ones inside red boxes), the splitting points of the new classifier would be calculated as done in Equations 5.1 and 5.2.

$$\begin{aligned} aEDA &= (aEDA_{Exp1} + aEDA_{Exp2})/2 \\ &= (0.1997 + 0.2032)/2 = 0.2014 \approx 0.2 \end{aligned} \tag{5.1}$$

$$\begin{aligned} sEDA &= (sEDA_{Exp1} + sEDA_{Exp2})/2 \\ &= -(0.3000 + 0.3014)/2 = -0.3007 \approx -0.3 \end{aligned} \tag{5.2}$$

However, there is an exception to the calculations of the splitting criteria of label HRResps. If attention is paid to Figure 5.6 and 5.7, the reader will realise that the DT of Exp 1 has a case that has not been considered in the other DT: when sEDA<-0.04861 and aEDA≤0.03999. Consequently, leaving these cases out of consideration of the new synthesized rule system would worsen

considerably its performance. Therefore, to avoid false detections, the team opted to add this rule using a logic OR to the splitting criterion calculated with the method already explained. Consequently, after calculating all the splitting criteria, the team obtained the new synthesized rule system that is presented in Table 5.3.

Table 5.3: The decision rules the new synthesized rule system.

State	Synthesized/unified rules
LRResp	(sGSR<-0.3) AND (aGSR>0.2)
MRResp	(sGSR<-0.05) AND (0.07<aGSR<0.2)
HRResp	((-0.05<sGSR<0) AND (aGSR<0.04)) OR ((sGSR<-0.06) AND (aGSR<0.07))
NRResp	Any other situation

The new system was then taken into test, obtaining the results presented in Table 5.4. These results show that the new system has a very good performance, which in the case of Exp 1 obtained better results that the DTs designed expressly for it ($F1_{\text{SYNTHESIZED}}$=0.994 vs $F1_{\text{DT-Exp 1}}$=0.990).

Table 5.4: Performance indicators of the unified classification system.

State	Manual label	Unified rules					
		TP	FN	FP	P	R	F1
Exp1	4354				0.994	0.994	0.994
NRResp	2177	2176	1	9	0.996	1.000	0.998
LRResp	349	339	10	12	0.969	0.969	0.969
MRResp	923	909	14	4	0.996	0.985	0.990
HRResp	905	905	0	0	1.000	1.000	1.000
Exp2	513				0.992	0.992	0.992
NRResp	257	257	0	2	0.992	1.000	0.996
LRResp	44	42	2	1	0.977	0.955	0.966
MRResp	87	85	2	1	0.988	0.977	0.983
HRResp	125	125	0	0	1.000	1.000	1.000

Finally, the team considered interesting to verify empirically that the new algorithm performed correctly over time. To do so, they decided to input the EDA signals of the experiments into the algorithm and see if the outputs given over time were correct. For the sake of consistency, the following Figures 5.8 and 5.9 respectively show the new algorithm's outputs of for the EDA signals of subjects A and B from Exp 1 and C and D from Exp 2. For these simulations the RResp levels have been represented numerically, ranging from 0 to -3: 0 corresponds to NRResp, -1 to LRResp, -2 to MRResp and -3 to HRResp.

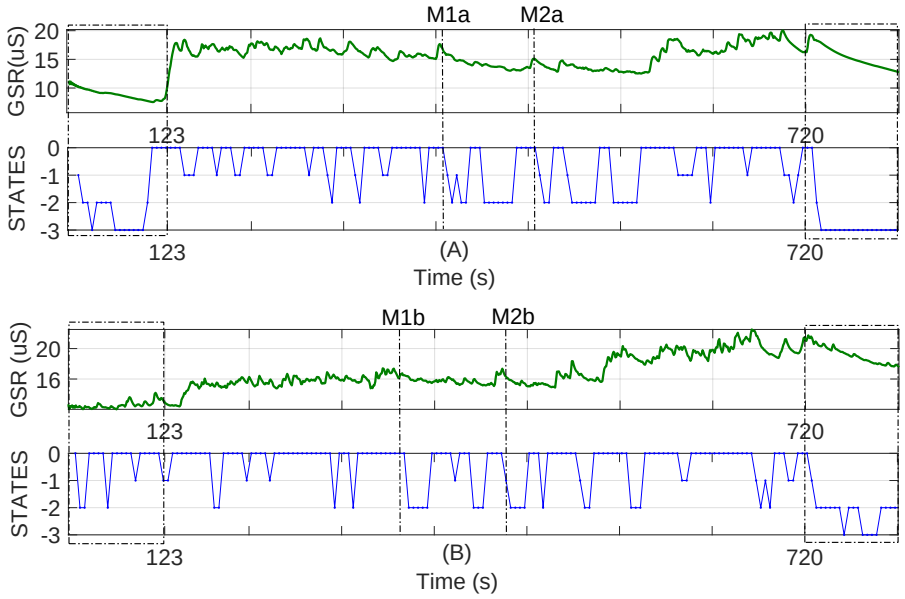

Figure 5.8: During the videos, the RResps measured for subject A have much higher intensity than subject B's.

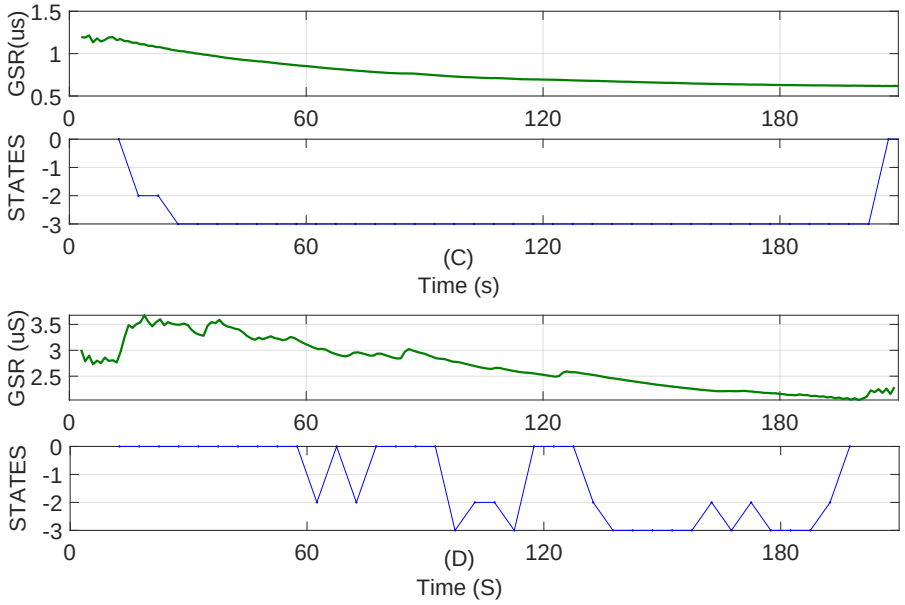

Figure 5.9: Subject C achieved much deeper RResps compared to subject D.

In Figures 5.8 and 5.9 the EDA signals have been plotted in the green at the top part of the graphs of each user. Below them, the synthesized systems RResp level output has been depicted in blue and it updates its value every 5s according to the sliding window approach. In Figure 5.8 it can be seen that during the relaxing videos of Exp 1, compared to subject B, the algorithm gave much higher RResp outputs for subject A's EDA. This estimation is coherent to what the subjects stated in their interviews. Besides, the algorithm also gave medium intensity RResps for the events marked as M1a, M1b, M2a and M2b. The good performance also gets reflected for the cases of Exp 2 shown in Figure 5.9. There, subject C got high levels of relaxation during the whole session, but, on the contrary, the algorithm's outputs did not indicate that maintained RResps until the second half of the session.

5.5 Summary

New paradigms of medicine and psychology pose relaxation as a key tool for preventing and helping to heal certain diseases from the clinical psychology discipline, such as stress, anxiety disorders or depression. Thus, getting to know how and when patients get relaxed would be helpful for professionals when assessing patients. Moreover, as relaxation produces no side effects but only benefits to the organism, creating affordable technological solutions for helping users understand how relaxation takes place in their organism would be very helpful for the public health of a fast-living society like ours.

Nevertheless, there is not much technical literature in the field of computer sciences that study relaxation, and when they do so, most of them analyse it transversally and focus on stress. Apart from these works, some others study relaxation within an emotional recognition approach, which is also different from detecting stress from the perspective of physiological computing. Therefore, considering there is a gap in the literature, the team decided to study the physiological changes that relaxation produces in the organism, the RResps, and to subsequently design an automatic system capable of detecting and classifying those RResps.

The study analysed RResps from two different points of view. First, the team collected the signals from an experiment (Exp 1) that produced RResps during a stressful situation. Second, the team also collected signals from an experiment (Exp 2) especially aimed to produce relaxation in the participants. Analysing these data with the help of experts in physiology and psychology the team could detect and define the boundaries in the EDA signal that reflect different levels of RResps, cataloguing them with four labels: NRResp (0), LRResp (-1), MRResp (-2) and HRResp (-3).

As said, the patterns that permitted to detect those four levels of RResps were found thanks to the thorough analysis of the EDA signal of sweating. More certainly, the patterns were found in the two features that were extracted from the signal: the slope and the area created by the instant EDA signal and its linear regression. Although the first has been used in the literature since

decades, the second feature is an innovative feature that gives a little twist to the feature extraction related to the areas produced by the EDA curve, which normally tend to analyse the phasic and tonic components separately.

Finally, having detected the EDA features and patterns that indicative of the RResps, the team tried different supervised learning algorithms to find the best performing algorithm for detecting the RResps taking place in the signals collected in Exp 1 and Exp 2. Despite different algorithms obtained similarly good performances for the two experiments, the group opted to choose DTs as unlike the others, they have explanatory properties. Later, the two DTs created for signals of the experiments were further studied. During this final study, the team saw that the rules of both DTs were similar and that they also performed well with the dataset of the experiment with which they had not been trained. Therefore, as a last step of the study, the team merged the most relevant rules of the two DTs to build a simpler yet similarly effective synthesized rule system.

Limitations and future lines

This work has shown that it is possible (and also interesting) to detect physiological responses towards relaxation applying machine learning algorithms to just two features coming from a single physiological variable: the EDA. Nevertheless, it is a matter that has not been sufficiently covered by the literature and so, there is still a lot to research. For instance, as the number of participants in Exp 2 was relatively reduced, the population that took part in this study mostly consisted of students of a very specific age range. Therefore, it would be convenient to expand the study to different ages and a wider population in general and, then, see if the extracted features and the classifiers perform as well as they did with the collected data from the experiments.

Apart from that, as also commented in Subsection 4.7, one of the most complicated problems in this type of experiments is the labelling of data. As they imply subjective concepts such as stress and relaxation, it is complex to determine whether the labels given to the registers are trustworthy. Most of the times, the labels given to the instances rely solely on the answer that the participants give to interviews and questionnaires at the end of the data collection experiments. Nevertheless, what stated in these interviews reflects only a general impression and leaves out what happening in eventual situations. This is the major weak point of most of the studies that involve impressions and feelings.

However, being a weak point, the problem arising from data labelling may also be the most relevant open line to research in the future. For the writer of this thesis, automatic detection tools and strategies as the one presented in this contribution can result very helpful for online labelling of the signals collected in future experiments. This way, if this detection algorithm (or any other depending on the experiment) was run while the signals are being collected, then it would be possible for the researchers to have all the signals labelled even before interviewing the participants. This way, the researchers would get

the possibility to ask the participants about eventual situations in which the algorithms reacted very remarkably and could directly contrast both the labels and the veracity of the detection algorithm itself.

Besides, it would be very interesting to implement the solution in a portable hardware platform. Whereas most of the solutions to problems found in the literature of physiological computing imply collecting various signals, the proposed system only uses one signal and, so, it should be relatively easy to implement (if compared to others).

Finally, in this study, the team saw that RResps produced by relaxation techniques are normally more intense than those produced during a stressful situation. However, it might be possible to find certain patterns hidden in the physiological reactions that indicate to which type of situation does the detected RResp belong to. Consequently, the team considers that it would be interesting to give an extra twist to the algorithm to see if is possible to differentiate between these two types of RResps.

Chapter 6

Contributions to ERD/ERS Brain-Computer Interfaces

6.1 Introduction

The first human electroencephalogram was collected in 1924 by Hans Berger [144]. Nevertheless, it was not until the decade of 1970 that brain activity caught the attention of the researchers of engineering disciplines. It was in those years that the first Brain-Computer Interfaces (BCI hereinafter) were produced. Those BCI systems mostly focused on helping people restore the auditive, vision or limb moving capabilities they had previously lost. Therefore, it was only in clinical environments that BCI systems were used.

However, the fast development in electronics of the last decades have made affordable computer devices with high computational capabilities. Related to that, the number of studies around machine learning has grown proportionally to the power of CPUs. At the same time, the number of technological methods to capture the activity of the brain has also increased [145], and so, the same has happened with the number of companies producing EEG headsets. The technical and economical synergy produced between disciplines has made much easier to collect, detect and classify brain activity patterns using computer devices. Consequently, the use of BCI systems has expanded from the medical environments to many other applications, such as videogames, caregiving applications for elderly people, domotics or even the military.

Although there are some differences in the implementation, all BCI systems have the same basic structure [146, 147]. First, EEG signals are collected using electrodes, which can be either wet (if they require using contact gel) or dry. Then, the signals are processed by the computer device in search of specific signal patterns. Finally, the computing unit will send a command to an end-effector in those cases that a known pattern is detected. As explained in Subsection 3.5, there various types of BCI which differ from each other in the way they produce the desired brain activity patterns: event-related potentials (ERPs),

steady-state visually evoked potentials (SSVEP), ERD/ERS motor imagery...

This work presents the contribution done to the state of the art of ERD/ERS motor imagery BCI, proposing to combine supervised and unsupervised learning algorithms in a hierarchical classification system for a 5 class motor imagery classification problem. The chapter will develop in a similar manner to Chapter 4 and 5: the context and literature related to the work will be explained in Section 6.2; then Section 6.3 will present BCI Competition III dataset IIIa and the modification done to expand it from being a 4-class problem to a 5-class problem. Later, the analysis done using only supervised learning algorithms will be given in Section 6.4 and, as the results were not satisfactory, a new solution will be presented in Section 6.5, were the team proposes to combine the K-means unsupervised learning algorithm with SVMs in a hierarchical classification system. Finally, the experimental results will be presented in Section 6.6 and the work will be summarised in Section 6.7 along with the limitations of the work and the future lines.

6.2 Context and related work

Apart from the study of mental states of the previous two contributions, researching about brain-activity is also among the interests of the research team. Concerning this area, the team considered interesting to apply their experience to tackle the classification problems related to brain-computer interfaces (BCI). However, collecting electroencephalograms (EEGs) is a far from simple task, and so, instead of creating their experimental design, the team opted for using an already existing dataset for the study. Therefore, as they had already worked with it [148], the team chose to use dataset IIIa from the BCI Competition III database, which is a competition referenced very often in the literature. In it, participants were given a set of EEG signals and they have to perform the best classification possible for a specific motor imagery problem.

As already explained, there are different types of BCI systems. However, having chosen to use dataset IIIa from BCI Competition III [149], the type of the BCI system for this study already was selected: it was an ERD/ERS motor imagery BCI. These systems are among the most commonly studied in the literature [150–153]. They base on detecting variations of the brain potentials that are produced when the user moves or thinks of moving a part of the body [154, 155]. To train these systems, users are told to think about moving a part of the body after seeing or hearing a specific signal. Therefore, it can be said that these systems work synchronously, being time-locked to specific cues.

Normally, motor imagery systems intend to discriminate between various classes of brain activity patterns. The simplest systems only try to determine whether it is the right hand or the left hand that wants to be moved. Some others get more complex and try to detect movement intentionality of various parts of the body. Nevertheless, sometimes researchers focus too much on identifying brain patterns related to intentional movements and forget that human brains do not continuously send commands to the limbs. Being like that, detecting the

absence of intentional control states (non-intentional control, NC) can be said to be as important as correctly classifying the different intentional control states (IC). If not, the system would detect many false positives and produce several undesired end-effector actuation, resulting in user discomfort and discarding the BCI system as an adequate technological solution [156, 157].

Considering this, some studies have catalogued motor imagery as NC pattern rejection problem in which the unrejected IC patterns have to be correctly differentiated. This can be done either by classification [158] or by setting thresholds for EEG signals below which brain activity is ignored [156, 159, 160]. Therefore, it can be said that the problem can be divided into two classification levels: first, NC and IC patterns have to be separated and, later, differentiate the classes of the instances labelled as IC. Whereas the first classification tries to keep the false positive rate (FPR, $FPR = FP/(FP + TN)$) as low as possible for not being refused by the user (scoring over 10% is considered not to be good enough [158, 161, 162]), the second classification searches to maximise the accuracy classifying ICs.

Following this line, the contribution work explained in this chapter looks to give a solution to the ERD/ERS motor imagery problem presented in the BCI Competition, but adding a new NC class for representing the situations in which no movement is intended. Consequently, the addition of the NC class also increases the difficulty of the problem, as the FPR of the system has to be kept below 10%. The proposal presented in this chapter uses a two-level hierarchical system. First, it uses the K-means unsupervised learning algorithm for the initial NC-IC classification, keeping the FPR under 10%. And then, in the second level classification, it uses a Support Vector Machine (SVM) to determine the exact class of the instances classified as IC by the K-means algorithm.

6.3 Data and experimental methodology

6.3.1 The original dataset

As explained in the previous sections, the team decided to star the research using the IIIa dataset from the BCI Competition III database. This dataset has already been used several times in the literature and so it is a perfect dataset for comparing the results of the proposal against other references in the literature. Furthermore, being easily accessible, using this dataset helps other researchers reproducing the work presented in this chapter.

BCI Competition III dataset IIIa contains EEG activity from 3 volunteers (named K3b, K6b and L1b) participating in a motor imagery data collection experiment. For the experiment, the researchers of [149] placed 60 electrodes along the scalp of the volunteers to collect the EEG signals. These experimental sessions consisted of several trials in which the participants had to think of moving a specific part of their body when a PC screen gave them a signal indicating to do so. Depending on the given signal they had to think of moving the right hand, the left hand, the tongue or the feet. Each motor imagery

trial followed the same experimental protocol, which is depicted in Figure 6.1. For the moment the reader will be asked to ignore the "IC" and "NC" class annotations of Figure 6.1, they will be clarified later on.

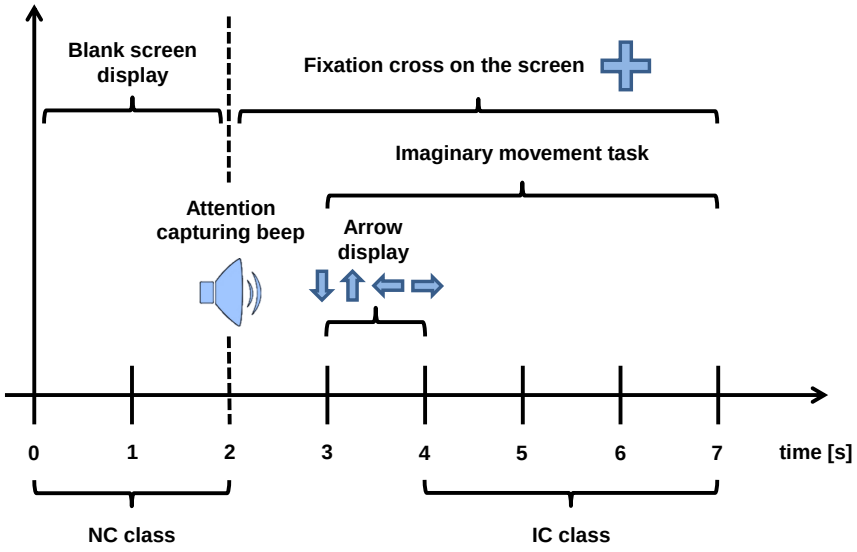

Figure 6.1: **Structure of a imaginary movement trials.**

As shown in Figure 6.1, each trial lasted for 7s during which different events took place. In the first 2s the screen remained blank and at t=2s a beeping sound was reproduced to capture the volunteer's attention. Since then, the screen showed a fixation cross "+" that would remain displayed until the end of the trial. After the beep, during the [2-3]s interval, the screen would solely show the cross and, at t=3s, an arrow would appear for 1s (i.e., until t=4s). These arrows point to one direction, indicating the body part to think of moving of. If the arrow point up it meant that the volunteers had to think of moving their tongue; when pointing down it meant thinking of moving the feet, and, lastly, left and right-pointing arrows indicated the imaginary movement of the left and right hands respectively. Although the arrows were only displayed from t=3s to t=4s, the volunteers were asked to keep on thinking of moving the part of the body since the apparition of the arrow until the end of the trial, which is from t=3s to t=7s. Finally, after the 7s trial, the participants were given a 2s break to rest before starting the next signal collection trial.

To finish with the presentation of the original dataset, the signals were sampled with a 250Hz sampling rate and after being collected they were band filtered in the [1-50]Hz range and notch filtered at 50Hz for removing any noise coming from the power-line. After the three subjects' sessions had finished, the dataset

consisted of 840 instances: 360 cases for subject K3b, 240 for K6b and 240 for L1b. During their sessions, each participant had the same number of trials for each part of the body. Hence, the classes inside the dataset are balanced, which prevents classification biasing towards any class.

6.3.2 Expansion and preprocessing of the original dataset

Expanding the original dataset

So far, this document has only explained the original dataset, which only considers intentional movement control states (IC) for given parts of the body. Nevertheless, as explained in section 6.2, a BCI system needs to be capable of recognising the absence of movement as it prevents the system from triggering undesired end-effector actuations. Being aware of this problem, the team decided to make the problem closer to real-life application and to expand the original dataset creating a new one that included non-intentional control states (NC).

For the expansion of the dataset, the team opted for taking the first seconds of the trials and label them as belonging to the NC class. Initially, as they had done in their previous approach to this dataset [148], the team considered taking the EEG signals until t=3s. The reason to do so was because it was not until then that the participants were told which limb to move imaginatively (recall Figure 6.1). Nevertheless, they discarded this idea because the beeps of t=2s would most probably alter the brain potentials and bias the classification. Instead, to avoid the problem, the team only catalogued the EEG signals from t=0s to t=2s as belonging to NC. On the contrary, there was no such a problem for the IC instances, because Blankertz's team had been cautious and prevented the influence of the arrows appearing and disappearing by only using the [4-7]s signals intervals for building the dataset. The annotations of the bottom part of Figure 6.1 give a graphical indication of the trial time intervals that were used for creating the 4 types of IC and the NC class instances.

After completing the expansion process the newly created dataset included a total of 1400 instances. From those 1400 cases 600 came from subject K3b, having 360 IC instances and 240 NC instances and, the other 800, came from subjects K6b and L1b (400 for each subject: 240IC and 160NC).

Preprocessing and feature extraction

Once the team got the new dataset, they had to do some preprocessing and preparation work over it before approaching the classification problem. First of all, as using 60 EEG signals was too much, the team thought simplifying the problem would result in favour of a better classification. Consequently, following the work of [163], the team applied the multi-class common spatial pattern (CSP) method to extract the 5 most meaningful projections for each class (please refer to Subsection 3.2.4 for the explanation of CSP). So, having 4 IC classes and 1 NC class, this process was applied 5 times (once per class).

Having kept only the most meaningful projections the team though that it would also be good to filter out the information of brain activity bands not related to movement. Accordingly, the 5 most significant projections of each class were band filtered, keeping only 3 frequency bands: 8-12Hz, 12-20Hz and 20-30Hz. Recalling what explained in Subsection 3.2.4, the first division of these bands is related to Alpha and Mu rhythms, which are highly related to movement and take place within the 8-13Hz range. Besides, lying in the 14-30Hz, the Beta band is related to concentration. Thus, it is also interesting to analyse this band as it is prompt to vary during motor imagery sessions. Anyway, it is a very wide frequency band and so it made sense dividing it into two smaller bands.

The last step before creating the partitions for the classifier training and testing was to extract the features of the information that had been kept after the filtering. As the team wanted to focus the interest of the contribution on classification, they decided to continue with AlZoubi's approach [163] and extract the following seven features: the minimum and maximum voltage values, mean voltage, voltage range, average power of the signals, rate of zero voltage crossings and rate of samples above zero volts. The mathematical formulae with which these features were calculated can be consulted in Table 6.1.

Table 6.1: Mathematical descriptions of the extracted 7 features.

Extracted features	
Minimum voltage	$V_{\min} = min(X_n)$
Maximum voltage	$V_{\max} = max(X_n)$
Mean voltage	$V_{\text{mean}} = \frac{1}{N} \cdot \sum_{n=1}^{N} X_n$
Voltage range	$V_{\text{mean}} = V_{\max} - V_{\min}$
Average signal power	$P_{\text{signal}} = \frac{1}{N} \cdot \sum_{n=1}^{N} X_n^2$
Rate 0 voltage crossings	$R_{\text{0-cross}} = \frac{1}{N-1} \cdot \sum_{n=1}^{N} [sign(X_{\text{n-1}}) \neq sign(X_{\text{n}})]$
Rate of samples above 0 volts	$R_{\text{positive}} = \frac{1}{N} \cdot \sum_{n=1}^{N} (X_{\text{n}} \geq 0)$

Dataset partitioning and feature selection

As the problem analysed in the new dataset is a classification problem, it was necessary to partition the datasets for training and testing the algorithms that would be used. Being considered a two-level classification problem, the expanded dataset had to be divided into three smaller datasets. Two of the partitions, TrainingSet1 and TrainingSet2, would be used for respectively training the classifiers of the first and second levels. Then, the third partition, named TestSet, would be used for testing the whole system.

On the one side, for subject K3b, TrainingSet1 and TestSet had both 240 instances, from which 120 were NC and the other 120 belonged to the IC classes

(30 instances for each class). Concerning TrainingSet2, it was smaller than the other partitions as the second level classifier only had to distinguish among the four IC classes. Therefore, TrainingSet2 consisted of 30 instances for each IC class (left hand, right hand, tongue and feet), i.e., a total of 120 instances. On the other side, subjects K6b and L1b had less data and so, their partitions followed a similar distribution but with less data in each partition: TrainingSet1 and TestSet contained 160 instances (80 NC and the other 80 equally distributing the IC classes) and TrainingSet2 consisted of 80 IC instances (20 cases for each class).

As the researchers wanted the results of their analysis to be rigorous, the dataset partitioning process separated the instances randomly into each subjects training and testing dataset. Furthermore, the process was repeated 5 times so that the training and testing of the algorithms could be thoroughly analysed. Also, the training of both the first level and the second level classifiers were trained using a 10-fold cross-validation approach (10-fold CV from now on). Consequently, the results of the training process presented in the following section will correspond to the average scores of the 50 trainings that resulted from this 5-run 10-fold CV process. On the contrary, the testing of the whole system did not follow a cross-validation methodology and so, the results given for the system tests will represent the average result of the algorithms over 5 runs. Also, as 5 training and testing runs were done, the scores given for both the two trainings and the testing will include the standard deviation over those 5 runs.

Finishing this section and before going onto the classification, the team had 5 variations of the dataset partitions in which each instance had 525 features: 5 projections were selected each class (1 NC and 4 IC) from the CSP processing, each projection was divided into 3 frequency bands and 7 features were extracted from each band (5x5x3x7=525). This might result in overfitting the classifiers and so, to avoid the curse of dimensionality, the team decided to dismiss the useless or redundant information using the correlation-based feature selection method (see Subsection 3.4.5) on Weka platform. This was done by creating an empty feature set and then adding features under the best first greedy searching approach. After applying it to the training partitions of the 5 runs, the number of instance features was considerably reduced, having an averaged number of 65, 31 and 36 features respectively for subjects K3b, L1b and K6b. Concerning TesSet partitions, it is important to bear in mind that there was no need to apply the feature selection to them. Instead, the same features of the training would be selected for the testing partitions. This was done because the testing process is done to assess an algorithm already trained using some specific features and, subsequently, the instances used for both purposes must have the same features.

Overview of the initial steps

As the team proceed many steps until reaching the point that training and testing partitions were ready, it has been considered useful to summarise all these steps with a fast overview of what done in Section 6.3. For the sake of

an easier comprehension, the whole process has been graphically represented in Figure 6.2.

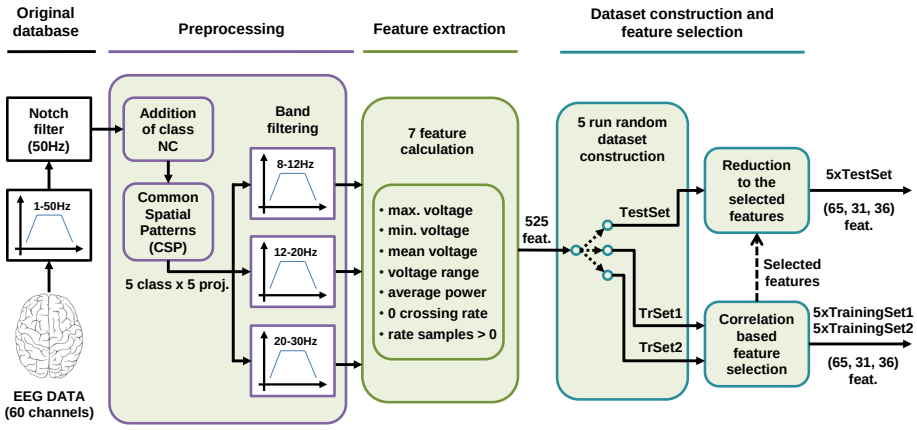

Figure 6.2: Diagram of the whole construction process of the new datasets.

First of all, it is important to remember that the original EEG signals were taken from dataset IIIa of BCI Competition III [149]. Those 60 EEG signals had been collected with a 250Hz sampling rate and then notch filtered to remove noises coming from the grid, i.e, at 50Hz. Then, the team modified the dataset by using the first 2s of the signals as belonging to NC control states. Later, the instances were taken through CSP processing, repeating the process once per class (hence, 5 times) to get the most meaningful projections of the signals for each of the classes. After the CSP, the 25 projections were band filtered, getting as result 3 frequency bands for each projection: 8-12Hz, 12-20Hz and 20-30Hz.

Finished the preprocessing, the seven features of Table 6.1 were extracted from each projection's three frequency bands. In the following step, the data coming from the different subjects went 5 times under random partitioning, obtaining three smaller datasets: TrainingSet1 and TrainingSet2 for training the first and second-level classifiers respectively and TestSet for testing the whole system. Finally, to avoid the curse of dimensionality, the best features for training the classifiers were chosen using the correlation-based feature selection method (those same features would be used for the TestSet partitions).

6.4 Previous approaches of the classification

This section will present the two supervised learning-based approaches that the team tried for the classification problem proposed with the newly expanded database. As stated in Section 6.2, for a system to be admitted as usable, the first level NC vs. IC classification's FPR has to remain below 10%. As it will be shown, any of these approaches fail to fulfil this restriction and shows how

solely using supervised learning methods limits the usability of the BCI systems, justifying the use of unsupervised learning proposed in Section 6.5.

This section has been divided into two subsections. First, Subsection 6.4.1 will present the most basic approach which is no other than trying to tackle the problem using a single supervised learning classifier to distinguish between the five classes. Second, Subsection 6.4.2 will give a proposal based on dividing them into two levels: the first for separating NC and IC class instances and the second for determining to which of the four IC classes belonged the instances considered as IC in the first level. Consequently, this two-level hierarchical system will consist of two classifiers, one for each level, coming both from the supervised learning algorithm family.

6.4.1 One-level system based on supervised learning

As an initial approach to solve the motor imagery proposed in the dataset, it made sense to try a single level solution. So far, many different supervised learning algorithms have been used for BCI systems [164, 165]. For this initial single-level approach the team considered interesting to try 11 different algorithms that can be considered to represent the state of the art of the different machine learning paradigms (see Subsection 3.4.1 for more information). The chosen algorithms and their paradigms are listed below:

- **Tree-based:** decision trees (DTs), random forests (RFs).
- **Ensembles of classifiers:** 10 decision tree combinations of Ada boost (AdaB) and bagging (Bag) algorithms.
- **Function-based:** support vector machines (SVMs) and radial basis function (RBF) network.
- **Distance-based:** 1-NN and 5-NN versions of k-nearest neighbours.
- **Rule-based:** 1R rule.
- **Probabilistic:** naïve Bayes (NB) and logistic regression (LR).

Initially, the team tried to optimise the parameters of each of the classifiers to get the best possible result. However, after some trials, the team saw that optimising the classifier towards improving the accuracy for one class resulted in worsening the performance for the other classes. Thus, in the end, Weka platform's default structure and parameter settings were used for all the classifiers. Being a single level system, the classifiers had to differentiate between the 5 classes, i.e., 1 NC class and the 4 different IC classes (left hand, right hand, tongue and feet).

For this experimentation, the team decided to compare the performance of the classifier using part of the training datasets and, then, only taking the best performing onto the testing phase (with TestSet). The reason not to use the whole TrainingSet1 and TrainingSet2 datasets was to avoid biasing the classifiers

towards any of the classes. Therefore, those datasets were reduced to 300 cases for subject K3b (60 instances of each class) and to 200 cases for the other two subjects (40 instances of each class). The average accuracy and FPR values (with the standard deviation) were the selected performance indicators for these comparisons. These indicators were obtained over 5 runs, and for the training, using the 10-fold CV methodology. The results obtained from the comparison of the 11 classifiers are the ones presented in Table 6.2.

Table 6.2: Accuracies and FPRs of the classifiers of the one-level system over 5-runs, with 10-fold CV for the training.

		Estimation of the best classifier										**Test**
		1R	DT	1-NN	5-NN	NB	RBF	SVM	LR	AdaB	Bag	RF	SVM
K3b	Accuracy (%)	39.1	64.8	72.2	75.6	75.7	77.0	81.6*	67.7	75.8	71.9	73.0	77.5±3.7
	FPR (%)	75.0	52.0	48.3	48.7	35.7	33.0	31.6	42.3	37.3	44.0	47.0	29.2±4.9
L1b	Accuracy (%)	45.8	62.6	62.8	70.3	70.3	70.5	73.5*	60.3	69.3	68.7	68.4	74.4±3.2
	FPR (%)	79.0	50.5	48.5	41.5	45.0	37.0	34.0	42.5	42.0	41.5	45.5	27.5±7.2
K6b	Accuracy (%)	35.5	47.7	49.9	52.8	51.3	57.1	57.3*	48.5	56.2	56.1	55.4	55.6±3.9
	FPR (%)	55.0	55.0	51.0	55.0	45.0	41.0	44.5	53.0	41.5	38.5	39.0	36.2±10.0
Aver.	Accuracy (%)	40.0	59.3	63.1	67.6	67.2	69.5	72.3*	60.1	68.3	66.5	66.6	70.4±3.6
	FPR (%)	70.4	52.4	49.1	48.4	41.0	36.4	36.0	45.4	39.9	41.7	44.3	30.7±7.0

As shown in Table 6.2, SVMs were the best performing classifiers of the training. However, the team wanted to check if the performance differences were statistically significant before choosing them as the best classifier. Consequently, the results were compared employing a 95% significance Wilcoxon non-parametric test (explained in Subsection 3.4.5). The results from these test proved SVMs to be significantly better, obtaining p-values below 0.02. Thus, the SVMs were finally tested using TestSet (see the last column of Table 6.2, obtaining an average accuracy of 70.4% using the three subjects' testing datasets. Despite it seems a relatively good result, the average FPR was 30.7%. This FPR is far above the maximum value of 10% and so, the practical use of such a system would not be viable.

6.4.2 Two-level system based on supervised learning

Once the team saw that single level solutions using only supervised learning strategies could not score an FPR lower than 10%, they decided to try to separate the whole problem and divide it into two tasks. Whereas the first task would correspond to the initial NC-IC class separation, the second task would be determining the specific class of the instances previously classified as IC.

For the first classification level, the team decided to use the same 11 classifiers used in the previous Subsection 6.4.1. In this case, it was only TrainingSet1 that was used to train the first level system and TrainingSet2 was kept for training the second level classifier. The results obtained from the training of the first level

have been presented in Table 6.3. These results show how although the FPRs values had improved in general, they were still above the allowable maximum of 10%: the best average results over the 5-run 10-fold CV process was obtained by the 5-NN algorithm (82.4% accuracy and 16.4% FPR).

Table 6.3: Estimated performances over 5-runs with 10-fold CV for the first level classifiers of the supervised learning two-level system: accuracy and FPR.

		1R	DT	1-NN	5-NN	NB	RBF	SVM	LR	AdaB	Bag	RF
K3b	Accuracy (%)	66.9	71.9	79.6	84.0*	78.3	80.2	81.8	73.7	79.7	78.7	78.4
	FPR (%)	34.0	28.5	22.5	12.8	26.2	19.5	17.5	25.7	21.3	20.3	24.8
L1b	Accuracy (%)	71.4	75.5	79.9	82.5	77.1	79.9	81.1	80.5	81.3	80.4	83.1*
	FPR (%)	30.8	24.3	18.8	17.0	38.8	31.5	25.5	18.8	19.5	22.3	22.8
K6b	Accuracy (%)	76.6	75.5	76.9	80.0	79.4	81.5	82.1	76.5	81.0	82.4*	81.8
	FPR (%)	20.3	24.8	25.5	21.0	22.5	15.3	18.3	24.5	18.0	16.8	20.3
Aver.	Accuracy (%)	71.0	74.0	78.9	82.4*	78.3	80.5	81.7	76.4	80.5	80.2	80.7
	FPR (%)	29.1	26.2	22.3	16.4	28.7	21.7	20.0	23.4	19.9	19.9	22.9

Despite the team had already seen that the FPR values obtained in the first level were unacceptable, they still wanted to see how good would the second level classifier perform using those 11 algorithms. Therefore, those classifiers were trained using TrainingSet2, getting the results presented in Table 6.4. During this process, the SVM proved to be the best algorithm for determining the specific class of IC instances (left hand, right hand, tongue and feet), scoring an accuracy of 78.2%.

Table 6.4: Estimated accuracies over 5-runs with 10-fold CV for the second level classifiers of the supervised learning two-level system.

Accuracy (%)	1R	DT	1-NN	5-NN	NB	RBF	SVM	LR	AdaB	Bag	RF
K3b	46.5	69.0	79.2	83.5	81.2	81.2	88.3*	69.7	79.2	75.8	81.5
L1b	54.8	68.5	66.3	70.3	74.5	71.3	80.8*	69.8	75.5	73.3	73.5
K6b	35.8	57.5	51.0	57.5	54.5	55.5	60.3*	49.8	59.0	57.8	56.5
Average	45.8	65.6	67.4	72.3	71.7	71.0	78.2*	64.0	72.4	69.9	72.1

Finally, to be consistent with the methodology, the classifier results of both levels were analysed using the 95% significance Wilcoxon non-parametric test. On the one hand, the test showed that the results of the 5-NN algorithm of the first level were not significantly different from a statistic perspective. On the other hand, the test concluded that the results of the SVM were significantly better compared to the other classifiers' (scoring p-values below 0.02).

Consequently, after the whole analytic process, the team concluded that it was not possible to have FPR values below 10% using Weka's default values. Anyway, the team decided to fine-tune the 5-NN classifier of the first level to

see if it was possible to obtain permissible FPRs. When they adjusted the thresholds to determine if an instance corresponded to the IC class, they were able to take the FPR below 10%. However, the overall accuracy of the system decreased considerably. Thus, the team opted for trying the new approach of using the K-means unsupervised learning clustering algorithm for the first level classification. This way, as it will be presented in the next Section 6.5, the team could define the minimum proportion of IC belonging instances inside a cluster for labelling the whole cluster as IC, obtaining acceptable FPRs with a good overall performance accuracy.

6.5 Two-level hierarchical classification system using unsupervised learning

As done in Subsection 6.4.2, for the proposal of this contribution the team decided to divide the problem into two levels. However, in the proposal of this section, the team decided to attempt the first level classification using unsupervised learning clustering techniques while maintaining supervised learning techniques for the second level to determine whether the instances labelled as IC corresponded to moving either the left or right hands, the feet or the tongue. The proposal has been graphically summarized in Figure 6.3.

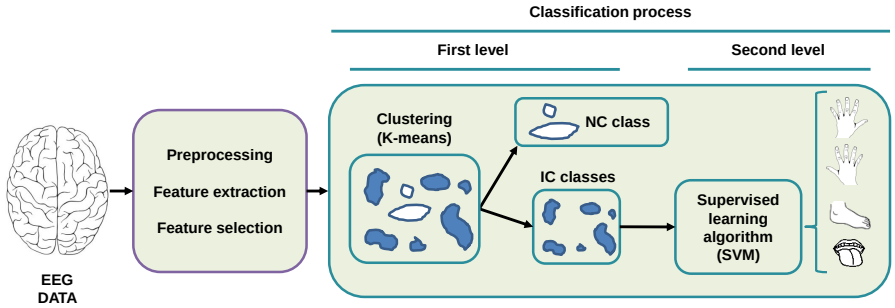

Figure 6.3: Structural diagram of the proposal: it uses clustering and supervised learning algorithms for the first and second levels respectively.

6.5.1 First level: detection of movement intentionality

Not having obtained an acceptable FPR without affecting the overall system performance, the team considered interesting to vary the strategy and to use an unsupervised learning algorithm for the first level classification (IC vs. NC class). Concretely, the algorithm selected for this task was K-means, which again, was applied using Weka's default settings. However, the team decided to

tackle the problem as done in [166], setting a minimum number of IC instances in a cluster as the threshold to exceed for considering the whole cluster as belonging to the IC class (called the IC-threshold). This way it is possible to get FPR scores below 10%, as setting threshold conservatively to avoids undesired end-effector actuation produced by false positive detections.

After having chosen the new approach to the problem, the team tried different K parameter settings for the K-means algorithm and estimated the performance of the different variations using TrainingSet1 with a 5-run 10-fold CV methodology. These trials were done for 12 different K parameter values, starting at K=5 and increasing the value in increments of 5 until K=60. Moreover, the performance estimations of the algorithm variations were also calculated with different IC-thresholds for considering a whole cluster as being labelled as IC. The results of this experimentation can be found in Table 6.5 of page 134.

Looking at Table 6.5 it is possible to see how the algorithm performed with the datasets of each subject averaged from the 5-run 10-fold CV experimentation. Table 6.5 gives the average accuracy and FPR values for all the K parameter and IC-threshold settings. From all, the best performing setting is K=35 combined with an IC-threshold of 80%, which scored 73.5% of accuracy and 7.4% of FPR. It has been considered that setting IC-threshold to 80% is the best option as it produces the highest accuracies among all the settings that have FPRs of less than 10%. Having defined IC-threshold as 80%, the team decided that K=35 was the best setting as, using the three subjects' results, it is the one getting the best average accuracy.

Finally, as it was also done with the two systems presented in Section 6.4, the significance of the results was statistically analysed with the 95% significance Wilcoxon test. The comparative analysis showed that setting IC-threshold to 80% produced significant differences with the other IC-threshold values. On the other side, concerning the K parameter, the test concluded that setting K inside the [20,40] range did not produce significant differences in the results. Anyway, the settings inside this range were proven to give statistically significant differences compared to the values of K out of this range (p-values$<$0.02).

6.5.2 Second level: intentional movement classification

Once the classifier of the first level had been tuned up it was time to select a classifier for the second level classification. Seeing the experimentation of Subsection 6.4.2 where the SVM obtained the best performance classifying the IC instances (their average accuracy was 78.2%, see Table 6.4), the team opted for directly using again this type of algorithm for the second level.

Therefore, as a summary, having chosen to use SVMs for the second-level classifier, the proposed two-level hierarchical classification BCI system consisted in combining a thresholded K-means algorithm for the first level and an SVM algorithm for the second level classification. By doing this, the FPR of the system can be kept under 10% while at the same time maintaining the second level IC classification performance. The following Section 6.6 will present evidence of this, presenting the results of the system during the test as well as comparing

them with other research focused on the same problem.

Table 6.5: Estimation of the K value and IC threshold: accuracy and FPR depending on the value of the K parameter and the IC threshold.

IC-threshold	5-run / 10-CV		K value											
			5	10	15	20	25	30	35	40	45	50	55	60
90%	K3b	Accuracy (%)	52.0	62.6	65.8	68.2	67.5	67.1	68.3	68.5	71.1	72.3*	71.2	70.7
		FPR (%)	1.0	2.5	2.8	3.7	4.0	4.3	4.2	3.8	4.7	5.3	6.8	6.7
	L1b	Accuracy (%)	51.1	57.0	59.5	64.8	64.8	64.8	67.1	68.8*	67.9	68.4	67.8	68.2
		FPR (%)	2.0	3.5	3.5	5.0	5.8	5.8	5.5	6.3	7.8	8.0	9.3	9.8
	K6b	Accuracy (%)	53.0	63.9	64.5	66.3	65.3	67.6	68.6	68.6	69.1	69.4*	69.1	69.2
		FPR (%)	1.0	2.8	6.0	6.5	5.3	7.0	6.8	8.5	9.3	8.8	9.3	10.8
	Aver.	Accuracy (%)	52.0	61.4	63.6	66.6	66.1	66.6	68.1	68.6	69.6	70.3*	69.7	69.5
		FPR (%)	1.3	2.9	3.9	4.9	4.9	5.5	5.3	5.9	6.9	7.1	8.2	8.8
80%	K3b	Accuracy (%)	58.9	70.8	71.9	73.2	74.7	74.3	75.8	76.3	76.3	76.6*	74.7	71.9
		FPR (%)	6.5	7.5	6.0	6.7	7.3	7.7	7.2	6.7	6.5	7.7	8.0	7.8
	L1b	Accuracy (%)	54.3	62.8	64.8	69.9	71.9	70.1	71.4	71.5*	69.6	70.3	72.0	72.8
		FPR (%)	3.0	7.8	6.8	8.8	7.3	7.8	6.8	8.5	9.5	9.8	10.0	10.5
	K6b	Accuracy (%)	65.5	71.5	71.1	73.3	71.6	72.6*	72.3	70.6	71.0	71.1	72.5	71.8
		FPR (%)	6.5	5.8	8.8	9.0	8.8	8.3	8.3	9.5	11.0	10.0	10.0	11.0
	Aver.	Accuracy (%)	59.5	68.7	69.6	72.3	73.0	72.6	73.5*	73.3	72.6	73.2	73.3	72.1
		FPR (%)	5.5	7.1	7.0	7.9	7.7	7.9	7.4	8.0	8.6	8.9	9.1	9.5
70%	K3b	Accuracy (%)	70.8	74.9	76.4	76.9	76.4	76.7	76.9	79.7*	78.3	78.1	77.2	77.1
		FPR (%)	17.5	13.5	8.5	9.0	10.8	12.0	11.8	9.8	9.8	10.7	12.3	11.5
	L1b	Accuracy (%)	61.3	70.1	73.9	74.8	75.8	74.0	75.1	75.6	73.8	73.5	74.6	76.1
		FPR (%)	14.8	18.3	15.0	13.0	12.8	13.8	14.3	15.0	14.8	15.5	15.5	13.5
	K6b	Accuracy (%)	75.4	76.8	75.3	75.4	74.6	75.8	74.9	74.5	74.1	73.6	74.4	74.0
		FPR (%)	15.5	10.5	13.5	13.5	13.0	13.5	12.8	14.0	15.5	16.3	16.3	16.3
	Aver.	Accuracy (%)	69.4	74.1	75.4	75.9	75.7	75.6	75.8	77.0	75.8	75.5	75.7	75.9
		FPR (%)	16.1	14.0	11.8	11.4	12.0	12.9	12.8	12.5	12.9	13.6	14.4	13.5
60%	K3b	Accuracy (%)	73.8	76.9	78.8	78.7	77.8	77.5	77.5	80.7	79.4	79.0	78.3	78.4
		FPR (%)	21.7	15.5	14.5	11.7	14.0	15.5	15.7	13.3	13.3	14.0	16.0	15.2
	L1b	Accuracy (%)	68.8	73.6	76.5	77.0	76.8	74.9	76.9	77.0	74.6	74.6	73.6	77.0
		FPR (%)	34.8	24.8	21.3	16.5	18.0	20.0	19.0	19.0	20.5	21.0	20.0	19.0
	K6b	Accuracy (%)	77.5	77.6	76.3	75.6	75.6	76.8	75.4	75.6	75.6	76.1	76.1	75.0
		FPR (%)	20.0	13.5	15.5	16.8	18.0	17.5	17.3	18.3	18.8	18.3	20.1	20.0
	Aver.	Accuracy (%)	73.4	76.2	77.4	77.3	76.9	76.5	76.7	78.2	77.0	76.9	76.3	77.0
		FPR (%)	24.9	17.6	16.7	14.5	16.3	17.4	17.1	16.4	16.9	17.2	18.2	17.6

Note: FPRs below 10% have been shaded in grey; asterisks "*" indicate the highest accuracy for a subject with an specific IC-threshold and the results of the best performing option have been underlined. The bold box encloses the cases without statistically significant differences in accuracy.

6.6 Experimental results

6.6.1 Testing the hierarchical two level system

Testing the first-level classifier

So far the proposed two-level hierarchical classifying BCI system has been designed. However, as explained in Section 3.4, every machine learning solution has to be tested with data that has remained unused for this purpose. In this case, TestSet dataset was the data partition that was created for the testing, which was built five times for all the subjects for getting reliable results from 5-runs. Hence, first of all, it was the first level classifier that was tested to check that the system obtained good FPR results. The results of this test are presented in Table 6.6, where IC-NC confusion matrix has been given along with the average accuracy and FPR over the 5-runs.

Table 6.6: Confusion matrix, accuracy and FPR of the first level classifier using TestSet.

(K-means, K=35) (IC-threshold 80%)	Confusion Matrix			FPR (%)	Accuracy (%)
K3b		IC-estimated	NC-estimated	7.2±4.1	73.5±2.9
	IC-real	65	55		
	NC-real	9	111		
L1b		IC-estimated	NC-estimated	8.5±3.6	74.3±5.8
	IC-real	46	34		
	NC-real	7	73		
K6b		IC-estimated	NC-estimated	9.5±4.2	70.5±2.9
	IC-real	40	40		
	NC-real	8	72		
Average				**8.2±3.9**	**72.9±3.7**

The results of Table 6.6 show that the FPR constraint was successfully satisfied as the FPR for the three subjects' were lower than 10%, being 8.2% the average FPR for them. Whereas the best FPR was obtained for subject K3b (FPR of 7.2%, 9 misclassified NC instances), it was slightly worse for the other two: 8.5% for L1b (7 misclassified instances) and 9.5% for K6b (8 misclassified instances). Concerning the instances correctly estimated as IC, 65, 46 and 40 cases that would be passed onto the second level classifier for subjects K3b, L1b and K3b respectively.

Testing the second-level classifier

After testing the first classification level it was time to do the same for second-level classifier. For that purpose, the SVM algorithm was used to de-

termine the specific class IC class of the correctly classified IC instance. As shown in the confusion matrix of Table 6.7, the four classes between which the algorithm had to differentiate are left hand, right hand, tongue and feet. The average accuracy of the SVMs for each of the subjects over the 5-runs has also been given in Table 6.8, along with its standard deviation from all the runs.

Table 6.7: Classification performance for the second level with the TestSet dataset.

	Confusion Matrix					Accuracy (%)
K3b		Left-estimated	Right-estimated	Tongue-estimated	Foot-estimated	89.2±2.0
	Left-real	11	3	0	0	
	Right-real	2	21	0	0	
	Tongue-real	0	0	12	1	
	Foot-real	0	0	1	14	
L1b		Left-estimated	Right-estimated	Tongue-estimated	Foot-estimated	78.3±1.4
	Left-real	9	1	0	0	
	Right-real	2	12	0	0	
	Tongue-real	1	1	5	3	
	Foot-real	0	0	2	10	
K6b		Left-estimated	Right-estimated	Tongue-estimated	Foot-estimated	57.5±8.5
	Left-real	6	5	0	0	
	Right-real	4	7	0	0	
	Tongue-real	2	3	2	0	
	Foot-real	1	1	1	8	

The values of Table 6.7 show relatively good accuracies for K3b and L1b (89.2% and 78.3% respectively) and worse results for K3b, getting only an accuracy of 57.5%. Concerning the classes, from an overall perspective, the most difficult task for the classifier was to distinguish between the left and right hands. However, in the particular case of K6b, it was the tongue moving class that was the most difficult to classify.

Overall performance analysis

The last step of the test phase was to obtain the general performance of the BCI system, involving in the performance indicator calculations the numbers from both classification levels. Therefore, considering the classification of the 5 classes (1 NC and 4 IC), the team calculated the system's overall performance indicators (FPR and accuracy) of Table 6.8.

Table 6.8: Classification performance for the proposed system with TestSet dataset over 5-runs.

	K3b	**L1b**	**K6b**	**Average**
FPR (%)	7.2±4.1	8.5±3.6	9.5±4.2	**8.2±3.9**
Accuracy (%)	70.4±2.2	68.1±2.0	59.4±3.7	**66.6±2.6**

Analysing the values given in Table 6.8 it can be seen that the average accuracy obtained for K3b and L1b (70.4% and 68.1% respectively) differ considerably from the poorer result obtained for K6b (59.4%). Thus, the average accuracy of the system taking into account all the subjects was 66.6% with an average FPR of 8.2%. Reading in the literature the team found that a similar situation happened in [167]. Further reading [163], the differences in the performance between participants got justified as the system performance is very influenced by the expertise of the subjects using the system. Accordingly, having some previous experience with motor imagery, K3b and L1b were considerably better creating clear motor imagery patterns compared to K6b, who was a beginner in the matter.

6.6.2 Comparison of the results in the literature

Having tested the system the team considered interesting to compare their results against the obtained by other research teams. Although they might not be equal, the work presented in this chapter is very similar to other works found in the literature and so, the team considered that it was possible to establish some comparisons. Table 6.9 gives a summary of the results of the similar works found in the state of the art, giving for each of those research the following information: the number of classes considered in the classification problem, the FPR false positive rate and accuracy scores of their solution and type of algorithm they used for the classification. Concerning the number of classes considered by the works, Table 6.9 considers that IC classes are different to the NC class and so, if the work took into account the NC class then it will be marked as "(+1)" in the table.

Table 6.9: Comparative table of the research found in the literature.

Reference	**Number of classes (+ NC class)**	**FPR (%)**	**Accuracy (%)**	**Algorithm employed**
[163]	4 (+0)	—	74.2	Support Vector Machine
[168]	4 (+0)	—	64.4	Hierarchical Support Vector Machine
[37]	4 (+0)	—	77.6	Multi-class CSP + Fuzzy System
[158]	2 (+1)	10	83.4	Support Vector Machine
[156]	2 (+1)	26.7 / 28.3	73 / 75	ROC Curve Analysis
[157]	1 (+1)	17	72	Support Vector Machine
[161]	3 (+1)	19	84.3	Mahalanobis Linear Distance Classifier
[169]	2 (+1)	21.7	73.5	Correlation-Based
[170]	2 (+1)	1	54	k-Nearest Neighbour + Linear Discriminant Analysis
Our approach	4 (+1)	8.2	66.6	K-means + Support Vector Machine

Looking at Table 6.9 three main blocks can be differentiated. The results of the works that did not consider the NC class occupy the first three lines of the table. As the NC class was not taken into account, these results will also lack the score for the FPR. Then, there is a second block that contains six works that did consider the NC class. Finally, the system proposed in this chapter takes the space of the last block for the sake of making easier the comparison.

Concerning the first block (the ones not considering the NC class), the researchers of [37] applied their solution to the original IIIa dataset of BCI Competition III and they achieved an accuracy of 77.6% combing a fuzzy system with a multi-class CSP algorithm. The work presented in [163] also used this dataset getting an overall accuracy of 74.2%. Finally, a different dataset was used in [168], dataset IIa from BCI Competition IV. To solve the problem of the dataset, which also included 4 IC classes, they used a hierarchical SVM classifier and they obtained an accuracy of 64.4%. Comparing these results to the obtained in the work presented, it can be seen that the accuracy of [168] is worse. Then, the systems of [37] and [163] obtained better accuracies than the system proposed in this chapter. However, none of these works took into account the problem of keeping the FPR low, which affects considerably the performance of the solutions for motor imagery problems.

About the second block, they analysed a two IC lass (left and right hands) motor imagery problem in [156], respectively achieving an accuracy of 73%

and 75% for the left and right-hand movements. Anyway, having such good accuracies, the FPRs of their solution were far above 10%: 26.7% for the left hand and 28.3% for the right hand. Similar results were obtained [169], an accuracy of 73.5% and an FPR of 21.7%, for the hand or foot motor imagery classification problem they chose to analyse. Then, the team of [157] considered a single class problem for detecting the intention of moving the arm in the signals of 6 volunteers. They compared the performance of seven different algorithms, among which SVMs were found to be the best (72% accuracy). However, keeping the FPR low was the biggest problem for them, which was above 17% and in some cases even reached a value of 75%. They stated that these FPR values were a problem as they would trigger actions when the user wanted to remain at rest. Something similar was mentioned in [161], where they got better accuracy results (84.3%)y and 19% for the FPR) but still did not satisfy the restriction for the FPR (they obtained an FPR of 19%).

So far, none of the works referenced so far obtained an acceptable FPR value. However, in [158] they scored an FPR of 10% and an accuracy of 83.4%. In that work, they used an SVM to face a motor imagery problem in which they had to recognise the intentions of moving either hand. Apart from that work, they also achieved a successful FPR (1%) in [170]. In this work their team merged the k-NN algorithm with linear discriminant analysis (LDA), scoring accuracy of 54%. However, in both cases, it is a two IC class problem that is being analysed and so the difficulty of the classification can be considered to be lower compared to the problem chosen for this contribution.

Therefore, the comparative analysis of the results of this systems proves that the proposed hierarchical system could be a feasible solution for multiple class BCI applications, such as controlling a wheelchair, using a prosthetic limb or controlling a pointer, among others [153, 161, 171].

6.6.3 Other results

Finally, the team did some final extra research around the proposed classification system. Thus, closing this section, this thesis will present the analysis of the times required for the implementation of the system as well as some further results that were obtained from slightly varying the classifiers of the system.

Timing results for an on-line implementation

Beyond analysing the performance of the system, the team also considered interesting to check whether it would be feasible to implement this system in an on-line application. Thus, it was important to verify that the system was able to give a classification response in the time between two epochs.

Accordingly, the team calculated the times to train and test the proposed hierarchical system, presented in the following Table 6.10. Regarding the training, Table 6.10 provides the time required for training just an instance and for training the whole model, including preprocessing, feature extraction and selection steps. Besides, Table 6.10 also shows the times needed to classify an

instance during the testing phase. The times given in Table 6.10 correspond to the average time calculated from using all the available data for training and testing the BCI system for the three subjects. These results were obtained using a machine running Windows 7 (64 bit) with an Intel Core i7-3770, 3.4 GHz processor.

Table 6.10: Average training and testing times needed by the system.

Average (K3b / L1b / K6b)	**Training Model**	**Training Instance**	**Test Instance**
CSP	65.006	0.108	< 0.001
Feature extraction	5.246	0.009	0.009
Feature selection	13.337	0.022	-
K-means	1.047	< 0.001	< 0.001
SVM	0.929	< 0.001	< 0.001
Total time (s)	**84.636**	$\mathbf{< 0.140}$	$\mathbf{< 0.012}$

The time values presented in Table 6.10 show that it would be possible to implement this system in a real-life on-line application as only 12ms are needed for classifying a test instance. This time is much lower than what needed in other references of the literature [156], [157] and [161], which use sliding windows that require times between 0.5s and 2s. Apart from that, the total training time is less than 2 minutes, especially if the training of the K-means and SVM are done in parallel.

Further tuning of the system

To finish with this section the team decided to do a last modification of the hierarchical system, adding the NC class to the 4 IC classes to the second level classifier. The intention to do this was to try to improve the overall performance by giving the system the chance to recover from the NC instances incorrectly classified as IC. Additionally, this would allow to be less conservative in when setting the IC-threshold to keep the FPR under 10%.

By doing this it was possible to see that the overall accuracy can be increased by setting sub-optimal K parameter and IC-threshold values. Different tests were done using the 5-runs of TestSet with the modified system, setting IC-threshold to 70% and K to 20, 25 and 35. The performance obtained from these tests are reflected in Table 6.11, where it is shown that the average accuracy of the system increased at the cost of worsening the FPR (especially for K6b, going from 9.5% to 15%). However, the improvement of the accuracy of the system is clear, even increasing by more than 3 points when K=35 and IC-threshold=70%. Consequently, depending on the case, this type of approach might be feasible, specially when an experienced person used the system.

Table 6.11: Classification performance for the system with a 5-class classifier in the second level.

		K3b	**L1b**	**K6b**	**Average**
K=20	FPR (%)	8.5±2.6	8.5±4.2	14.5±7.4	**10.2±4.4**
IC-threshold=70%	Accuracy (%)	73.9±3.4	69.1±3.6	61.1±5.3	**68.9±4.0**
K=25	FPR (%)	9.5±4.5	8.0±3.5	13.5±4.5	**10.2±4.2**
IC-threshold=70%	Accuracy (%)	74.5±4.0	70.3±3.4	60.1±3.4	**69.2±3.7**
K=35	FPR (%)	10.5±5.9	7.5±3.9	15.0±5.4	**10.9±5.2**
IC-threshold=70%	Accuracy (%)	74.9±2.6	70.6±3.6	61.1±3.7	**69.8±3.2**

6.7 Summary

Among the different types of BCI systems, ERD/ERS motor imagery is one of the most studied in the literature. This type of systems base on detecting brain activity potential changes happening in specific frequency bands of EEGs when the user tries to move or thinks of moving a given part of the body. If different types of potential variations are detected, the BCI system can give specific commands to any end effector or interface, permitting user interaction with the environment without needing any muscular activity. Thus, apart from detecting movement intentionality, it is also vital for the system to recognise when the user wants to be inactive and so, avoid false tripping end-effector activity. These type of applications are starting to be used in the area of neural rehabilitation for helping people regain control over their limbs. Nevertheless, the applications of motor imagery are expanding to new horizons, such as the video-game industry, caregiving or domotics.

Along this contribution chapter, this thesis presented a BCI system that differentiates non-intentional (NC) and intentional movement control states (IC), being also able to classify four different types of intentional movement IC patterns: right hand, left hand, tongue and feet. To do so, the system divides the EEG signal classification problem into two levels. The first determines whether the instance entering the system corresponds to the NC or the IC class to subsequently pass those classified as IC to the second-level classifier. Then, the second classifier determines the specific class of the instance among the four possible IC patterns. Concerning the types of algorithm, whereas the first level of classification employs an unsupervised learning clustering algorithm, a thresholded K-means, the second level classifier is a supervised learning support vector machine (SVM).

Regarding the training and testing of the system, the researchers used an already existing dataset that has been often referenced in many works related to BCI systems: BCI Competition III dataset IIIa. This dataset contains 60 EEG signals for three different subjects repeating several motor imagery trials. However, the original data did not include instances of the NC type and so, the team had to expand the dataset by adding new instances that were created from the initials seconds of the EEG signals acquired during the experiments of the

original dataset. After that, the was sequentially taken to pre-processing and feature extraction stages, and later, three partitions were created: two for training (TrainingSet1 and TrainingSet2) and one for testing the system (TestSet). The partitioning process took care of creating class-balanced smaller datasets so that the classifiers would not get biased. Furthermore, the process was repeated 5 times to later apply the 5-run 10-fold cross-validation methodology to the experimentation.

Once several classifier had been estimated with the training partitions, the team chose the combination of the K-means and the SVM classifiers as the best option and tested it using TestSet. The system obtained an overall average accuracy of 66% for the three subjects. Also, concerning the minimisation of the false positive detections of the NC class, the system was able to keep the false positive rate (FPR) below 10% (the maximum acceptable), getting 8.2% as the average FPR for the tree subjects. Therefore, it can be considered that the intentionality control state detection is robust enough for being used in a real-life application out of the laboratory.

Once the system's performance had been tested, the team did a comparison of their results against other studies found in the state of the art. First, they compared the results of the system against other solutions that used 4-class and 5-class classifiers. These systems presented different degrees of accuracy, some of them better and some of them worse than the proposed solution. However, none of the ones with better accuracy could fulfil the restriction of keeping the FPR under 10%. Even more, in spite of its importance, some of those works did not take the problem into account. After that, the proposed solution was compared to some other systems that kept the FPR low, but they either got worse accuracy or they analysed a classification problem with less classes (a simpler and easier problem). Therefore, the comparative analysis reinforced the positive opinion towards the utility of the system.

Finally, some further analysis was done, checking the times for implementation of the system. These times were short enough to be applied in an online solution. Besides, the team did extra tuning of the classifiers to improve the accuracy of the system, taking the FPR values closer to the limit marked by the 10%. This approach might be valid only for those users with experience in motor imagery systems.

Limitations and future lines

The study presented in this section analysed a motor imagery problem that works in synchronously to given cues indicated on a PC screen. Therefore, any possible real-life application would be strongly limited to having a screen always in front of the user. This limitation opens a new study line in which the system could be applied to non-segmented data and, so, work in a self-paced manner as the systems presented in [172] and [173].

Apart from this line, the team also sees interesting applying automatic EEG channel selection techniques as done [174] and [175]. By doing this it would be

possible to reduce even more the dimensionality of the system, resulting in a simpler problem that permits the use of simpler and faster classifiers.

Another important future line for the team focuses on the training and feedback of the BCI system users [176]. As seen in the experimentation, the performance of the system is highly influenced by the expertise of the users. Consequently, creating a software platform to train users on the proposed motor imagery BCI system would be useful for increasing its accuracy and designing other aspects concerning the usability of the solution.

Part IV

Conclusions

Chapter 7

Conclusions

This thesis dissertation presents contributions to improve both psychological state detection strategies from the field of physiological computing and ERD/ERS motor imagery BCI systems using machine learning techniques.

In the field of physiological computing, this thesis has been focused on two main areas. The first area corresponds to the continuous assessment of the mental stress level, studying pre-processing and new feature extraction techniques as well as redefining a fuzzy logic stress level estimation system. The data for this study was collected from volunteers taking part in a dexterity experiment expressly designed to produce stress.

Concerning the second area, this study was focused on the detection of the physiological changes that are related to the opposite psychological state to stress, the physiological responses towards relaxation. In these regards, the study was focused on detecting which are the EDA features that are most indicative for the activations of the parasympathetic nervous system. The physiological signals were collected from two experiments, one designed for studying relaxation inside a longer stressful situation and the other for analysing the relaxation produced by relaxation techniques. Besides, this part of the work also studied the use of different supervised machine learning algorithms for the detection of those physiological responses towards relaxation.

Finally, the third research line presented in this thesis diverged from the physiological computing area to focus on Brain-Computer Interfaces. This contribution focused on applying both unsupervised and supervised learning algorithms to a motor imagery problem that is well-known in the literature, which is getting good accuracy classifying the classes of the intended imaginary movements while keeping the false positive rate (FPR) below 10% for the cases in which no intentional movement is desired.

The contributions related to physiological computing and motor imagery BCI systems prove the importance of using intelligent control and machine learning strategies to create new human-signal processing systems or to improve the performance of the already existing human-machine and human-computer interfaces. The coming sections 7.1 and 7.2 will present the main conclusions

obtained from the studies related these two areas. Later, the main future work lines will be presented in Section 7.3 and, finally, the reference bibliography that supports this thesis will be presented in Section 7.4.

7.1 Physiological computing

7.1.1 Continuous estimation of stress

The analysis of different physiological signals (EDA, ECG and RESP) showed that the activations of the sympathetic nervous system (SNS) can be detected on the changes in the dynamics of physiological signals. Concerning signal processing and feature extraction, the contribution presented a methodology to extract a heart-rate signal robust to artefacts by combining median filtering, wavelet analysis and a window sweeping r-peak detection algorithm. Besides, to prevent the accumulative nature of the EDA from biasing the stress estimation of the later fuzzy logic analysis, the EDA was decomposed into two components: the average value and the increment of the signal, both within the signal analysis window of 20s.

Apart from processing well-known EDA and ECG signals, it was proven that respiration signals also carry this type of information. The information was extracted by applying a frequential correlation analysis less costly than traditional Fourier analysis. However, although the correlation value of the most highly correlated frequency provides this information, the work showed how to increase the difference between the values that are obtained during stressful and non-stressful situations. This was done by multiplying the correlation value of the most highly correlated frequency with the RMS value of the heart-rate variability signal.

Finally, all those features were used to get a continuous stress level estimation using a fuzzy-logic system. Basing on a previous design in the literature, the team designed a new system that used the previously mentioned four features of the physiological signals already mentioned. To asses the influence of these signals, a new intermediate medium-stress (MS) membership function was added to the system for getting softer transitions between highly stressful and non-stressful situations. Also, the inference rules were weighted according to the meaningfulness of the input features related to the degree of activation of the SNS.

The results of the system was tested using the signals collected during the experimental stage, in which participants had to solve a 3D wooden puzzle within 10 minutes. When compared to other systems in the literature, the results of the assessment showed softer transitions between different situations as well as lower reactivity to sudden changes in the physiological signals. Therefore, having fulfilled these two objectives of the work, the results of the comparison seem promising to the team.

7.1.2 Detection of physiological changes towards relaxation

As a continuation of the study done related to stress assessment, the team designed a system to detect eventual physiological changes produced towards relaxation (called RResps, Relaxation Responses). In this contribution, the team used the signal database collected for the previous study related to stress to see the physiological changes that take place in the body when a person relaxes or tries to do so in the middle of a mentally demanding or stressful situation. Also, the team collected the EDA and HR of different volunteers in a second experiment, which was specially targeted to see the effects of different relaxation techniques in the physiology. During both experiments, the team collected the impressions of the participants through interviews and questionnaires, as well as marked the registers of the signals in the instants in which a relevant event took place.

In collaboration with a team of psychologists and neuro-physiologists, the researchers carried out both qualitative and quantitative analyses of the collected EDA signals. During these analyses, the team reached different conclusions. First, as the eccrine sweat glands are only innervated by the SNS, the team concluded that it is possible to detect RResps solely looking at the EDA signal. When the SNS activates it produces the so-called skin-conductance responses (SCRs), which get represented as a sudden burst of sweat in the EDA signal. Consequently, the team concluded that if no SCRs take place, it means that is the SNS is not but the parasympathetic nervous system (PNS), which is the one responsible for the "rest and digest" type of reactions in the organism.

Second, taking into account the answers of the participants to the interviews, the team concluded that it was possible to label the EDA signal within 20s analysis windows using four different labels: NRResp for the absence of RResps in the signal, LRResp for low-intensity RResps, and MRResp and HRResps for respectively indicating medium and high-intensity RResps. Consequently, the team gathered with the psychologists and physiologists and labelled the EDA signals collected from both experiments every 20s. In order not to bias their opinions, they were randomly given each EDA analysis window.

The last conclusion concerns feature extraction from the EDA signal. After thoroughly analysing the EDA signal with the physiologists, the team decided that two EDA features were indicative for detecting RReps. The first feature is the slope of the EDa within the 20 analysis window, which is indicative or relaxation as an increasing EDA denotes that the SNS is active and that no relaxation is taking place. The second feature, which is innovative to the team's knowledge, corresponds to the surface area contained between the linear regression of the EDA and the EDA instant signal, both of them within the 20s analysis window. The way to understand this feature is as follows: when the EDA decreases very linearly without any SCR, then the linear regression and the instant signal will be almost the same producing a negligible area meaningful of an intense and pronounced RResp. On the contrary, if the EDA decreases but it suffers sudden SCRs, then resulting area would be bigger and so the RREsp

less intense.

Finally, the team extracted these two features for all the subjects of the experiments and took those features to classification. They trained and tested 11 different supervised learning algorithms, among which decision trees (DT) showed themselves as the best option for having very good results and being their decision self-explanatory. Later, the most relevant rules of DTs designed expressly for each of the experiments were selected and merged to build a synthesized rule system valid for the two cases and scoring similar or better results than the experiment specific DTs (($F1_{SYNTHESIZED}$=0.994 vs $F1_{DT\text{-}EXP1}$=0.990 and $F1_{DT\text{-}EXP2}$=0.992).

In summary, in view that the system obtained good performance results using solely two features extracted from the EDA, the team believes that the results obtained during this work are satisfying.

7.2 Motor imagery BCI systems

During the period in which this thesis focused on BCI systems, the research team focused on solving the ERD/ERS motor imagery problem presented in BCI Competition III dataset IIIa. The problem consisted in correctly classifying 4 classes of intentional movement control states (IC) having 60 EEG channels: moving the right hand, left hand, tongue and feet. Nevertheless, the original dataset did not cover the control states in which no intentional movement is thought about (NC states). Therefore, they built an expanded dataset including the initial seconds of the motor imagery signal collection trials as belonging to the NC class. The expansion of the database added an extra level of difficulty to the classification task, as an extra constraint was added to the problem along with the addition of the NC class instances: the false positive ratio of the NC class instances had to be kept under 10% to avoid undesired actuator activity that would lead to user frustration.

After expanding the database, the team applied the CSP methodology to get the most relevant 5 projections for each of the 5 classes (1 NC, 4 IC) out of the 60 EEG channels. Then, the chosen projections were band filtered into three bands and subsequently, the three bands of each projections were used to extract 7 features. After that, the whole preprocessed dataset was partitioned over 5 runs to get three different smaller data subsets: two for training classifiers and one for testing the system estimated to be the best during the training. Finally, for the training partitions of each of the 5-runs, the most meaningful features for each class were selected using the correlation-based feature selection method.

Once all the data subsets had been prepared, the team used them to train different types of machine learning solutions to solve the classification problem. First, they tried solving the 5 class problem using a single classifier based on supervised learning algorithms. Besides, the team also tried to divide the problem and to use a two-level hierarchical classification system based solely on supervised learning methods: the first level was used to differentiate between NC and IC classes in general and the second level was used to differentiate among

the four different IC classes. Although eleven different types of algorithms were tried for these solutions, none of them succeed fulfilling the constraint of keeping the FPR under 10%.

Therefore, seeing the failure of the systems based on supervised learning, the team opted for using a hybrid two-level hierarchical classifier that combined the K-means unsupervised learning algorithm for the initial classification and support vector machines (SVM, a supervised learning algorithm) for the secondary classification inside the IC classes. Moreover, for keeping the FPR low, the team modified the K-means method setting a minimum threshold percentage for the first level classifier. The objective of this threshold was to ensure that all clusters labelled as corresponding to IC type instances at least had a minimum percentage of instances belonging to any of the IC classes. The proposed hybrid hierarchical succeed where the previous two approached had failed: it scored a 66.6% accuracy and an 8.2% FPR on average for the three subjects' signals included in the dataset.

In light of these promising results, the performance of the system was compared to other references in the literature facing the same problem. This comparison showed that proposed system performed better than the others in most of the cases, and, when it did not, it was because the other systems were facing a simpler problem with less classes or not taking NC states into account. Furthermore, the times required to train and test the system were also tested, proving that they are small enough to permit an online implementation of the system for a real-life motor imagery application.

7.3 Future lines

The works presented in the contribution chapters has left the gates opened to further work to be done in each of the three subjects.

In the context of the stress estimation system, the team left a gap in the R-peak detection algorithm as they did not define how to initialise it. Therefore, this had to be done manually, which is far from being useful in a real online implementation. This issue has already been solved by the GICI research team in [99]. However, the team continued improving the R peak detection algorithm making it more robust to noises as well as very computationally efficient [177].

Apart from that, the team saw a weakness in the fuzzy-logic system, which is requiring knowledge about the normal physiological parameters of the user. This is an important flaw of the system as it makes it requires the user to learn to program the solution. Therefore, it would be interesting to create automatically adapting templates using some type of machine learning method, as the use of neural network proposed in [86].

Concerning the line related to the detection the physiological patterns of relaxation, it is important remark that the study was limited to population age and so, it would be interesting to expand the study widening the studied population. It is important to bear in mind that the studies focusing on the automatic detection of the physiological variations indicative for relaxation

are practically non-existent. Therefore, expanding the study to a much bigger population would be useful to corroborate what concluded in this work.

Besides, the team finds interesting to further tune the detection system to make it capable of differentiating between those relaxation responses that take place in hostile situations and those that happen in calm situations. Also, it would be interesting to implement the proposal in a hardware platform because it should be relatively easy to do it as it only uses single physiological variable. Note that, in the end, the final objective of designing these types of solutions is that they can be used in the real-life.

With regard to the two lines commented so far, the author of this thesis would like to make mention a future line that concerns both of them. This future lines is related to the difficulties underlying signal labelling. The sense of stress and relaxation is a very subjective matter and despite the users can give certain indications about it, these are collected via interviews and questionnaires, which give a rather general impression instead of detailed information. Therefore, the author considers that it would be very interesting to use the proposed systems during new data collection experiments. If this was done, then it would be possible to get automatic marks of the most relevant events happening in the physiology of the participants during the experimental stage, and so, making it possible for the researchers to ask the participants about these events to get a better labelling. This future line would be interesting for all researchers that work towards detecting emotions, feelings and other human phenomena that are subjective and difficult to quantise.

Finally, some future lines also opened from the line analysing the motor imagery classification problem. First of all, the studied system was based on collecting EEG patterns time locked to specific indications on a PC screen. Requiring to have a PC screen in front of the user can be a serious limitation, because it might create a sensation on the user as he/she was marked and that catches too much the attention of the people around. Therefore, it would be very important for these systems to work asynchronously, not requiring external stimuli. This could be done by approaching the problem using sliding analysis windows, as proposed in [172] and [173].

The last future line derived from this contribution is related to building training platform so that users can practise using motor imagery systems. As it was shown, the performance of BCI systems is very influenced by how experienced the user is. Therefore, such a platform would be useful to get better accuracy and FPR values while at the same time making the training less tiring. Accordingly, the team started developing a car fetching pc game. The intention of the team is to integrate the proposed BCI system to the game and this way build a motor imagery training system.

7.4 Related publications

This thesis dissertation will finish with the presentation of the different journal and conference articles that were published during this period of thesis concerning the presented three main contributions. The publications are presented below, listed according to their category.

International journals:

- A. Salazar-Ramirez, E. Irigoyen, R. Martinez, and U. Zalabarria, "An enhanced fuzzy algorithm based on advanced signal processing for identification of stress," *Neurocomputing*, vol. 271, pp. 48 – 57, 2018.
- R. Martinez, A. Salazar-Ramirez, A. Arruti, E. Irigoyen, J. I. Martin, and J. Muguerza, "A self-paced relaxation response detection system based on galvanic skin response analysis," *IEEE Access*, vol. 7, pp. 43730–43741, 2019.
- A. Salazar-Ramirez, J. I. Martin, R. Martinez, A. Arruti, J. Muguerza, and B. Sierra, "A hierarchical architecture for recognising intentionality in mental tasks on a brain-computer interface," *PLOS ONE*, vol. 14, pp. 1–18, 06 2019.

International conferences:

- A. Salazar-Ramirez, E. Irigoyen, and R. Martinez, "Enhancements for a robust fuzzy detection of stress," in *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*, pp. 229–238, Springer, 2014.
- U. Zalabarria, E. Irigoyen, R. Martínez, and A. Salazar-Ramirez, "Detection of stress level and phases by advanced physiological signal processing based on fuzzy logic," in *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16* (M. Graña, J. M. López-Guede, O. Etxaniz, Á. Herrero, H. Quintián, and E. Corchado, eds.), (Cham), pp. 301–312, Springer International Publishing, 2017.
- A. Salazar-Ramirez., R. Martinez., A. Arruti., E. Irigoyen., J. I. Martin., and J. Muguerza., "A preliminary system for the automatic detection of emotions based on the autonomic nervous system response," in *Proceedings of the 6th International Congress on Neurotechnology, Electronics and Informatics - Volume 1: NEUROTECHNIX,*, pp. 46–52, INSTICC, SciTePress, 2018.
- M. Perez-Frutos, A. Salazar-Ramirez, M. L. Alvarez, R. Martinez, J. I. Martin, and A. Arruti, "Influence of the environmental hostility level in an hci system," in *Proceedings of the XX International Conference on Human Computer Interaction*, Interacción '19, (New York, NY, USA), Association for Computing Machinery, 2019.

National conferences:

- U. Zalabarria, E. Irigoyen, A. Salazar-Ramirez, and R. Martinez, "Procesamiento robusto para el análisis avanzado de señales electrocardiográficas afectadas por perturbaciones," in *Actas de las XXXVI Jornadas de Automática*, Comité Español de Automática de la IFAC (CEA-IFAC), Sept. 2015.

Bibliography

[1] D. C. Irimia, R. Ortner, M. S. Poboroniuc, B. E. Ignat, and C. Guger, "High classification accuracy of a motor imagery based brain-computer interface for stroke rehabilitation training," *Frontiers in Robotics and AI*, vol. 5, no. 130, 2018.

[2] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*. New York, NY, USA: John Wiley & Sons, 2001.

[3] World Health Organization, "Preamble to the constitution of the world health organization," in *Constitution of the World Health Organization* (W. H. Organization, ed.), pp. 1–2, World Health Organization, 1948.

[4] OpenStax, *Anatomy & Physiology*. OpenStax CNX, Feb. 2016. Accessed: 25.09.2019. `https://openstax.org/books/anatomy-and-physiology/pages/1-introduction`.

[5] J. T. Cacioppo, L. G. Tassinary, and G. G. Berntson, *Handbook of Psychophysiology*. Cambridge Univ. Press, third ed., 2007.

[6] W. B. Cannon, *Bodily changes in pain, hunger, fear, and rage: An account of recent researches into the function of emotional excitement*. D. Appleton & Company, 1915.

[7] Harvard Medical School, "Understanding the stress response: Chronic activation of this survival mechanism impairs health." Harvard Health Publishing, 2018. Accessed: 26.09.2019. `https://www.health.harvard.edu/staying-healthy/understanding-the-stress-response`.

[8] L. Lanteaume, S. Khalfa, J. Régis, P. Marquis, P. Chauvel, and F. Bartolomei, "Emotion induction after direct intracerebral stimulations of human amygdala," *Cerebral Cortex*, vol. 17, pp. 1307–1313, 07 2006.

[9] H. Gray and W. H. Lewis, *Anatomy of the Human Body*. Lea & Febiger, Philadelphia, 1918.

[10] B. Cornell, "Blood composition," 2018. Accessed: 30.09.2019. `https://ib.bioninja.com.au/standard-level/topic-6-human-physiology/62-the-blood-system/blood-composition.html`.

[11] Blausen.com staff [Internet], "Medical gallery of blausen medical 2014: Thoracic cavity." WikiJournal of Medicine 1(2): 10., 2014. Accessed: 26.09.2019. https://en.wikiversity.org/wiki/WikiJournal_of_Medicine/Medical_gallery_of_Blausen_Medical_2014.

[12] M. Cadogan, "Ecg lead positioning." Life in the fast lane [Blog], 2019. Accessed: 12.11.2019. https://litfl.com/ecg-lead-positioning/.

[13] S. C. Segerstrom and L. S. Nes, "Heart rate variability reflects self-regulatory strength, effort, and fatigue," *Psychological Science*, vol. 18, no. 3, pp. 275–281, 2007.

[14] A. L. Hansen, B. H. Johnsen, and J. F. Thayer, "Relationship between heart rate variability and cognitive function during threat of shock," *Anxiety, Stress, & Coping*, vol. 22, no. 1, pp. 77–89, 2009.

[15] M. Malik, "Heart rate variability: Standards of measurement, physiological interpretation, and clinical use," *Circulation*, vol. 93, pp. 1043–1065, 03 1996.

[16] M. A. Woo, W. G. Stevenson, D. K. Moser, R. B. Trelease, and R. M. Harper, "Patterns of beat-to-beat heart rate variability in advanced heart failure," *American Heart Journal*, vol. 123, no. 3, pp. 704–710, 1992.

[17] M. S. Houle and G. E. Billman, "Low-frequency component of the heart rate variability spectrum: a poor marker of sympathetic activity," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 276, no. 1, pp. 215–223, 1999.

[18] R. Edelberg, "Electrical activity of the skin," in *Handbook of psychophysiology* (N. S. G. . R. A. Sternbach, ed.), pp. 367–418, New York: Holt, Rinehart & Winston., 1972.

[19] G. H. Zimny and E. W. Weidenfeller, "Effects of music upon gsr and heart-rate," *The American Journal of Psychology*, vol. 76, no. 2, pp. 311–314, 1963.

[20] A. N. Ishchenko and P. P. Shev'ev, "Automated complex for multiparameter analysis of the galvanic skin response signal," *Biomedical Engineering*, vol. 23, pp. 113–117, May 1989.

[21] A. Greco, G. Valenza, and E. P. Scilingo, *Advances in Electrodermal Activity Processing with Applications for Mental Health*, ch. Feature extraction, pp. 31–33. Springer International Publishing, first ed., 2007.

[22] H. F. Posada-Quintero, J. P. Florian, A. D. Orjuela-Cañón, T. Aljama-Corrales, S. Charleston-Villalobos, and K. H. Chon, "Power spectral density analysis of electrodermal activity for sympathetic function assessment," *Annals of Biomedical Engineering*, vol. 44, pp. 3124–3135, Oct 2016.

[23] A. Greco, G. Valenza, A. Lanata, G. Rota, and E. P. Scilingo, "Electrodermal activity in bipolar patients during affective elicitation," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1865–1873, Nov 2014.

[24] N. Nourbakhsh, F. Chen, Y. Wang, and R. A. Calvo, "Detecting users' cognitive load by galvanic skin response with affective interference," *ACM Trans. Interact. Intell. Syst.*, vol. 7, pp. 12:1–12:20, Sept. 2017.

[25] A. Barreto, J. Zhai, and M. Adjouadi, "Non-intrusive physiological monitoring for automated stress detection in human-computer interaction," in *ICCV-HCI*, Oct. 2007.

[26] A. Morley, L. Hill, and A. G. E. Kaditis, "10-20 system eeg placement," 2016. Accessed: 08.10.2019. `https://www.ers-education.org/lrmedia/2016/pdf/298830.pdf`.

[27] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "An efficient p300-based brain–computer interface for disabled subjects," *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 115–125, 2008. Brain-Computer Interfaces (BCIs).

[28] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter, "Bci competition 2003-data set iib: support vector machines for the p300 speller paradigm," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1073–1076, June 2004.

[29] D. L. Schomer and F. H. Lopes da Silva, *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 6th ed., 2016.

[30] O. Simonova, B. Roth, and J. Stein, "Eeg studies of healthy population—normal rhythms of resting recording," *Acta Universtitatis Carolinae. Medica (Praha)*, vol. 13, no. 7, pp. 543–551, 1967.

[31] G. Pfurtscheller and C. Neuper, "Eeg-based brain-computer interfaces," in *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields* (D. L. Schomer and F. H. Lopes da Silva, eds.), pp. 1227–1236, Philadelphia, Pa.: Lippincott Williams & Wilkins, 2010.

[32] H. Berger, "Über das Elektrenkephalogramm des Menschen. Zweite Mitteilung (2nd report)," *Journal of Psychology and Neurology (Leipzig)*, 1930.

[33] N. S. Kort, P. Cuesta, J. F. Houde, and S. S. Nagarajan, "Bihemispheric network dynamics coordinating vocal feedback control," *Human brain mapping*, vol. 37, pp. 1474–1485, April 2016.

[34] E. M. Whitham, K. J. Pope, S. P. Fitzgibbon, T. Lewis, C. R. Clark, S. Loveless, M. Broberg, A. Wallace, D. DeLosAngeles, P. Lillie, A. Hardy, R. Fronsko, A. Pulbrook, and J. O. Willoughby, "Scalp electrical recording during paralysis: Quantitative evidence that eeg frequencies above 20hz are contaminated by emg," *Clinical Neurophysiology*, vol. 118, no. 8, pp. 1877–1888, 2007.

[35] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Trans Rehabil Eng*, vol. 8, pp. 441–446, Dec. 2000.

[36] D. Bruff, "Chapter 5 eigenvalues and eigenvectors," 2005. [Math 20 - Introduction to Linear Algebra and Multivariable Calculus lecture slides]. Accessed: 07.11.2019. `http://www.math.harvard.edu/archive/20_spring_05/handouts/ch05_notes.pdf`.

[37] T. Nguyen, I. Hettiarachchi, A. Khatami, L. Gordon-Brown, C. P. Lim, and S. Nahavandi, "Classification of multi-class bci data by common spatial pattern and fuzzy system," *IEEE Access*, vol. 6, pp. 27873–27884, 2018.

[38] M. Grosse-Wentrup and M. Buss, "Multiclass common spatial patterns and information theoretic feature extraction," *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 1991–2000, Aug 2008.

[39] J. M. Andujar and A. J. Barragan, "Hybridization of fuzzy systems for modeling and control," *RIAI*, vol. 11, no. 2, pp. 127–141, 2014.

[40] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[41] Y. Bai, H. Zhuang, and D. Wang, eds., *Advanced Fuzzy Logic Technologies in Industrial Applications*. Springer, Sept. 2006.

[42] W. V. Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy Sets and Systems*, vol. 108, no. 2, pp. 159 –178, 1999.

[43] C. Bielza and P. Larrañaga, "Suppervised pattern classification," June 2018. [Advanced Statistics and Data Mining Summeschool (ASDM) lecture slides].

[44] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: Classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, pp. 128–138, June 2017.

[45] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005)*, pp. 529–536, Aug. 2005.

[46] J. Han and M. Kamber, *Data Mining Concepts and Techniques (2nd Edition)*, ch. Classification by Backpropagation, pp. 327–336. Morgan Kaufmann, Jan. 2006.

[47] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, 1988.

[48] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, pp. 18–28, July 1998.

[49] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.

[50] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, January 1967.

[51] T. Bailey and A. K. Jain, "A note on distance-weighted k-nearest neighbor rules," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pp. 311–313, April 1978.

[52] S. Dhanabal and S. Chandramathi, "A review of various k-nearest neighbor query processing techniques," *International Journal of Computers and Applications*, vol. 31, pp. 14–22, Oct. 2011.

[53] J. Han and M. Kamber, *Data Mining Concepts and Techniques (2nd Edition)*, ch. Classification by Decision Tree Induction, pp. 291–310. Morgan Kaufmann, Jan. 2006.

[54] J. Haebichan, "Adaboost for dummies: Breaking down the math (and its equations) into simple terms." Accessed: 02.11.2019. `https://towardsdatascience.com/adaboost-for-dummies-breaking-down-the-math-and-its-equations-into-simple-terms-87f439757dcf`.

[55] K. Teknomo, "Numerical example if k-means clustering," 2017. Accessed: 07.11.2019. `https://people.revoledu.com/kardi/tutorial/kMean/NumericalExample.htm`.

[56] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, (New York, NY, USA), pp. 600–607, ACM, 2002.

[57] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

[58] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[59] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc Int Conf Mach Learn*, pp. 359–366, Morgan Kaufmann, 2000.

[60] J. Kropotov, *Quantitative EEG, Event-Related Potentials and Neurotherapy*. Philadelphia, PA, USA: Academic Press (Elsevier), 1st ed., 2009.

[61] L. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510 – 523, 1988.

[62] B. Z. Allison and J. A. Pineda, "Erps evoked by different matrix sizes: implications for a brain computer interface (bci) system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 110–113, June 2003.

[63] Nazarbayev University, "School of engineering and digital sciences," 2019. Accessed: 11.12.2019. `https://nu.edu.kz/faculty/berdakh-abibullaev`.

[64] X. Xing, Y. Wang, W. Pei, X. Guo, Z. Liu, F. Wang, G. Ming, H. Zhao, Q. Gui, and H. Chen, "A high-speed ssvep-based bci using dry eeg electrodes," *Scientific Reports*, vol. 8, Oct. 2018.

[65] Y. Zhang, P. Xu, Y. Huang, K. Cheng, and D. Yao, "Ssvep response is related to functional brain network topology entrained by the flickering stimulus," *PLOS ONE*, vol. 8, pp. 1–11, 09 2013.

[66] M. Nakanishi, Y. Wang, Y.-T. Wang, and T.-P. Jung, "A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials," *PLOS ONE*, vol. 10, pp. 1–18, 10 2015.

[67] G. Pfurtscheller and F. L. da Silva, "Event-related eeg/meg synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.

[68] E. Thomas, J. Fruitet, and M. Clerc, "Investigating brief motor imagery for an erd/ers based bci," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2929–2932, Aug 2012.

[69] R. Martinez, E. Irigoyen, A. Arruti, J. Martin, and J. Muguerza, "A real-time stress classification system based on arousal analysis of the nervous system by an f-state machine," *Computer Methods and Programs in Biomedicine*, vol. 148, pp. 81–90, Sept. 2017.

[70] R. Picard, "Affective computing," Tech. Rep. 321, M.I.T. Media Laboratory Perceptual Computing, 1995.

[71] D. R. López, A. F. Neto, and T. F. Bastos, "On line recognition of human actions based on patterns of rwe windows applied in dynamic moment invariants," *RIAI*, vol. 11, no. 2, pp. 202–211, 2014.

[72] W. Sato, M. Noguchi, and S. Yoshikawa, "Emotion elicitation effect of films in a japanese sample," *Social Behavior and Personality: an international journal*, vol. 35, no. 7, pp. 863–874, 2007.

[73] W. B. Cannon, "Stresses and strains of homeostasis," *The American Journal of the Medical Sciences*, vol. 189, no. 1, pp. 13–14, 1935.

[74] S. W. Porges, "The polyvagal theory: phylogenetic substrates of a social nervous system," *International Journal of Psychophysiology*, vol. 42, no. 2, pp. 123–146, 2001.

[75] S. Kreibig, "Autonomic nervous system activity in emotion: A review," *Biological psychology*, vol. 84, no. 3, pp. 394–421, 2010.

[76] C.-S. Poon and M. S. Siniaia, "Plasticity of cardiorespiratory neural processing: classification and computational functions," *Respiration physiology*, vol. 122, no. 2, pp. 83–109, 2000.

[77] X. Navarro, "Fisiología del sistema nervioso autónomo," *Revista Neurológica*, vol. 35, pp. 553–562, 2002.

[78] P. Ren, A. Barreto, J. Huang, Y. Gao, F. R. Ortega, and M. Adjouadi, "Off-line and on-line stress detection through processing of the pupil diameter signal," *Annals of biomedical engineering*, vol. 42, no. 1, pp. 162–176, 2014.

[79] K. Subramanya, V. B. Vishnuprasada, and S. Kamath, "A wearable device for monitoring galvanic skin response to accurately predict changes in blood pressure indexes and cardiovascular dynamics," in *INDICON 2013*, pp. 1–4, IEEE, 2013.

[80] J. Healey, R. W. Picard, *et al.*, "Detecting stress during real-world driving tasks using physiological sensors," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 156–166, 2005.

[81] H. Helson and L. Quantius, "Changes in skin temperature following intense stimulation," *Journal of Experimental Psychology*, vol. 17, no. 1, pp. 20–35, 1934.

[82] C. K. Lee, S. Yoo, Y. J. Park, N. H. Kim, K. S. Jeong, and B. Lee, "Using neural network to recognize human emotions from heart rate variability and skin resistance," in *IEEE-EMBS 2005.*, pp. 5523–5525, IEEE, 2006.

[83] G. E. Sakr, I. H. Elhajj, and H. A.-S. Huijer, "Support vector machines to define and detect agitation transition," *Affective Computing, IEEE Transactions on*, vol. 1, no. 2, pp. 98–108, 2010.

[84] G.-J. De Vries, S. Pauws, and M. Biehl, "Insightful stress detection from physiology modalities using learning vector quantization," *Neurocomputing*, vol. 151, pp. 873–882, 03 2015.

[85] N. Sharma and T. Gedeon, "Artificial neural network classification models for stress in reading," in *Neural Information Processing*, pp. 388–395, Springer, 2012.

[86] A. de Santos Sierra, C. S. Ávila, J. G. Casanova, and G. B. D. Pozo, "A stress-detection system based on physiological signals and fuzzy logic," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 10, pp. 4857–4865, 2011.

[87] P. Sasikala and R. Wahidabanu, "Robust r peak and qrs detection in electrocardiogram using wavelet transform," *International Journal of Advanced Computer Science and Applications-IJACSA*, vol. 1, no. 6, pp. 48–53, 2010.

[88] M. Talbi, A. Aouinet, L. Salhi, and A. Cherif, "New method of r-wave detection by continuous wavelet transform," *Signal Processing: An International Journal (SPIJ)*, vol. 5, no. 4, p. 165, 2011.

[89] G. de Lannoy, A. De Decker, and M. Verleysen, "A supervised wavelet transform algorithm for r spike detection in noisy ecgs," in *Biomedical Engineering Systems and Technologies*, pp. 256–264, Springer, 2009.

[90] R. J. Martis, C. Chakraborty, and A. K. Ray, "Wavelet-based machine learning techniques for ecg signal analysis," in *Machine Learning in Healthcare Informatics*, pp. 25–45, Springer, 2014.

[91] A. Ezeiza, N. Garay, K. L. de Ipiña, and A. Soraluze, "Ethical issues on the design of assistive technology for people with mental disabilities," in *ICEHVE*, Mar. 2008.

[92] J. J. Gross and R. W. Levenson, "Emotion elicitation using films," *Cognition & emotion*, vol. 9, no. 1, pp. 87–108, 1995.

[93] CSEA-NIMH, "The international affective picture system: Digitalized photographs," *Center of Research in Psychophysiology*, 1999.

[94] Z. Hong-tu and Y. Jing, "The wavelet decomposition and reconstruction based on the matlab," in *Proc. of the Third Int. Symposium on Electronic Commerce and Security Workshops (ISECS 2010), China*, 2010.

[95] D. Wei, *Coiflet-type wavelets: theory, design and applications*. PhD thesis, The Universtity of Texas, 1998.

[96] A. Salazar-Ramirez, E. Irigoyen, and R. Martinez, "Enhancements for a robust fuzzy detection of stress," in *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*, pp. 229–238, Springer, 2014.

[97] A. Salazar-Ramirez, E. Irigoyen, R. Martinez, and U. Zalabarria, "An enhanced fuzzy algorithm based on advanced signal processing for identification of stress," *Neurocomputing*, vol. 271, pp. 48 – 57, 2018.

[98] P. J. Lang, "Behavioral treatment and bio-behavioral assessment: Computer applications," *Technology in Mental Health and Delivery Systems*, pp. 119–137, 1980.

[99] U. Zalabarria, E. Irigoyen, A. Salazar-Ramirez, and R. Martinez, "Procesamiento robusto para el análisis avanzado de señales electrocardiográficas afectadas por perturbaciones," in *Actas de las XXXVI Jornadas de Automática*, Comité Español de Automática de la IFAC (CEA-IFAC), Sept. 2015.

[100] M. E. P. Seligman and M. Csikszentmihalyi, "Positive psychology: An introduction," *American Psychologist*, vol. 55, no. 1, pp. 5–14, 2000.

[101] H. Cuadra and R. Florenzano, "Subjective well-being: Towards a positive psychology," *Revista de Psicología de la Universidad de Chile*, vol. 12, no. 1, pp. 83–96, 2003.

[102] D. K. Simonton and R. Baumeister, "Positive psychology at the summit," *Review of General Psychology*, vol. 9, pp. 99–102, 06 2005.

[103] P. V. Cappellen, E. L. Rice, L. I. Catalino, and B. L. Fredrickson, "Positive affective processes underlie positive health behaviour change," *Psychology & Health*, vol. 33, no. 1, pp. 77–97, 2018. PMID: 28498722.

[104] R. J. Fehring, "Effects of biofeedback-aided relaxation on the psychological stress symptoms of college students.," *Nursing research*, vol. 32, no. 6, pp. 362–366, 1983.

[105] J. van der Zwan, W. de Vente, A. Huizink, S. Bögels, and E. de Bruin, "The effects of physical activity, mindfulness meditation, or heart rate variability biofeedback on executive functioning, worrying, and mindfulness," *Biological Psychology*, vol. 129, pp. 383 – 384, Oct. 2017.

[106] J. A. P. H. Perera, P. M. C. Perera, L. M. Rathnarajah, and H. B. Ekanayake, "Biofeedback based computational approach for working stress reduction through meditation technique," in *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 132–140, Sep. 2016.

[107] M. Díez-Cirarda, N. Ojeda, J. Peña, A. Cabrera-Zubizarreta, M. n. Gómez-Beldarrain, J. C. Gómez-Esteban, and N. Ibarretxe-Bilbao, "Neuroanatomical correlates of theory of mind deficit in parkinson's disease: A multimodal imaging study," *PLOS ONE*, vol. 10, pp. 1–16, 11 2015.

[108] Instituto Burmuin, "Equipo de profesionales," Oct. 2018.

[109] M. E. P. Seligman, *Florecer. La Nueva Psicología Positiva y la Búsqueda del Bienestar*. Oceano, 2016.

[110] A. Conrad and W. T. Roth, "Muscle relaxation therapy for anxiety disorders: It works but how?," *Journal of Anxiety Disorders*, vol. 21, no. 3, pp. 243 – 264, 2007.

[111] E. Donegan and M. Dugas, "Generalized anxiety disorder: A comparison of symptom change in adults receiving cognitive-behavioral therapy or applied relaxation," *Journal of consulting and clinical psychology*, vol. 80, pp. 490–6, 04 2012.

[112] J. Hoyer, K. Beesdo, A. T. Gloster, J. Runge, M. Höfler, and E. S. Becker, "Worry exposure versus applied relaxation in the treatment of generalized anxiety disorder.," *Psychotherapy and psychosomatics*, vol. 78, pp. 106–115, 2009.

[113] C. Blanco, F. J.Estupiñá, F. J. Labrador, I. Fernández-Arias, M. Bernaldo-de Quirós, and L. Gómez, "El uso de las técnicas de relajación en la práctica de una clínica de psicología," *Anales Psicología*, vol. 30, no. 2, pp. 403–411, 2014.

[114] F. Pagnini, G. M. Manzoni, G. Castelnuovo, and E. Molinari, "The efficacy of relaxation training in treating anxiety," *International Journal of Behavioral Consultation and Therapy*, vol. 5, no. 3-4, pp. 264–269, 2010.

[115] S. E. Stevens, M. T. Hynan, M. Allen, M. M. Braun, and M. R. McCart, "Are complex psychotherapies more effective than biofeedback, progressive muscle relaxation, or both? a meta-analysis.," *Psychological reports*, vol. 100, pp. 303–324, Feb. 2007.

[116] K. S. Carter and R. Carter, "Breath-based meditation: A mechanism to restore the physiological and cognitive reserves for optimal human performance.," *World journal of clinical cases*, vol. 4, pp. 99–102, Apr. 2016.

[117] H. Benson, S. Alexander, and C. L. Feldman, "Decreased premature ventricular contractions through use of the relaxation response in patients with stable ischaemic heart-disease.," *Lancet (London, England)*, vol. 2, pp. 380–382, Aug. 1975.

[118] H. Benson, J. F. Beary, and M. P. Carol, "The relaxation response," *Psychiatry*, vol. 37, no. 1, pp. 37–46, 1974. PMID: 27764631.

[119] J. N. Morey, I. A. Boggero, A. B. Scott, and S. C. Segerstrom, "Current directions in stress and human immune function," *Current Opinion in Psychology*, vol. 5, pp. 13–17, Oct. 2015. Health behavior.

[120] J. J. Kim and D. M. Diamond, "The stressed hippocampus, synaptic plasticity and lost memories.," *Nature reviews. Neuroscience*, vol. 3, pp. 453–462, July 2002.

[121] P. A. Low and E. E. Benarroch, *Clinical Autonomic Disorders*. Lippincott Williams and Wilkins, third ed., 2008.

[122] H. Sandler, U. Fendel, E. Peters, M. Rose, R. Boesel, and B. Klapp, "Subjective experience of relaxation – induced by vibroacoustic stimulation by a body monochord or cd music – a randomised, controlled study in patients with psychosomatic disorders," *Nordic Journal of Music Therapy*, pp. 1–20, 10 2015.

[123] N. Sharma and T. Gedeon, "Modeling a stress signal," *Applied Soft Computing*, vol. 14, pp. 53—61, 01 2014.

[124] K. Palanisamy, M. Murugappan, and S. Yaacob, "Multiple physiological signal-based human stress identification using non-linear classifiers," *Elektronika ir Elektrotechnika*, vol. 19, pp. 80–85, Jan. 2013.

[125] C. Lin, J. King, J. Fan, A. Appaji, and M. Prasad, "The influence of acute stress on brain dynamics during task switching activities," *IEEE Access*, vol. 6, pp. 3249–3255, 2017.

[126] J. Choi and R. Gutierrez-Osuna, "Using heart rate monitors to detect mental stress," in *Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009*, pp. 219–223, 06 2009.

[127] J.-S. Wang, C.-W. Lin, and Y.-T. C. Yang, "A k-nearest-neighbor classifier with heart rate variability feature-based transformation algorithm for driving stress recognition," *Neurocomputing*, vol. 116, pp. 136–143, 2013. Advanced Theory and Methodology in Intelligent Computing.

[128] M. M. Bradley and P. J. Lang, "Measuring emotion: the self-assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.

[129] Y. Z. Tan, S. Ozdemir, A. Temiz, and F. Celik, "The effect of relaxing music on heart rate and heart rate variability during ecg gated-myocardial perfusion scintigraphy," *Complementary Therapies in Clinical Practice*, vol. 21, no. 2, pp. 137–140, 2015.

[130] A. Goshvarpour, A. Abbasi, and A. Goshvarpour, "Evaluating autonomic parameters: the role of sleep duration in emotional responses to music," *Iranian journal of psychiatry*, vol. 11, pp. 59–63, Feb. 2016.

[131] P. J. Lang, M. M. Bradley, and B. N. Cuthbert, "International affective picture system (iaps): Technical manual and affective ratings," *NIMH Center Study Emotion Attention*, pp. 39–57, 1997.

[132] C. A. Gabert-Quillen, E. E. Bartolini, B. T. Abravanel, and C. A. Sanislow, "Ratings for emotion film clips," *Behavior Research Methods*, vol. 47, no. 3, pp. 773–787, 2015.

[133] M. Matsumoto and J. C. Smith, "Progressive muscle relaxation, breathing exercises, and abc relaxation theory," *Journal of Clinical Psychology*, vol. 57, no. 12, pp. 1551–1557, 2001.

[134] Y. Nagai, h. Critchley, E. Featherstone, M. Trimble, and D. R..J., "Activity in ventromedial prefrontal cortex covaries with sympathetic skin conductance level: a physiological account of a "default mode" of brain function," *NeuroImage*, vol. 22, no. 1, pp. 243–251, 2004.

[135] M. Benedek and C. Kaernbach, "Decomposition of skin conductance data by means of nonnegative deconvolution," *Psychophysiology*, vol. 47, pp. 647–658, Mar. 2010.

[136] J. Sugenoya, S. Iwase, T. Mano, and T. Ogawa, "Identification of sudomotor activity in cutaneous sympathetic nerves using sweat expulsion as the effector response," *European Journal of Applied Physiology and Occupational Physiology*, vol. 61, no. 3–4, pp. 302–308, 1990.

[137] M. Merino Monge, I. M. Gomez Gonzalez, A. Cantero, and K. Guzman, "Assessment of biosignals for managing a virtual keyboard," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2012, pp. 331–337, 07 2012.

[138] M. Saidi, H. Hassanpoor, and A. Azizi Lari, "Proposed new signal for real-time stress monitoring: Combination of physiological measures," *AUT Journal of Electrical Engineering*, vol. 49, no. 1, pp. 11–18, 2017.

[139] S. Jerritta, M. Murugappan, R. Nagarajan, and W. Khairunizam, "Physiological signals based human emotion recognition: a review," *Proceedings - 2011 IEEE 7th International Colloquium on Signal Processing and Its Applications, CSPA 2011*, pp. 410–415, Mar. 2011.

[140] K. H. Kim, S. W. Bang, and S. R. Kim, "Emotion recognition system using short-term monitoring of physiological signals," *Medical and Biological Engineering and Computing*, vol. 42, pp. 419–427, May 2004.

[141] D. R. Bach, K. J. Friston, and R. J. Dolan, "Analytic measures for quantification of arousal from spontaneous skin conductance fluctuations," *International Journal of Psychophysiology*, vol. 76, no. 1, pp. 52–55, 2010.

[142] A. R. Subhani, W. Mumtaz, M. N. B. M. Saad, N. Kamel, and A. S. Malik, "Machine learning framework for the detection of mental stress at multiple levels," *IEEE Access*, vol. 5, pp. 13545–13556, 2017.

[143] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, Mar 1986.

[144] J. W. Britton, L. C. Frey, J. Hopp, P. Korb, M. Koubeissi, W. Lievens, E. Pestana-Knight, and E. L. St, *Electroencephalography (EEG): An introductory text and atlas of normal and abnormal findings in adults, children, and infants*. American Epilepsy Society, Chicago, 2016.

[145] I. Choi, I. Rhiu, Y. Lee, M. H. Yun, and C. S. Nam, "A systematic review of hybrid brain-computer interfaces: Taxonomy and usability perspectives," *PloS one*, vol. 12, no. 4, p. e0176674, 2017.

[146] C. Guger, B. Z. Allison, and G. Edlinger, *Brain-Computer Interface Research: A state-of-the-art summary*. Springer, 2014.

[147] J. R. Wolpaw, "The bci endeavor and the mission of this new journal," *Brain-Computer Interfaces*, vol. 1, pp. 2–4, 2014.

[148] J. Abascal, A. Arruti, J. I. Martin, and J. Muguerza, "A hierarchical bci system able to discriminate between non intentional control state and four intentional control activities." Proceedings of the International Conference on Physiological Computing Systems, 2014.

[149] B. Blankertz, K. R. Muller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. D. R. Millan, M. Schroder, and N. Birbaumer, "The bci competition iii: Validating alternative approaches to actual bci problems," *IEEE Trans Neural Syst Rehabil Eng*, vol. 14, pp. 153–159, 2006.

[150] P. J. García-Laencina, G. Rodríguez-Bermudez, and J. Roca-Dorda, "Exploring dimensionality reduction of eeg features in motor imagery task classification," *Expert Syst Appl*, vol. 41, pp. 5285–5295, 2014.

[151] W.-Y. Hsu, "Fuzzy hopfield neural network clustering for single-trial motor imagery eeg classification," *Expert Syst Appl*, vol. 39, pp. 1055–1061, 2012.

[152] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review.," *Sensors (Basel, Switzerland)*, vol. 12, pp. 1211–1279, 2012.

[153] T. Shi, H. Wang, and C. Zhang, "Brain computer interface system based on indoor semi-autonomous navigation and motor imagery for unmanned aerial vehicle control," *Expert Syst Appl*, vol. 42, pp. 4196–4206, 2015.

[154] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proc IEEE*, vol. 89, pp. 1123–1134, July 2001.

[155] L. Cao, B. Xia, O. Maysam, J. Li, H. Xie, and N. Birbaumer, "A synchronous motor imagery based neural physiological paradigm for brain computer interface speller.," *Front Hum Neurosci*, vol. 11, p. 274, 2017.

[156] G. Townsend, B. Graimann, and G. Pfurtscheller, "Continuous EEG classification during motor imagery-simulation of an asynchronous bci," *IEEE Trans Neural Syst Rehabil Eng*, vol. 12, pp. 258–265, June 2004.

[157] D. Planelles, E. Hortal, A. Costa, A. Ubeda, E. Iáez, and J. M. Azorín, "Evaluating classifiers to detect arm movement intention from eeg signals.," *Sensors (Basel, Switzerland)*, vol. 14, pp. 18172–18186, Sept. 2014.

[158] F. Lotte, H. Mouchere, and A. Lecuyer, "Pattern rejection strategies for the design of self-paced EEG-based brain-computer interfaces," in *Proc. 19th IEEE Int Conf Pattern Recognit*, pp. 1–5, Dec. 2008.

[159] R. Leeb, D. Friedman, G. R. Müller-Putz, R. Scherer, M. Slater, and G. Pfurtscheller, "Self-paced (asynchronous) bci control of a wheelchair in virtual environments: a case study with a tetraplegic.," *Comput Intell Neurosci*, p. 79642, 2007.

[160] J. Pan, Y. Li, R. Zhang, Z. Gu, and F. Li, "Discrimination between control and idle states in asynchronous ssvep-based brain switches: A pseudo-key-based approach," *IEEE Trans Neural Syst Rehabil Eng*, vol. 21, pp. 435–443, May 2013.

[161] D. Huang, K. Qian, D.-Y. Fei, W. Jia, X. Chen, and O. Bai, "Electroencephalography (eeg)-based brain-computer interface (bci): a 2-d virtual wheelchair control based on event-related desynchronization/synchronization and state control.," *IEEE Trans Neural Syst Rehabil Eng*, vol. 20, pp. 379–388, May 2012.

[162] R. Scherer, F. Lee, A. Schlogl, R. Leeb, H. Bischof, and G. Pfurtscheller, "Toward self-paced brain-computer communication: navigation through virtual worlds.," *IEEE Trans Biomed Eng*, vol. 55, pp. 675–682, Feb. 2008.

[163] O. AlZoubi, I. Koprinska, and R. A. Calvo, "Classification of brain-computer interface data," in *Proc 7th Australasian Data Mining Conf*, vol. 87 of *AusDM '08*, (Darlinghurst, Australia, Australia), pp. 123–131, Australian Computer Society, Inc., 2008.

[164] H. Bashashati, R. K. Ward, G. E. Birch, and A. Bashashati, "Comparing different classifiers in sensory motor brain computer interfaces.," *PloS one*, vol. 10, p. e0129435, 2015.

[165] A. Celecia, R. Gonzalez, M. Vellasco, and P. Vellasco, "Ensemble of classifiers applied to motor imagery task classification for bci applications." Int Jt Conf Neural Netw, 2017.

[166] A. Bashashati, S. G. Mason, J. F. Borisoff, R. K. Ward, and G. E. Birch, "A comparative study on generating training-data for self-paced brain interfaces," *IEEE Trans Neural Syst Rehabil Eng*, vol. 15, pp. 59–66, Mar. 2007.

[167] F. Y.-T. Lee, R. Scherer, R. Leeb, C. Neuper, H. Bischof, and G. Pfurtscheller, "A comparative analysis of multi-class eeg classification

for brain-computer interface." 10th Computer Vision Winter Workshop (CVWW), 2006.

[168] E. Dong, C. Li, L. Li, S. Du, A. N. Belkacem, and C. Chen, "Classification of multi-class motor imagery with a novel hierarchical svm algorithm for brain-computer interfaces.," *Med Biol Eng Comput*, vol. 55, pp. 1809–1818, Oct. 2017.

[169] S. Aliakbaryhosseinabadi, N. Jiang, A. Vuckovic, R. Lontis, K. Dremstrup, D. Farina, and N. Mrachacz-Kersting, *Replace, Repair, Restore, Relieve-Bridging Clinical and Engineering Solutions in Neurorehabilitation*, vol. 7, ch. Detection of Movement Intention from Movement-Related Cortical Potentials with Different Paradigms., pp. 237–244. Springer, Jan. 2014.

[170] A. Bashashati, R. K. Ward, and G. E. Birch, "Towards development of a 3-state self-paced brain-computer interface.," *Comput Intell Neurosci*, p. 84386, 2007.

[171] F. Nijboer, "Technology transfer of brain-computer interfaces as assistive technology: barriers and opportunities.," *Ann Phys Rehabil Med*, vol. 58, pp. 35–38, Feb. 2015.

[172] Y. Yu, Z. Zhou, E. Yin, J. Jiang, J. Tang, Y. Liu, and D. Hu, "Toward brain-actuated car applications: Self-paced control with a motor imagery-based brain-computer interface," *Comput Biol Med*, vol. 77, pp. 148–155, Oct. 2016.

[173] F. Melinscak, L. Montesano, and J. Minguez, "Asynchronous detection of kinesthetic attention during mobilization of lower limbs using eeg measurements.," *J Neural Eng*, vol. 13, p. 016018, Feb. 2016.

[174] T. Alotaiby, F. E. A. El-Samie, S. A. Alshebeili, and I. Ahmad, "A review of channel selection algorithms for eeg signal processing," *EURASIP J Adv Signal Process*, vol. 2015, 2015.

[175] A. Astigarraga, A. Arruti, J. Muguerza, R. Santana, J. I. Martin, and B. Sierra, "User adapted motor-imaginary brain-computer interface by means of eeg channel selection based on estimation of distributed algorithms," *Math Probl Eng*, vol. 2016, pp. 1–12, 2016.

[176] M. Alimardani, S. Nishio, and H. Ishiguro, "The importance of visual feedback design in bcis; from embodiment to motor imagery learning.," *PloS one*, vol. 11, p. e0161945, 2016.

[177] U. Zalabarria, E. Irigoyen, R. Martinez, and A. Lowe, "Online robust r-peaks detection in noisy electrocardiograms using a novel iterative smart processing algorithm," *Applied Mathematics and Computation*, vol. 369, p. 124839, 2020.