Faculdade de Engenharia da Universidade do Porto

Automatic Detection of Physiological Events in Neurocritical Patients

Sofia Moreira

VERSION 1

Mestrado Integrado em Bioengenharia – Engenharia Biomédica

Supervisor: Professor Maria Celeste Dias (FMUP)

Co-supervisor: Professor Miguel Velhote Correia (FEUP)

June, 2019

Abstract

Described for the first time in 1950 as a “sudden increase of ICP”, plateau waves are complex phenomena that consists in a sudden rise of ICP above 40 mmHg for a minimum period of five minutes. Along with the rise of ICP, there are changes in signals such CPP and HR. Triggered by a vasodilation cascade loop, plateau waves can end naturally or in response to treatment, activating the cerebral vasoconstrictive mechanism to decrease the amount of blood present in the brain, and thus decrease the intracranial pressure.

Plateau waves are very common phenomena observed in acute brain lesion patients. Traumatic brain injury, subarachnoid haemorrhage, acute hydrocephalus and benign intracranial hypertension are some of the reasons that may cause plateau waves and thus an increase in ICP. TBI patients are admitted in Intensive Care Units to be managed under continuous multimodal monitoring to avoid secondary brain injury, namely due to long periods of increased ICP and impaired cerebral autoregulation. In addition, this continuous monitoring is not only important to target management but also to trail. Therefore, an algorithm for automatic detection of plateau waves, that also may help to preview their appearance and eventually avoid them, would be a very helpful clinical tool.

The present project consisted in the development of an algorithm for automatic detection of plateau waves, that demonstrated a very good performance and computational analyses of monitoring data that may generate a pattern that may help clinicians to predict their appearance. The data used was anonymously provided by the *Unidade Cuidados Neurocríticos do Centro Hospitalar S. João* (UCNC) and it was pre-processed. In addition, since machine learning is associated with the development of models that can better describe the provided data based on artificial intelligence, some techniques were applied to assess the existence of a relationship between the EEG signal and the beginning of a plateau waves. Among the set of available techniques, some of them use the label to supervise the learning process, designated supervised learning techniques, which is the case of logistic regression, support vector machine and neural network. On the other hand, the techniques that do not require the label are designated as unsupervised learning techniques and the data is grouped in classes according to the similarities between data points.

Based on the results obtained in the first phase of this project, computed in Weka software, some machine learning techniques were trained. Despite the high performance values they present, most of the results obtained were not satisfactory. However, there is a gaussian process model with a good accuracy, which may indicate the presence of the mentioned relationship between EEG signal and the beginning of plateau waves.

Resumo

Descritas pela primeira vez em 1950 como “aumentos repentinos da pressão intracraniana”, as ondas plateau são fenómenos complexos que consistem num aumento abrupto da pressão intracraniana (PIC) para valores superiores a 40 mmHg por um período mínimo de cinco minutos. Em conjunto com o aumento na PIC, há alterações noutros sinais, nomeadamente na pressão de perfusão cerebral (PPC) e a frequência cardíaca (FC). Acionado pelo ciclo de cascada de vasodilatação, as ondas plateau podem terminar de forma natural ou em resposta a tratamento, ativando o mecanismo de vasoconstrição cerebral para diminuir a quantidade de sangue presente no cérebro e, deste modo, diminuir a PIC.

As ondas plateau são fenómenos muitos comuns observados em pacientes com lesão cerebral aguda. Traumatismo crânio-encefálico (TCE), hemorragia subaracnoide, hidrocefalia aguda e hipertensão intracraniana benigna são algumas das razões que podem causar ondas plateau. Pacientes com TCE são admitidos nas Unidades de Cuidados Intensivos para manuseamento sob monitorização contínua para evitar lesão secundária, nomeadamente por longos períodos de PIC aumentada e ausência de autorregulação cerebral. A monitorização multimodal contínua é importante não só para a orientação clínica do manuseamento do doente, mas também como ferramenta de previsão de prognóstico. Assim, o desenvolvimento de um algoritmo para a deteção automática das ondas plateau, que possa ajudar a prever o seu aparecimento e eventualmente evitá-las, será, sem dúvida um importante instrumento de apoio clínico. uma ferramenta útil.

Nesse sentido, o presente projeto consiste no desenvolvimento de um algoritmo para deteção automática das ondas plateau, que demonstre um bom desempenho e na análise computacional dos dados de monitorização multimodal, de modo a encontrar um padrão que ajude os clínicos a prever o seu aparecimento. Os dados usados foram anonimamente fornecidos pela Unidade de Cuidados Neurocríticos do Centro Hospitalar São João (UCNC) e foi pré-processada. Para além disso, uma vez que *machine learning* está associada ao desenvolvimento de modelos que podem descrever melhor os dados fornecidos baseados em inteligência artificial, algumas técnicas foram aplicadas para avaliar a existência de uma relação entre o sinal de eletroencefalograma (EEG) e o início das ondas plateau. Do conjunto de técnicas disponíveis, algumas usam as etiquetas para supervisionar o processo de aprendizagem, como é o caso da regressão logística, a máquina de vetores de suporte e as redes neuronais. Por outro lado, as técnicas que não requerem as etiquetas são designadas de técnicas de aprendizagem não-supervisionada e os dados são agrupados em classes conforme as similaridades entre os pontos de dados.

Baseados nos resultados obtidos na primeira fase deste projeto, analisados com o programa Weka, foram treinadas algumas técnicas de aprendizagem de máquinas. Apesar dos altos valores de desempenho conseguidos, a maior parte dos resultados obtidos não foram satisfatórios. Contudo, salienta-se um modelo de processamento gaussiano com uma boa precisão o que pode indicar a presença da referida relação entre o sinal de EEG e o início das ondas plateau.

Acknowledgements

First, I would like to acknowledge Professor Maria Celeste Dias, my supervisor, for all the support, availability and supervision demonstrated during the development of the project. The meetings and all the recommendations were important to improve the work. Furthermore, I have to thank all the shared knowledge. Secondly, I would like to acknowledge Professor Miguel Velhote Correia for all the advices and help provided all over the time. I also would like to thank to my parents and closest relatives for all the support provided, the encouragement and the unconditional love in the past five years. Lastly, I want to thanks to all my friends for never letting me give up and the trust placed on me.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

ABP	Arterial Blood Pressure
AMP	Amplitude of Intracranial Pressure
ARD	Automatic Relevance Determination
ARFF	Attribute-Relation File Format
CAR	Cerebral Autoregulation Pressure
CBF	Cerebral Blood Flow
CBFV	Cerebral Blood Flow Velocity
CBFx	Cerebral Blood Flow Reactivity Index
CBV	Cerebral Blood Volume
CPP	Cerebral Perfusion Pressure
CPPopt	Optimal Cerebral Perfusion Pressure
CSF	Cerebrospinal Fluid
CVR	Cerebral Vascular Resistance
$ETCO_2$	End-tidal Carbon Dioxide
FC	*Frequência Cardíaca*
FN	False Negative
FP	False Positive
GCS	Glasgow Coma Score
HMF	Highest Modal Frequency
ICM+	Intensive Care Monitoring
ICP	Intracranial Pressure
LSTM	Long-Short Term Memory
MAP	Mean Arterial Pressure
mmHg	millimetre of mercury
Mx	Index of Autoregulation with Transcranial Doppler
NAR	Nonlinear Autoregressive
NARX	Nonlinear Autoregressive with External Input
NCCU	Neurocritical Care Unit
ORx	Oxygen Reactivity Index
PAx	Pressure Amplitude Index

pbtO_2	Brain Tissue Oxygen Tension
PIC	*Pressão Intracraniana*
PPC	*Pressão de Perfusão Cerebral*
PRx	Pressure Reactivity Index
RAP	Index of Cerebrospinal Compensatory Reserve
RSO_2	Cerebral Haemoglobin Oxygenation Index
SVM	Support Vector Machine
TBI	Traumatic Brain Injury
TCE	*Traumatismo Cranio-Encefálico*
TICR	Time in Critical Region
TN	True Negative
TP	True Positive
UCNC	*Unidade de Cuidados Neurocríticos do Centro Hospitalar S. João*

Chapter 1

Introduction

1.1 - Context

The skull provides support to the brain and protection against falls and accidents. As a closed box that contains the brain, the blood vessels and the cerebrospinal fluid (CSF), an increase in the volume of one of the mentioned components will require changes in the volume of the other to restore the pressure. However some events, such as brain tumour, acute hydrocephalus and benign intracranial, may result in a more permanent increase in the pressure inside the skull, and originate physiological events. Designated as plateau waves or A-waves, these episodes are characterised by the rise in intracranial pressure (ICP) signal to values above 40 mmHg, accomplished with changes in other signals, namely a constant ABP and consequently a decrease in cerebral perfusion pressure (CPP).

These episodes are typically developed in traumatic brain injury (TBI) patients, with low compensatory reserve and good autoregulation. Nevertheless, during the top of ICP wave the autoregulation mechanism becomes impaired and does not hinges the activation of the vasoconstriction mechanism to decrease the quantity of blood in the brain and thus reverse the increased ICP value triggered by the vasodilation cascade loop. This vasoconstrictive mechanism can be activated naturally or in response to active treatment.

Patients with severe TBI are frequently admitted in neurocritical care units to be managed targeted by continuous multimodal monitoring and avoid the development of a secondary brain injury. The *Unidade de Cuidados Neurocríticos do Centro Hospitalar S. João* (UCNC) uses the ICM+ software developed by the Cambridge University, which is a software that not only displays in a screen in real time at bedside all patient signals, but also records the raw data signals collected allowing offline analysis. The online analysis has two possibilities, namely time-domain analysis looking at the time series of the waveforms and trends or frequency-domain analysis looking at the frequency distribution of biosignals and their relationship (Figure 1.1). Furthermore, it incorporates toolboxes of statistical analysis of data already collected from patients.

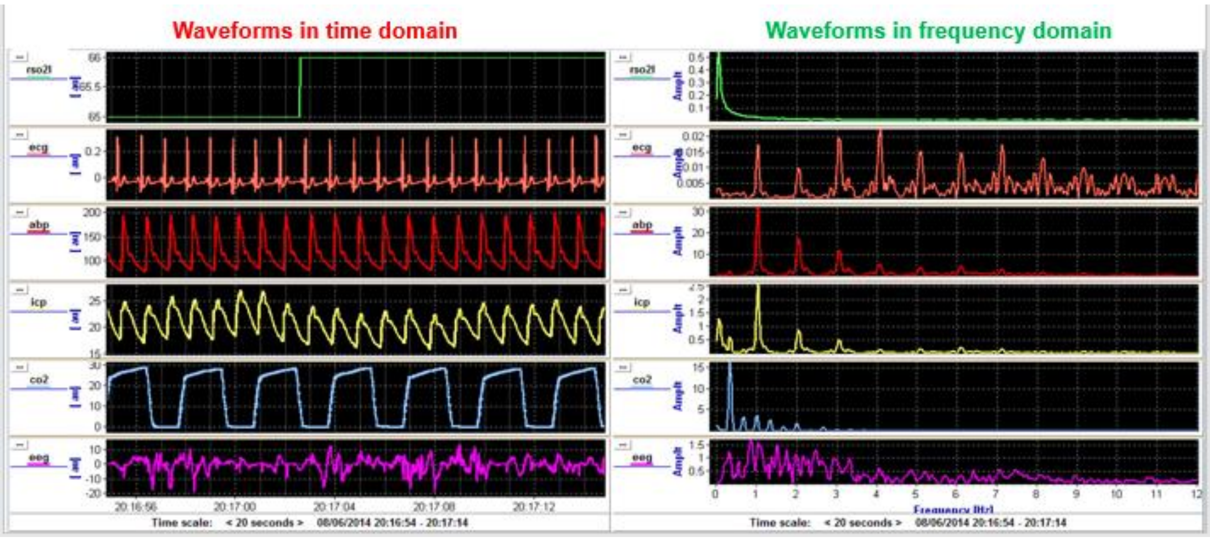

Waveforms in time domain
Waveforms in frequency domain
rso2l
ecg
abp
icp
co2
eeg
Amplt
20:16:56
20:17:00
20:17:04
20:17:08
20:17:12
0
1
2
3
4
5
6
7
8
9
10
11
12
Frequency [Hz]
Time scale: < 20 seconds > 08/06/2014 20:16:54 - 20:17:14
Time scale: < 20 seconds > 08/06/2014 20:16:54 - 20:17:14

Figure 1.1 – Online analysis in ICM+ allows a time-domain analysis (in the left side) and a frequency analysis (in the right side) of the same biosignals.

Regarding with the importance associated with the detection of plateau waves, the need of an algorithm capable to automatically detect these waves arose. For that and based on the type of data collected from patients, the application of machine learning techniques seems to be most appropriate, because the development of the models is based on artificial intelligence. Thus, the first part of this project consisted in the study of several machine learning techniques. They can be divided in supervised learning techniques if the label is used to supervise the learning process, as neural networks and support vector machine (SVM), and in unsupervised learning techniques if the models does not require the labelling of the data. After that, a preliminary evaluation of their performance in the data files, similar to that used in this part of the project, was performed using Weka. Weka is an open source software developed by the Waikato University of New Zealand written in Java. It contains several machine learning techniques, designated as classifiers, than can be applied to data for data mining tasks [1]. The model is not designed, but it provides an overview about the its performance. The data used to train was anonymously provided by UCNC.

The training data in Weka requires a specific format, designated as Attribute-Relation File Format (ARFF) format. However, the data extracted from ICM+ cannot be directly applied in the software, because an ARFF file has a specific structure. At the beginning of the file, the name of the relation must be defined using *@relation*, followed by the definition of the name and the type of each variable declared as *@attribute name type* and, finally, *@data* that determines the beginning of the data.

Neural networks, SVM and logistic regression were the supervised learning techniques trained in Weka software, while random tree was the selected unsupervised techniques. Regarding the preliminary results obtained, presented in Table 1.1, the performance of the models was calculated by correlation coefficient and mean absolute value. While a higher correlation coefficient is associated with a model with better performance, a lower mean absolute error also reveals the effectiveness of the model.

Table 1.1 — Performance values obtained from the methods trained in Weka.

Classifier	Correlation Coefficient	Mean Absolute Error
Neural Networks	0,997	0,0474
Random Tree	0,8811	1,5934
Linear Regression	0,893	1,7556
SVM	0,998	0,0304

As it can be observed, the neural networks and SVM methods had shown the best values in both evaluators. However, the computational time required to neural networks is lower than the one required for SVM.

Although the promising performance values, it is important to note that the data used did not contain the EEG signal, which was always considered to train the models presented in this part of the study. Despite this, the models selected to be trained were based on these results.

1.2 - Motivation

About 600 patients die every year as victims of TBI in Portugal, according to a study provide by *Associação Novamente*, which is a portuguese association develop to provide support to TBI patients and their families [2]. In addition, a study developed by Oliveira et al. [3] in 2012 shows that the TBI incidence in European Union was 235 in 100000 individual. The incidence values for some countries were also collected, such as Finland and Portugal whose values represent 101 and 137 new cases per 100000 people each year, respectively. Thus, being Portugal a smaller country, its incidence value is more worrying.

The same study shows that the number of total TBI patients in Portugal has decreased over the years between 1993 and 2009. However, it remains an alarming situation. Regarding the mortality, the value was estimated to vary between 15 and 24.6 for each 100000 patients, with the portuguese value estimated at 17/100000 in 1997 [3].

The correct prognosis and the continuous monitoring are important to improve patient's health, since it can avoid the development of an undesirable outcome to the patient. In this sense, the presence of an additional tool that can help in the detection of the pathological events, namely the plateau waves, would be advantageous.

Thus, one of the focus of the present project consisted in the development of the mentioned algorithm to detect the plateau waves contained in data already collected from patients. The algorithm would be a more interesting application if it is a plugin of ICM+ to compute the predictions in real time. In this regard, it would provide a second opinion that does not invalidate the prognosis made by the physicians.

Furthermore, there are some relationships founded between the presence of plateau waves and some of the monitoring signals. However, there is no study focused in the EEG signal influence at the beginning of plateau waves. In these, the EEG signal and features extracted from the signal was used to investigate the mentioned relationship. Thus, it could be a starting point for further medical studies.

1.3 - Main Objectives

Plateau waves are frequent episodes in patients with acute brain lesion. Associated with the impaired cerebral autoregulation mechanism, the continuous multimodal monitoring helps to avoid long periods of increased ICP, and the emergence of a secondary brain injury.

To address the problem, the development of the algorithm for the detection of plateau waves was established as the first objective of this project. Such development should be based on machine learning techniques due to the capacity they have to design appropriate models for the data provided. Along with that, the selected software was MATLAB, because the data extracted from ICM+ can be imported and used there. It is a software developed by MathWorks, containing a several toolboxes to be applied in a wide range of areas, such as signal and image processing, wavelet analysis, neural networks, compute vision, deep learning, data analysis among others [4]. Moreover, it is a software that allows the training of several machine learning techniques. However, due to the unsatisfactory results obtained, the algorithm was developed using a different approach.

EEG signal present an unpredictable behaviour, because it is a stochastic signal. Despite this, it is not known if there is any change before all the plateau episodes that may represent a typical behaviour. Since there are some relationships already established between plateau waves and some of the monitored signals, the research for a possible relationship between the EEG signal and the beginning of the plateau waves was defined as a new objective to be achieved with this project. Thus, machine learning techniques were applied to investigate the mentioned relationship, using the labels computed by the previous algorithm and the data provided by UCNC.

1.4 - Contributions

Since from the first part of this project some interesting results arose, the developed work was presented in a meeting organized by the University of Porto, nominated as *Encontro de Investigação Jovem da Universidade do Porto*. Such meeting is organised for the student community to show the research projects that they are involved. Thus, dissertation projects can be presented in oral or poster presentation. This study was presented as a poster, containing a brief contextualization of the subject, accomplished with the results obtained from Weka and their discussion.

Associated with the scope of the plateau waves problem, there are some conferences and congress in several places to discuss this issue. One of them is the International Symposium on Intracranial Pressure and Neuromonitoring, which serves as a forum to exchange ideas and results associated with ICP, cerebrovascular autoregulation and other brain topics relevant to conditions such as TBI. They emphasise the multidisplinarity of ICP which allows the communications between engineering and clinicians. In this sense, this project was also submitted to be presented in this year edition that will take place in Leuven at the beginning of September.

Finally, due to the resulted obtained with the developed algorithm for the automatic detection of plateau waves, a contact with ICM+ owners was already established to try to create a plugin in the software using the model developed to detect the plateau. With this, it would be possible to use the data collected from patients in real time, as a tool that can provide some type of feedback about the patient's data.

1.5 - Structure of the Dissertation

This dissertation is organized in five chapters, in order to present a contextualization, the methodologies applied, and the results obtained. In this sense, this first chapter provides a brief contextualization about the subject addressed in the project and it defines the main objectives to be achieved. Also, it refers the importance of the development of the algorithm in the medical field. Finally, it presents the contributions of this study in this area.

The second chapter reviews the more important concepts for a fully understanding of the subject matter hereof. For that, the first section presents the several concepts related with neurophysiology to explain the definition and the detection of a plateau waves and the importance of the multimodal monitoring. In the second section, the machine learning concepts are introduced for an easier understand about the working principle of techniques that were applied to build the different models. Also in this chapter, there is a section dedicated to expose some studies already developed with application in NCCU.

Chapter 3 presents the most important techniques applied to develop the models that could better describes the data. Along with it, the input data used, the computation of the labels, and the definition of the parameters is also presented.

Regarding chapter 4, it shows the most significant results associated with the techniques applied, and it contains the discussion of the results to emphasise which proposed objectives were achieved.

The conclusion and future works are presented in chapter 5. Along with a summary of the work and the most promising results, there is a section containing suggestions for future work.

Chapter 2

Fundamentals of Concepts

Plateau waves are complex phenomena characterized by changes in brain signals. They are triggered by vasodilation cascade, resulting in an increase in ICP accompanied by a decrease in CPP. These phenomena may naturally finish by activating the vasoconstrictive mechanism, or in response to treatment. This later possibility is frequent in patients with severe TBI admitted in NNCU to avoid secondary brain injuries caused by long periods of high ICP and low CPP. The pathophysiologic concepts associated with the plateau waves, monitoring signals and their changes will be presented in the first section of this chapter to allow a better understanding of the problem addressed in this study.

Machine learning is a computational technique based on artificial intelligence to build the model that better describes data. Depending on whether the data label is available or not, the techniques can be separated into two groups, as supervised or unsupervised learning techniques respectively, and there are an extensive variety of them. In this sense, the second section of this chapter presents the concepts associated with machine learning and some of the techniques. It is important to acquire the knowledge about the working principle to decide which are the methods that may have a better performance for the available data.

Some of the works already developed in this field are presented in the third section of this chapter to get an idea about what have already been applied and what could result better. At the end of the chapter, the conclusion recaps the main concepts and ideas that must be retained from this chapter.

2.1 - Neurophysiological Concepts

2.1.1 - Introduction

The skull is the physical barrier that protects the brain, the blood vessels and the CSF from external damages. As a closed box, and accordingly to Monro-Kellie doctrine, these components have a total constant volume, which implies that any changes in the content of one of the components will have negative effect in the other two [5].

2.1.2 - Intracranial Pressure (ICP)

In normal conditions, ICP values should be between 0 and 15 mmHg. TBI, brain tumour, subarachnoid haemorrhage, benign intracranial pressure and space-occupying lesions [6-9] are some of the factors capable to cause an increase in ICP. Accordingly to the Davson's formula [10], the ICP can be defined as the sum of the CSF pressure component with the vasogenic component that is related with the continuous fluctuation in the cerebral blood volume (CBV) (equation 2.1).

$$ICP = ICP_{csf} + ICP_{vas} \tag{2.1}$$

ICP pulse waveform can be decomposed in three major peaks, P1, P2 and P3, where P1>P2>P3 is the amplitude relation in normal conditions, while P2>P1>P3 is the amplitude relation verified if the brain compliance decreases (Figure 2.1) [11].

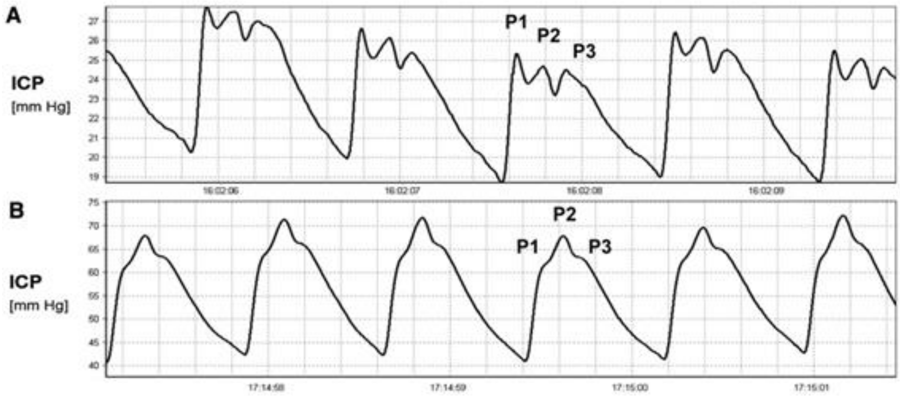


Figure 2.1 – ICP pulse waveform (a) during normal ICP and (b) during intracranial hypertension (adapted from [11]).

2.1.3 - Cerebral Perfusion Pressure (CPP) and Critical Closing Pressure (CCP)

CPP is the difference between the inflow and outflow pressures, represented by the mean arterial blood pressure (ABP) and ICP, respectively (equation 2.2). Since CPP is the driving pressure of cerebral blood flow (CBF), CBF is directly related with CPP and inversely related with the cerebral vascular resistance (CVR) (equation 2.3). Changes in brain vessels diameter in response to the variations of

ABP modify vascular resistance in order to protect the brain from ischemia and hyperaemia episodes. [12, 13].

$$CPP = ABP - ICP \tag{2.2}$$

$$CBF = CPP/CVR(ml/100g/min) \tag{2.3}$$

Critical Closing Pressure (CCP) is the pressure below which a blood vessel will collapse. If a CPP value is equal to the CCP value of the brain vessels, a critical situation of zero flow may develop. Thus, CCP represents the minimum ABP value to provide perfusion [12].

2.1.4 - Cerebral Autoregulation (CAR) and Cerebrovascular Reactivity Pressure Index

Cerebral Autoregulation (CAR) is a vital physiological mechanism of survival that assures a constant CBF between a lower and upper limit of CPP. Within the CAR interval, constant CBF is assured because the changes of CPP are coupled with the changes of the CVR and CBV. ICP is influenced by changes in CBV and therefore may be considered as a surrogate of the CBF (Figure 2.2). Thus, within the CAR interval, ICP and ABP changes are inverse related, whereas outside the interval they are directly related. The calculation of the coefficient correlation between CPP and ICP gives information about the cerebrovascular reactivity status to pressure. This coefficient was designated as pressure reactivity index (PRx) index. A negative PRx or below 0.2 estimates a normal CAR status, while a positive and above 0.2 index implies CAR impairment [14].

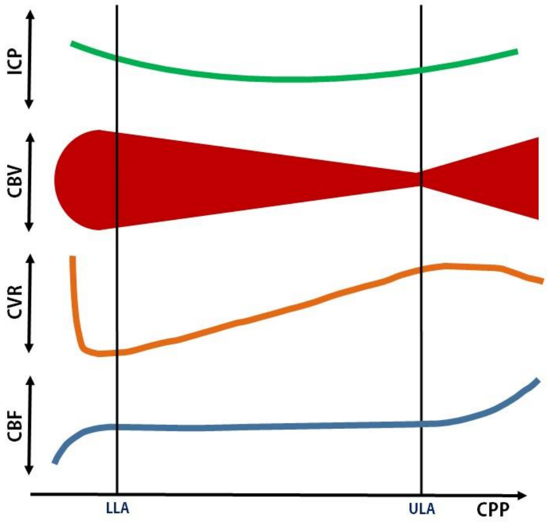


Figure 2.2 – Lower limit (LLA) and upper limit (ULA) of CCP that define the interval of CAR. ICP, CBV, CVR and CBF changes are represented according to the CPP variations.

2.1.5 - Pressure-Volume Relationship and Brain Compliance

Difficulty in the CSF absorption, CBF variations because of metabolic needs, abnormal tissue growth and accumulation of liquid in the interstitial and intracellular spaces are some of the causes that may increase the volume inside the skull, decrease brain compliance and, therefore, increase ICP. To minimize these impacts, the blood vessels are the first component to suffer alterations to reverse the rise in ICP. At the beginning, the volume of the venous compartment is reduced, followed by a decrease in the arterial one. The second component to adapt is the CSF. The final change to compensate the ICP rise, involves the brain parenchyma with a decrease in the water tissue content [15].

An exhausted pressure-volume curve relationship denotes a decrease in the compensatory reserve mechanisms and a decrease in brain compliance. This will result in an exponential increase in ICP (Figure 2.3).

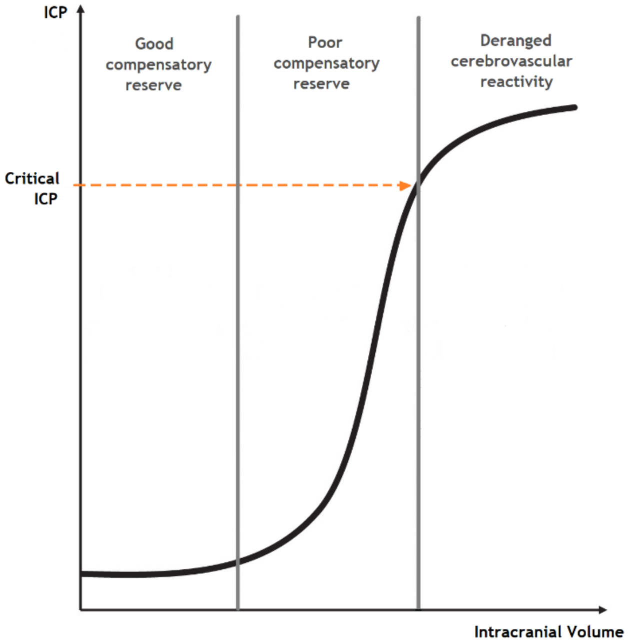


Figure 2.3- The relationship between the intracranial volume and ICP is described by pressure-volume curve. It has three parts that describes the state of the compensatory reserve: the good compensatory reserve, the poor compensatory reserve and the deranged cerebrovascular reactivity.

2.1.6 - Lundeberg A-waves or Plateau Waves of Intracranial Pressure

In 1950, Lundberg defined for the first time the sudden elevations of ICP above 40 mmHg during a minimum period of five minutes accomplished with changes in signals as CPP and HR, and with a spontaneous termination or in response to a treatment as “sudden rises in ICP”. Thus, this phenomenon is designated as Lundberg A-wave, A-wave or plateau wave [11].

Plateau waves are triggered by a vasodilation cascade loop that appear in patients with preserved autoregulation and low cerebrospinal compensatory reserve. These phenomena may end naturally or in response to an adequate treatment, activating a vasoconstriction mechanism that counters the vasodilation to reduce the amount of blood present in the brain and thus a decrease in the pressure inside the skull. The aim is to prevent the development of a secondary brain lesion.

Plateau waves can be divided and classified in four phases: the premonitory drift phase, plateau phase, ischemic response and the resolution phase. The first phase is marked by the reduction of CPP values and the beginning of the vasodilation mechanism. The beginning of the plateau phase is reached when ICP value is about 40 mmHg. A continuous vasodilation accomplished with a rise in ICP and a decay of CPP to ischemia level is observed in the third phase. This ischemic response ends with a sudden termination spike, that may not be noticed if the CPP restoration to baseline values is slow. The fourth phase is controlled by the vasoconstriction mechanism in order to reduce CBV and to restore and stabilize ICP and CPP signals to the baseline level [8, 12].

Intracranial hypertension is commonly observed in patients admitted in NCCU with TBI and low Glasgow Coma Score (GCS) [8]. Nevertheless, the rise in ICP may be related to several causes apart from A-waves. There are three other situations when a rise in ICP can result in a misclassification of intracranial hypertension: a passive transmission of the pressure in the arterial compartment to the intracranial compartment, a decrease in ABP accomplished with a rise in the ICP due to increase in CBV to compensate the changes in brain perfusion and higher ICP values secondary to severe hyperaemia [16]. Therefore, continuous multimodal brain monitoring is recommended to help physicians with the differential diagnosis, to manage brain hemodynamic and to avoid secondary brain injury (Figure 2.4).

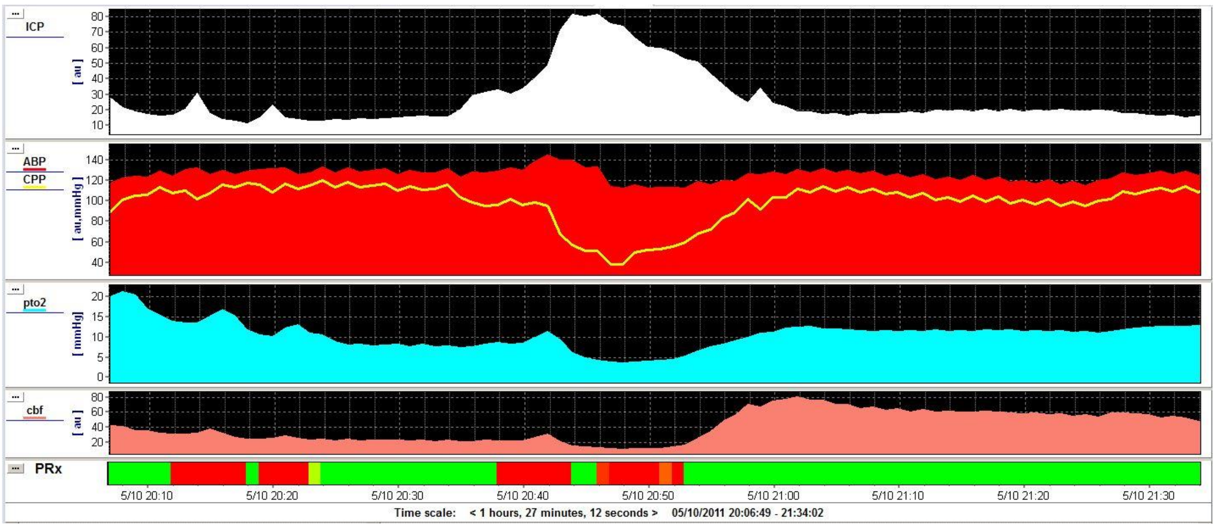


Figure 2.4 – Multimodal brain monitoring in ICM+, in which plateau wave can be observed. From upper to lower chart, ICP, ABP, CPP, pbtO$_2$, CBF and PRx are displayed.

To study a plateau wave, the highest modal frequency (HMF) can also be considered, since it is reversely related with CPP. In this sense, while the CPP decreases the HMF increases and it decreases with a rise in CPP, which is explained by the reduction and elevation in the intracranial compliance, respectively [17].

2.1.7 - Multimodal Brain Monitoring

Several brain and systemic signals, such as ICP, CPP, HR, ABP, pulse amplitude, brain temperature, brain tissue oxygen tension (pbtO2), cerebral oximetry with transcutaneous near-infrared spectroscopy, respiratory markers (end-tidal carbon dioxide ($ETCO_2$) and respiratory rate), CBF with thermal diffusion method and EEG, are simultaneously recorded and saved as raw data in ICM+ software.

ICM+ is a software developed by the clinical neuroscience department of Cambridge University, specially designed to display the data collected from neurocritical patients in real time. Alternatively, it can be used for later analysis in offline mode to develop clinical decision tools.

Besides the collection of data of ICP, ABP, CPP, HR and amplitude of intracranial pressure (AMP) signals from patients, ICM+ calculates secondary variables. Most of those variables are related with the cerebrovascular reactivity and the cerebral compensatory reserve, such as: PRx, pressure amplitude index (PAx), oxygen reactivity index (ORx), cerebral blood flow reactivity index (CBFx), index of cerebrospinal compensatory reserve (RAP) and the index of autoregulation with transcranial Doppler (Mx). PRx results from five minutes moving window considering ICP and ABP, while PAx considers the same window but with ABP and the pulse amplitude of ICP signals. ORx is the average value of 60 minutes between CPP and $pbtO_2$ and CBFx consists in the moving correlation established between CBF and CPP. Mx describes independency between the changes that occur in the blood flow and in CPP if it takes a negative value; otherwise it describes a dependency. RAP coefficient describes the state of the compensatory reserve since it correlates AMP and mean ICP [9]. Therefore, a coefficient close to 0 reveals a good compensatory reserve index, since there is no relation between the mentioned signals. On the other hand, a coefficient below 0 indicates an exhausted cerebrovascular reactivity index, and a RAP close to 1 suggest a poor compensatory reserve.

There are other parameters that can be calculated in ICM+ as a representation of the relationship between PRx and CPP. It is the U-shaped curve and it indicates the way in which the failure of the cerebral autoregulation evaluated with PRx is distributed across CPP values along time using a window of 4 hours. The minimum value of U-shaped curve corresponds to the optimal cerebral perfusion pressure (CPPopt) (Fig 2.5), which is the value of CPP that optimizes CAR.

2.2 - Machine Learning Concepts

2.2.1 - Introduction

Machine learning consists in a set of techniques, in which a mathematical model that better describe the input data is built. The model learns and trains from the meaningful information extracted from data, known as data mining. For this reason, a larger and heterogenous dataset used in the training process will result in a model with a higher accuracy, once a set of different situations is

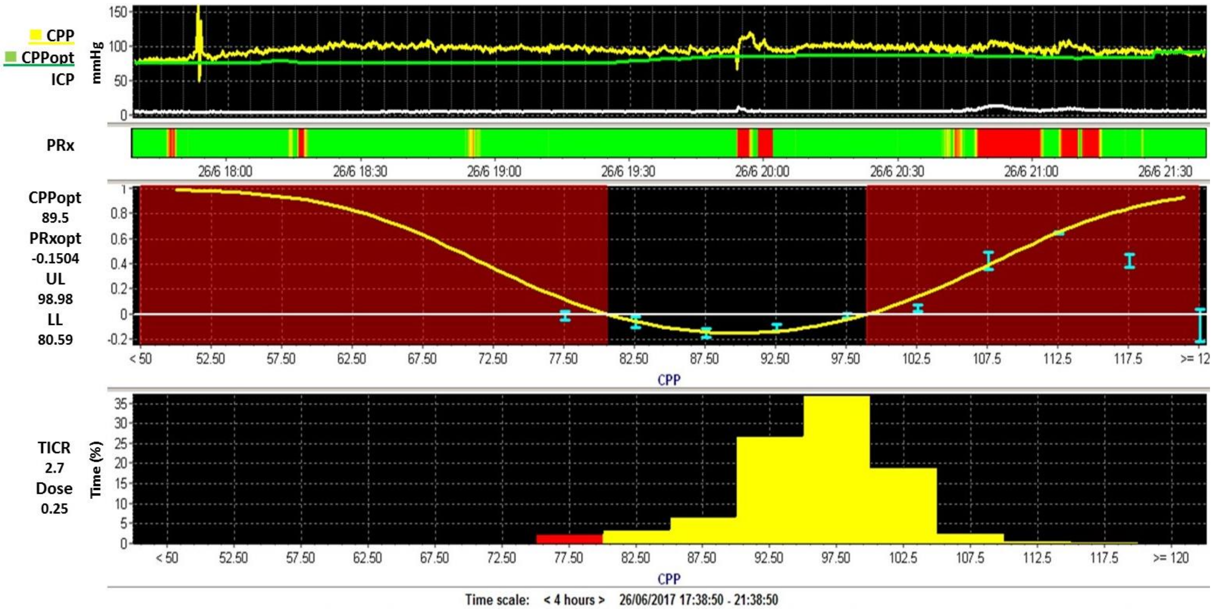


Figure 2.5 – Multimodal brain monitoring chart of ICM+, with the plot of CPP, CPPopt, ICP and PRx in the first two panels. The visualisation of PRx against CPP results in the plot of U-curve, presented in the third panel. Its minimum value corresponds to CPPopt. The black section represents the optimal range, which corresponds to the PRx values above the threshold set to 0. The last panel contains the time in critical region (TICR), represented in red in the histogram, corresponding to 2.7% of the total time of the window presented. The dose value results from the computation of TICR with the mean CPP value during that period of time.

presented and consequently, in testing process, the model already knows how to manage different scenarios of data. Ideally, one set of data must be provided to train, one set for the validation process and other must be used to perform the tests in the model. If only one dataset is available, it can be split in two subsets, for the training and testing processes, or in three if a validation subset is required [18, 19]. The validation step is used to adjust the hyperparameters to avoid the overfitting of the data. A model that fits perfectly each training data is designated as overfitting, because the classifier hyperparameters have a biased behaviour. However, if the error of the model is high, it represents a underfitting of the data. A model without an overfitting or underfitting behaviour has an appropriate capacity since it fits in a larger range of training and testing values. This results in a machine learning model with better performance [18].

The performance of the machine learning algorithms is evaluated through the accuracy and the error rate. The accuracy represents the proportion of the output data that were correctly classified, while the error rate consists in the proportion of data that were incorrectly classified. In testing process of a supervised learning technique, it is also possible to calculate the confusion matrix, which is a summary of the predictions that were correctly and incorrectly assigned. Thereby, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) instances are calculated. The true term is associated with a correct prediction whereas the false exposes an incorrect classification. On the other hand, the positive and negative terms reveal the presence or the absence of an abnormality, if it is a problem with two classes. After confusion matrix calculation, it is possible to determine the specificity and the sensitivity of the model. These performance evaluators are calculated based on the results from test set, since it is the data set that the model had not interact so far [18].

The machine learning techniques can be separated in two groups: the supervised and the unsupervised learning techniques, if the input data is accomplished by their label or not, respectively. A supervised technique learns from the data and the label, while an unsupervised technique groups data accordingly with the similarity between data and the probability distribution [18]. Once there is an extensive list of algorithms that can be classified as supervised or unsupervised learning techniques, just the ones applied in the development of the algorithm will be presented and explained.

2.2.2 - Supervised Leaning Techniques

As previously mentioned, the supervised learning techniques use the input data and its label, even if it is automatically generated, to design the mathematical model. These techniques can be used for classification if the data points are classified into one of two possible classes or for regression tasks if the output is a real number [18].

2.2.2.1 – Linear and Logistic Regression

Linear and logistic regression are both linear techniques, although the logistic model considers the logarithmic scale to obtain the result. These methods place a hyperplane that minimizes the distance between itself and the data point provided. The main difference between these regression techniques is the number of output values that can be assumed: two in the case of linear regression and more than two for the logistic regression [20, 21].

Linear regression models are widely used in computer sciences, quantitative problems, and in areas such psychology and medicine, since it allows the prediction of clinical outcome for patients and to quantify the influence of some drugs considering age, sex and other factors. The predictor outcome of the linear regression is based on the result obtained from the sum of the weighted features with the intercept, which is a coefficient without any feature associated, and with the error portion that describes the difference between the real and the predicted output, ε. Therefore, a linear regression model can be calculated using (equation 2.4):

$$y = b_0 + b_1 x_1 + \cdots + b_p x_p + \varepsilon\,, \tag{2.4}$$

or alternatively as (equation 2.5):

$$y = b_0 + \sum_{i=1}^{p} b_i x_i + \varepsilon\,. \tag{2.5}$$

The weight for each x_i is calculated through the ordinary least squares method to minimize the squared difference defined between the real and the predicted output (equation 2.6) [21].

$$\hat{b} = \min \sum_{i=1}^{n} \left(y^{(i)} - b_0 + \sum_{j=2}^{p} b_j {x_j}^{(i)} \right)^2 \tag{2.6}$$

Logistic regression models are also used in medical context and one of its applications is the prediction of the amount of microbiology present in food. This type of regression calculates the probability of the occurrence of an event, designated as dependent variable, which is a dependency of factors as nominal or ordinary variables. Hence, the output value must be between zero and one.

The probability is calculated based on a logistic sigmoid function, shown in equation 2.7, where z is the sum of the weighted features with the intercept and the error. The result of the mentioned sum can give the probability of the event logarithmically (equation 2.8).

$$\sigma(z) = 1/\left(1 + e^{(-z)}\right) \tag{2.7}$$

$$logit\ [P(event)] = b_0 + b_1x_1 + \cdots + b_px_p \tag{2.8}$$

In logistic regression models, the weights of the features are calculated through the maximum likelihood, which consists in a method used to estimate the parameters to maximize the probability of the data and to avoid the overfitting of the data. It is achieved when the hyperplane separates the two cases. Thus, it does not depends on the number of data points neither the number of parameters [20, 21].

2.2.2.2 - Support Vector Machine (SVM) Technique

SVM is a feedforward network technique where the training data and their labels are used to position an optimal hyperplane to separate each class in each side. In the case of a two-dimensional space, the hyperplane corresponds to a line that splits the plane in two classes and each data point is predicted to belong to one of them. Each prediction is calculated using a weight vector, w, and a bias, b, and it belongs to the positive class if the result of $w^Tx + b$ is positive (equation 2.9); otherwise, the point belongs to the negative class (equation 2.10).

$$w^Tx + b \geq 0 \tag{2.9}$$

$$w^Tx + b < 0 \tag{2.10}$$

The support vectors correspond to the most difficult points to classify since they satisfy both equalities presented. Thereby, they are placed in the hyperplane or decision surface and are the responsible for the optimal positioning of the hyperplane (Figure 2.6). The remaining data points are not relevant in this learning process. The number of the support vectors extracted from the training data determines the dimensionality of the feature space [18, 22, 23].

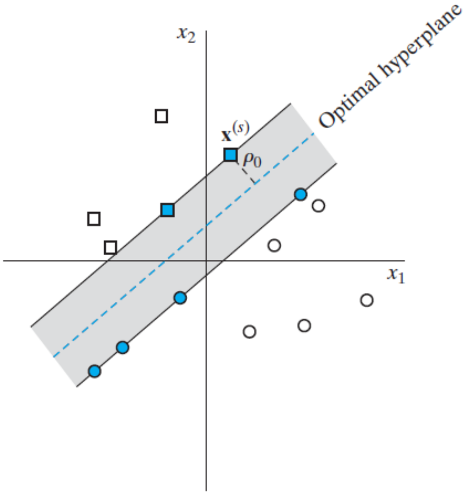


Figure 2.6 – Representation of the optimal hyperplane positioning in case of linearly separable data. The blue points represent the support vectors (adapted from[22]).

The margin of separation, ρ, consists in the separation between the hyperplane and the nearest points of each class. Thus, the hyperplane must be positioned to maximize the margin of separation. Once in most of the cases it is impossibility to find an optimal hyperplane that separates data without errors, the aim is to find the one that minimizes the probability of classify errors. The deviation of the point from the correct pattern of separation is designated as slack variable, ξ, and its value varies from 0 and 1 if the point is placed in the correct side of the region of separation; otherwise, its value is above 1 [22, 23].

Non-linear regression and pattern-classification are examples of regression and classification tasks, respectively, that can be performed using SVM. In addition, SVM can be used as unsupervised learning technique if the label of the data is not available [18].

2.2.2.3 - Neural Networks Technique

In a neural process, a stimulus is detected by the receptors presented in the body. That sensorial information is processed in the brain to generate an appropriate response executed by the effectors (Figure 2.7). This approach can also be used to explain, in a similar way, the neural network technique. Thus, the input data is provided to the algorithm and based on the information extracted from that, a model that better describes the data and better predict new instances is designed [22].

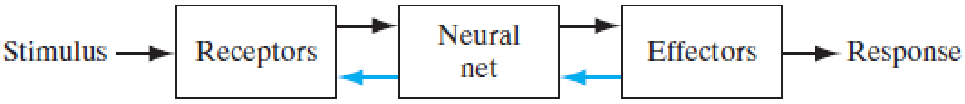


Figure 2.7 – Block diagram of a neural process that can be used to explain the basics of the neural network technique (adapted from [22]).

The good performance of this technique is based on a massive interconnection of the simple cells designated as “neurons” or “processing units”. As in the human brain, the death of the cells results in new connections, although the artificial neurons are rudimentary compared with the ones present in the brain of the humans.

Considering x_i as the input of the synapse and w_{kj} as the weights connected to a neuron k, that may assume positive or negative values in contrast to the weights of the cerebral synapses. An adder, Σ, sums the weighted input signals as a linear combination resulting in a value that can trigger the activation function to generate an output signal (Figure 2.8). The presence of the bias, b_k, increases or decreases the input of the activation function as an affine transformation, that consists in a linear transformation, mx, followed by a translation, b.

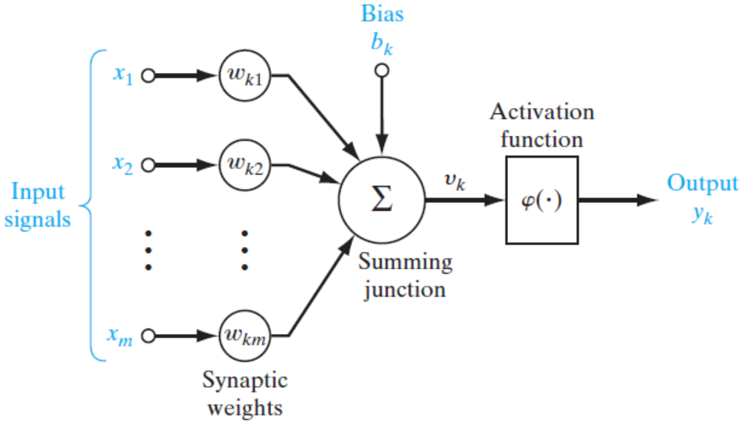


Figure 2.8 – Representation of a neural network model, considering a bias (adapted form [22]).

Instead of the bias, b_k, a fixed weight value, w_{k0}, of $+1$ can be used (Figure 2.9).

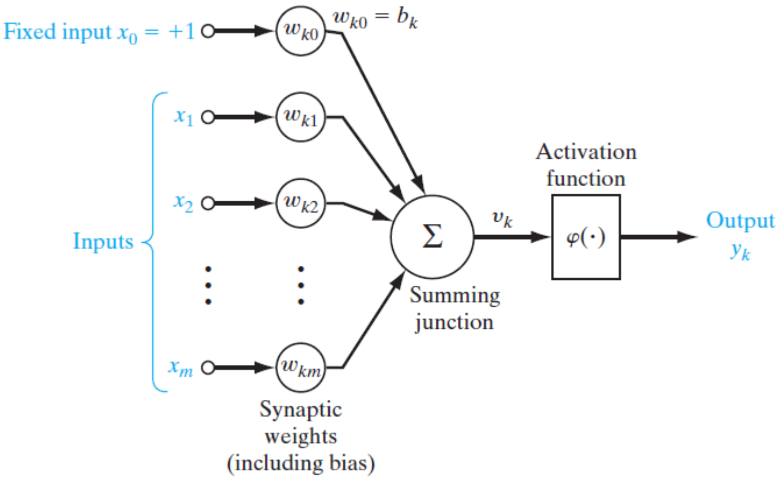


Figure 2.9 – Representation of a neural network model, considering a fixed weight value (adapted from [22]).

The activation function, $\varphi(v)$, influences the output of the neuron and it can be a threshold or a sigmoid function. The threshold function limits the output value accordingly to the induced local field of the neuron (equation 2.11), v, while the resulting value of the sigmoid function (equation 2.12) assumes any value between 0 and $+1$, or if redefined in the range of -1 and $+1$.

$$\varphi(v) = \begin{cases} 1 & if\ v \geq\ 0 \\ 0 & if\ v <\ 0 \end{cases} \tag{2.11}$$

$$\varphi(v) = \frac{1}{1+ e^{-av}} \tag{2.12}$$

Figure 2.10 illustrates the behaviour of both activation functions mentioned, accordingly to the induced local of field of the neuron value.

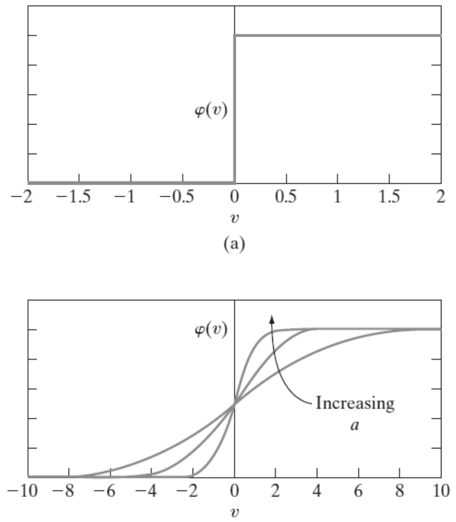


Figure 2.10 – Graphical representation of both activation functions: (a) the threshold function and (b) the sigmoid function, according to the local of field of the neuron *v* (adapted from [22]).

Euclidean distance is used to measure the similarity of the data, since the minimal distance is an indicator of a higher approximation between the elements of the input vectors x_i and x_j.

The neural networks architecture is divided in single-layer feedforward networks, multilayer feedforward networks and the recurrent networks. The feedforward concept implies that the input layer of the sources nodes is projected to the output layer and not in the opposite direction. Moreover, while the single-layer feedforward network has a unique layer of neurons (Figure 2.11 (a)), or computation nodes, the multilayer feedforward network has at least one hidden layer with the hidden units or hidden neurons. These hidden units are the output of the computation of the previous layer and, simultaneously, the input of the subsequent layer [22]. The “hidden” term reflects the fact that these layers are not observed neither from the input or from the output. If in this type of network every node is connected to each node present in the subsequent layer, it is called as a multilayer feedforward network fully connected (Figure 2.11 (b)). In the case of recurrent network, the output signal can be used to produce the same output or others, acting as feedback with an impact in the performance and on the learning capability of the network (Figure 2.11 (c)) [22].

In a supervised learning approach, it is possible to compare the predictions with the real labels and calculate the error signal, the mean-square error and the sum of the squared errors. However, this technique can also be used as unsupervised learning to create new classes automatically that better represent the features of the input data [22].

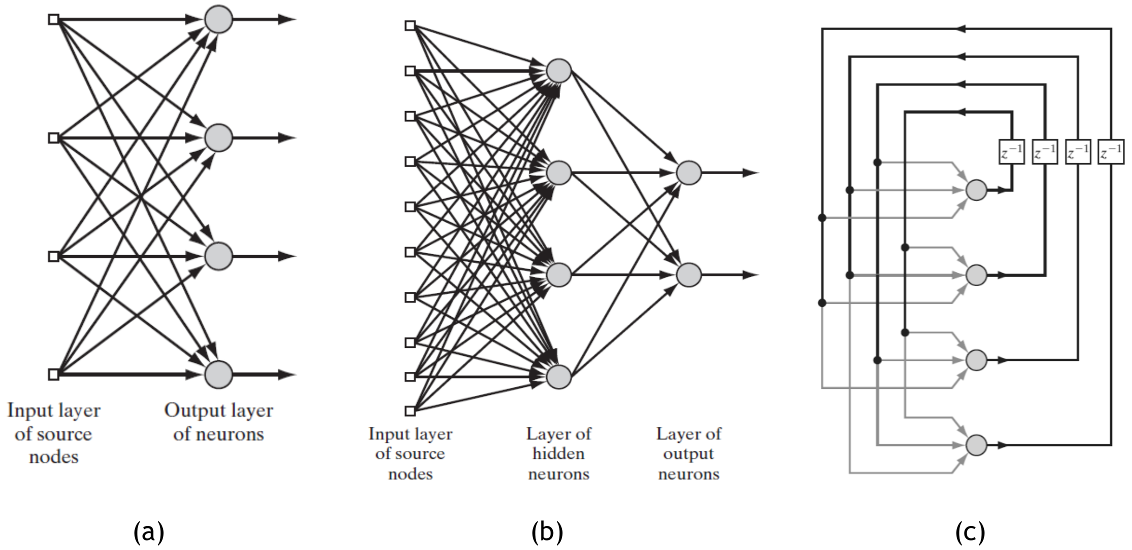


Figure 2.11 - Representation of the tree neural networks architecture: (a) a single-layer feedforward network, (b) a multilayer feedforward network fully connected and (c) a recurrent network where the outputs also are the input of the model (adapted from [22]).

As the main properties and capabilities of neural networks it is possible to enumerate: the non-linearity, input-output mapping, adaptability, evidential response, contextual information, fault tolerance, very-large-scale-integrated (VLSI) Implementability, uniformity of analysis and design and the neurobiological analogy [22].

2.2.2.4 – Gaussian Process

Gaussian process is a supervised learning technique and a nonparametric kernel-based probabilistic model convenient for modelling tasks in machine learning and statistics. It can be used for reinforcement and multi-task learning, spatial model, times series and optimization [24-26]. Neal demonstrate in 1995 that a large class of neural network models will converge to a Gaussian process in the limit of an infinite value of hidden units [27].

This is a stochastic process that can be fully defined by mean and covariance functions, which are difficult functions to be totally defined *a priori* since the hyperparameters must be inferred [25, 27]. The covariance function contains three parts: the first term that expresses the idea that similar inputs originate more correlated outputs, the linear regression term and a noisy term. The predictors of this model will depend on the covariance function chosen. The hyperparameters can be estimated using a maximum likelihood or a Bayesian approach. The maximum likelihood adjusts the hyperparameters to maximize the likelihood of the training data points. The values are randomly initialized and an iterative process leads to the optimal values [20, 27]. Another important feature is the simplification of the training process since there are no weighting parameters, and hence none external method as cross validation is needed. However, the computational limitations is an issue associated with this technique, because of the memory and the cubic time complexity [24].

The gaussian process infer the latent function, f, using the n training points and their label y in the function space defined by the mean, m.

Once it is kernel based, there are ten different kernel covariance functions available in MATLAB to build the gaussian process model: squared exponential kernel, exponential kernel, matern 3/2, matern 5/2, rational quadratic kernel, automatic relevance determination (ARD) squared exponential, ARD exponential, ARD matern 3/2, ARD matern 5/2 and ARD rational quadratic [28]. The kernel is responsible to control the smoothness of the process and the most used is the ARD squared exponential [26].

2.2.3 – Unsupervised Learning Techniques

The models in which the label of the input data, x, is not available to train and to supervise the learning process are known as unsupervised learning techniques. In this sense, these models try to find the representation that better describes the data, using density estimation by the distribution of the data, clustering of similar examples and visualisation based on a projection of the data from a high to a low-dimensional space [18, 20].

2.2.3.1 – Principal and Independent Component Analysis

Principal and independent components analysis are both statistical transformation techniques, mainly used to collect information from second or higher order statistics, respectively, in applications such as feature extraction and neuroscience [29].

PCA is a simple and effective method used to compress data since it learns a representation of the data. Therefore, it is widely used to reduce the representation of the original data to a lower dimensionality [18]. The new coordinates are obtained by rotating or translating the original, starting by moving the centre of the coordinate system to the centre of the data. This is followed by the repositioning of the x-axis into the principal axis of variation, which corresponds to the axis where more variations of all data points occur, and the other axis into the orthogonal direction of variation [29]. The data transformation results in elements mutually uncorrelated, which is an ability and an important property of PCA [18].

The principal components can be derived from a maximization of the variance or from a minimization of the mean-squared error representation. Using the maximization of the variance, the first principal component is placed in the direction with the largest variance while the second principal component is placed to maximize the variance in all the orthogonal directions to the first. Eigenvectors result from the covariance matrix and they are the principal components of the data. The aim in the minimization of the mean-squared error is to find a projection with the smallest average distance, or mean-squared, between the original vectors and the projection in the principal components [30].

ICA is an analysis tool widely used to separate signals in their components, but it can also be used to generate data using a nonlinear generative model, to estimate the density of the data, or feature extraction. The independent components, known as latent variables, are estimated throughout a whitening or a sphering process and are used to find the transformation of the feature space into a new space [18, 29]. Associated with the ability to separate the mixing components of a signal, ICA model can be described as a linear combination of n variables (equation 2.13):

$$x = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}\, s_i \tag{2.13}$$

or, using a matrix A containing all a_{ij} elements with a random vector s representing all $s_1, \dots, s_n$ elements, as (equation 2.14):

$$x = As \tag{2.14}$$

The ambiguities of ICA are associated with the impossibility to determine the variances or energies and the order of the independent components. The fact that the independent components are assumed as statically independent, that their distribution must have nongaussian distribution and the number of components is equal to the number of observed mixtures represent restrictions of this method [29].

2.2.3.2 - Clustering Algorithms

Clustering is an unsupervised learning algorithm that groups the data accordingly to the similarities between the features to find a trend in the data. This technique has two different approaches: the divisive and the agglomerative clustering. The first consists on a recursively split of the data into smaller groups until a good clustering is achieved. An algorithm illustrative of the divisive clustering is random tree. The agglomerative clustering starts with each point as a cluster, and the two clusters with the smallest inter-cluster distance are merged until a satisfactory clustering [31].

The similarity between two objects is calculated by the distance between them, using the Euclidean distance (equation 2.15) or, alternatively, using Manhattan or Minkowski distance metrics [31, 32].

$$d_p(x, y) = \sqrt[p]{\sum_{i=1}^{N} |x_i - y_i|^p}, \qquad for\, p \geq 1 \tag{2.15}$$

K-means is the most known clustering algorithm and it requires the definition of the number of clusters to initialize the centres randomly. Then, each point is associated to the class, as a class membership $c^{(i)}$, where the distance to the centre point is minimal. In this sense, each point is assigned to the centroid of a cluster using one of the indicated distance formulas. After this association, each centre is repositioned to be more equidistant to each data point of the cluster. The reassigning of the points to a cluster and the relocation of the centroid position are two steps repeated until the positions of the centres are unmoved between two subsequent iteration or until the maximum number of iterations [20, 31].

The aim of the k-mean algorithm is to minimize the cost function (equation 2.16) to obtain the optimization function (equation 2.17).

$$J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k\right) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2 \tag{2.16}$$

$$min_{c,\mu}J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k\right) \tag{2.17}$$

Figure 2.12 illustrates the working principle of the k-means algorithm. Two centres are randomly placed and, as the iterations progress, the centres are repositioned until their location is unchanged, as in the last two iterations. Usually and as illustrated in this example, the lower number of initial clusters requires fewer iterations until the model converges.

One of the problems encountered in k-means is the sensitivity of the method to noise and outliers, the bias and the difficult to determine an initial suitable value for k.

Fuzzy C-means is another clustering method in which each sample belongs to two or more clusters. Alternatively, mean-shift clustering is a non-parametric clustering method, which does not require the definition of the initial number of clusters. It detects arbitrary shapes of clusters because it is a density estimator [32].

2.3 - State of Art: Similar Applications in Neurophysiological Field

There are numerous machine learning applications in ICP, increased ICP and NCCU. Therefore, a set of studies and works already developed in this field will be presented. The purpose of the present research sought to understand which software are already available in the market and which techniques have been already applied in this context.

Technological advances enable several progresses in NCCU. One of them is the software developed by the University of Cambridge, ICM+, used to display the signals collected from the patients during continuous monitoring. Besides, machine learning techniques were suggested by Sivaganesan et al. (2014) [33] to develop alarm algorithms applied in real-time data. One of the machine learning algorithms is classification trees since it is capable to create patients' categories and define data relationships for each one. The other applied technique is neural network because it can find patterns in data to estimate the parameters. Moreover, neural network can extract rules from data about a relation between the diagnostic classifications and the patient attributes. The last machine learning algorithm mentioned is decision trees that have been used to extract some attributes to classify the input data as signal or noise.

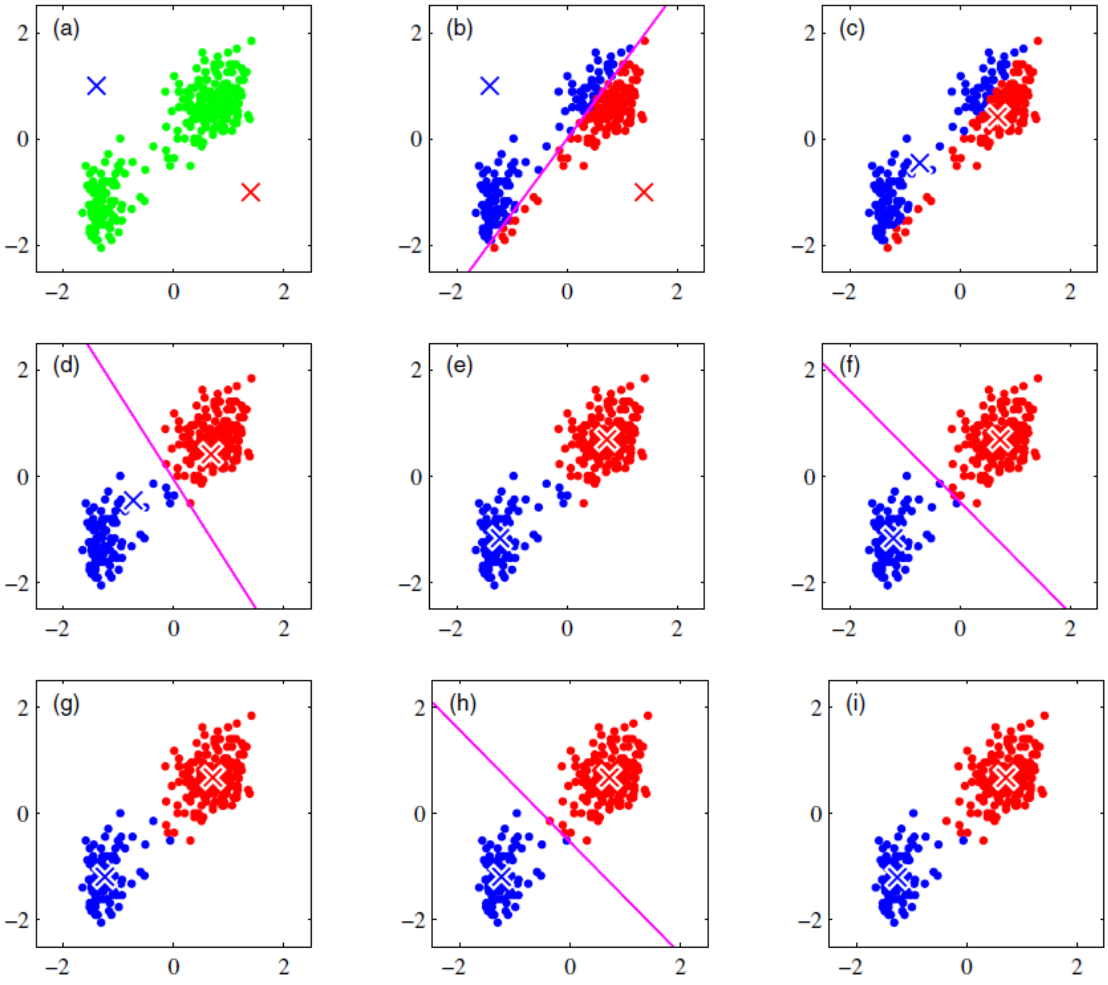


Figure 2.12 – Representation of k-means model showing the evolution of the model with the increased number of iterations: (a) the model starts with two centres randomly positioned, (b) dividing the data into two classes, (c) the first iteration is performed and (d) two new classes arise. The (e)-(h) steps are repetition of the process performed in (c)-(d) steps until the centres between two subsequent iterations do not change their position, as is observed in (g) and (j) (adapted from [20]).

In addition, a treatment recommendation system for TBI patients had been developed using a combination of machine learning techniques with logistic regression. Another evolution in informatics in NCCU use a set of rules with a fuzzy logic model to develop an automatic tool for adjusting the fluid and drug infusion rates to maintain the haemodynamic of each patient [33].

A work published by Güiza et al. (2013) [34] developed predictive models using multivariate logistic regression and gaussian processes to predict an increase in ICP 30 minutes in advance and for early prediction of the neurological outcome after 6 months. For that, both models consider dynamic predictors derived from the analyses of ICP and mean arterial pressure (MAP) signals. In the case of increased ICP episodes, the predictors use 4 hours intervals of ICP and MAP, and the number of episodes in the first 24 hours after the admission in the intensive care unit. The early prediction of the outcome considers the predictors derived from ICP and MAP in the first 24 hours in the intensive unit, since it improves the algorithm performance.

Gaussian process models were successfully employed in applications in intensive care units because of its flexible modelling, high predictive performances, small number of model parameters and probabilistic outputs. This method uses means and covariance functions to evaluate the similarity between data points, resulting a prediction which is the combination of the known outcomes of the similar data used in training process. The performance of the increased ICP episodes model was evaluated using the area under the receiver operating curve, with a value of 0.847 in the development cohort and a value of 0.87 in the validation. In the case of the early prediction at 6 months neurological outcome, logistic regression models had poor performance, while gaussian process models had an excellent overall performance [34].

To conclude, this study [34] creates a gaussian process model that accurately predict increased ICP episodes 30 minutes in advance using the information of ICP and MAP in the previous 4 hours, suggesting that ICP and MAP correlation is a relevant predictor that indicates the importance of cerebrovascular autoregulation in the acute and chronic evolution of the TBI patient. Besides, the number of increased ICP episodes during the first 24 hrs in intensive care unit is predictable for neurological outcome at 6 months, also using the gaussian process model with better performance.

Chen et al. (2010) [35] study developed a SVM model to predict ICP level in TBI patients. The model uses an extensive set of features based on midline shift measurements and texture extracted from CT scan images, and demographic information, such as age, of each of the 17 patients involved. After the cross validation, the sensitivity of the SVM model is around 65%, whereas the specificity and the accuracy round 70%.

The analysis of ICP waveform can be difficult because of multi-scale features, signal morphology and signal-to-noise limitations. In this sense, there is an algorithm, Scholkmann algorithm, developed to be used in peak and trough detection. However, it requires a high computational time and a sizeable memory. The work developed by Bishop and Ercole (2018) [36] was focused on the improvement of the algorithm runtime performance. Thus, after changes in trough detection and in calculations associated with local maxima scalogram, the modified Scholkmann algorithm shows false-positives (0%) and false-negatives (0%) rates comparable with the ones obtained using the original algorithm. However, both algorithms cannot detect the first peak and the last trough, which denotes a limitation.

Some intensive care units have alarms that trigger when abnormal values of ICP are detected. However, in some situations, the alarm detection is false, and it may be due to noise or signal artefacts. Thus, there is the need to create an intelligent system to reduce false alarms. In this sense, Scalzo et al. (2012) [37] study introduces a framework to detect false alarms based on patterns in ICP morphology. For that, three machine learning techniques — spectral analysis, kernel spectral analysis and SVM — were applied and they showed improvements in alarm detection, compared with the conventional method that uses threshold values.

Meyfroidt et al. (2011) [38] developed models to predict discharge after a cardiac surgery based on the gaussian process method. One of the gaussian process models were designed to predict the probability of discharge the day after the surgery, as a classification task, and to predict the day of the discharge in a discrete variable, as a regression task. Both models use the collected data of four hours after the admission of the patients in the intensive care unit. As results, the prediction of both models has better performance than the prediction made by nurses and similar to physicians.

The dynamic cerebral autoregulation assessment is based on the analysis of the cerebral blood flow velocity (CBFV) as a response to change in ABP. To evaluate the dynamic cerebral autoregulation, instead of producing ABP fluctuations by manoeuvres, such as thigh cuff and sit-to-stand, Chacón et al. (2018) [39] developed a linear and a nonlinear finite impulse response models. These support vector regression models use the spontaneous ABP and CBFV variations of the twenty healthy patients submitted to the study. The dynamic cerebral autoregulation efficiency measurement using the developed models is statistically equivalent to the most used methods based on the manoeuvres.

In terms of unsupervised learning, there are some applications using ICA, namely in neuroscience, since this is a technique commonly used to separate an observed signal into its underlying signals that must be fully independent. An electroencephalogram consists in the set of signals captured by the electrode sensors positioned on the head. Besides the brain signal, the eyes and heart signals are also captured and mixed together, resulting in a confusing measurement at the scalp of the subject. Therefore, ICA separates the brain and the heart signals and, in addition, the brain signal accordingly to regions [18]. This technique has other applications for biomedical data, such as in fMRI, optical imaging, ECG and in the extraction of artefact in EEG and MEG data [29].

2.4 - Conclusions

Subarachnoid haemorrhage, traumatic brain injury and acute hydrocephalus are some examples of acute brain injury that might increase the pressure inside the skull. If the increase of ICP is above 40 mmHg for a minimal period of five minutes accomplished with changes in signals such CPP, CBF, AMP and FV, it can be a plateau wave. Plateau waves are triggered by a vasodilation cascade loop that may end naturally if there is an active cerebral autoregulation or in response to active treatment.

Patients with TBI are admitted in NCCU and there management is ruled by continuous multimodal monitoring to avoid the development of a secondary brain injury, namely due to plateau waves.

Machine learning includes a set of techniques based on artificial intelligence capable to learn and extract important information from the provided data to design a model. This model is the one that better describes and fits in the data, and its performance is evaluated using a set of new data to test it. As evaluators, there is the error rate and accuracy, among others.

Linear and logistic regression, SVM and neural networks are supervised learning techniques since they use the label of the data during the learning process of the model to guide it. The regression models place a hyperplane to minimize the distance between itself and the data points of the different classes. In both cases, the outcome results from the sum of weighted features with a coefficient without a feature associated and an error portion. Although, these are both linear techniques, the logistic regression uses the logarithmic scale.

SVM also positions a hyperplane but the prediction is calculated using a weight vector, w, and a bias, b. In the case of a problem with two classes, the data point belongs to the positive class if the result of the indicated sum is positive; otherwise it belongs to the negative class. The hardest points to classify are those that verify the mentioned equalities, and, for that reason, they are called the support vectors. These points are responsible for the positioning of the hyperplane, that must maximize the separation between the positive and the negative classes.

A neural network technique can be explained as a neural process, since the input data is a stimulus that is processed in the neural net to generate an appropriate output, which is the response. In this sense, the input data and the weights are summed as a linear combination to trigger the activation function — a threshold or a sigmoid function —, that determines the resulting value. There are hidden neurons where the computation of features and intermediate calculations occur.

On the other hand, the PCA, ICA and clustering algorithms are unsupervised learning techniques, because the label of the data is not used in the learning process. In this sense, the algorithms must find a trend of the data or learning from the distribution of the data to build the best model. PCA and ICA are both linear transformation techniques, where the feature space is transformed into a new space. In the case of the principal components, the first axis is aligned with the principal axis of data variation. Although both analyses are used to reduce the dimensionality of the feature space, ICA is also used to separate a signal in its components.

Regarding clustering algorithms, the data is grouped based on similarities between data points. K-means is one of the clustering algorithms where the similarity of the data is calculated using the Euclidian distance, in most cases. This machine learning method requires the definition of the number of clusters, so that the centres are randomly positioned. Their position is changed with the reassignment of the data points to clusters, until they are unchanged between two subsequent iterations. Alternatively, there is Fuzzy C-means, where a data point can belong to two or more classes, and mean-shit clustering algorithm that does not require an initial number of clusters.

Based on the applications and studies developed in this area, the methods that may result in better models for the provided data and to achieve the defined objective are SVM and gaussian processes, since they were extremely used to predict increased ICP episodes.

Chapter 3

Methodology

3.1 - Available Data

The data available for the development of the present study were extracted and provided by the UCNC database. It contains a heterogeneous set of data collected from patients admitted in the intensive care unit in the last few years. The main reasons for admission in the NCCU are traumatic brain injury, subarachnoid haemorrhage and ischemic stroke.

A total of 100 files was anonymously provided to train and test the different machine learning techniques applied to find the model that better describe the input data and predict the new instances. All the files contain ICP, CPP, AMP, ABP, ETCO$_2$, HR, ETCO$_2$ and one-channel EEG signals. In addition, there are calculated variables, such as PRx, CPPopt, PAx, ORx, Mx and mean values using an average filter with a window of 10 seconds, based on the measured signals. In some patients, CBF, brain temperature and cerebral haemoglobin oxygenation index (rSO$_2$) were also recorded.

All the files were classified according to the quantity and quality of EEG data recordings. In this sense, files with the EEG signal in the entire or almost entire recording time were separated from those with an intermediate quantity of signal and from the ones with a few EEG signal. The purpose of this classification of data files was to create subsets of data for training and testing process with heterogeneous EEG information to design the model. Thus, since the model learns from data containing missing values and different scenarios, it will generate a more suitable response to similar data points in the testing phase. This approach resulted in a total of twenty-nine files initially reserved for testing and thirty-tree files to train.

Regardless of the amount of EEG signal present, all the files containing artefact information in the EEG signal, i.e. values close to 1×10^6, were distinguished to be pre-processed to smooth such values. These files were used in the first models trained, but due to the size of each file already existing in the train dataset they were not used in later models to avoid a higher computational cost. Besides, the files containing a very short amount of EEG signal were ignored.

The signals collected from each patient can be visualised in ICM+, the intensive care monitoring software, at the bedside in real time. The software developed by the Clinical Neuroscience Department of the University of Cambridge provides a useful tool for the continuous monitoring and clinical analyses in online mode. Moreover, it can also be used in offline mode for later review analysis and research use of raw data recorded.

Each ICM+ data file was exported from ICM+ in comma-separated values format, also known as csv format. This format is easily imported from the selected software to develop the algorithm, which is MATLAB software; in contrast to what happens with the original raw data files. Each csv file obtained contained thirty-one columns of data that can be pre-processed and used in training and testing processes. The first column of values of csv files shows the date and time of the signals collected as the number of days combined with the number of seconds after 31/12/1899 [40]. It can be converted to the conventional format using excel functions. Besides, it has been noticed that there is an interval of one minute between two consequent instances present in each csv file. For this reason, the frequency of each signal corresponds to one instance per minute (0.02 Hz).

The first step involving the data exported from ICM+ consisted in the representation of the data of a patient in MATLAB to ensure that there was no loss of information. Therefore, the signals available in ICM+ were represented in MATLAB and the obtained plot was highly compared. Figure 3.1 (a) shows some of the monitored signals displayed in ICM+, and the respective plot using MATLAB is available in Figure 3.1 (b). As it can be observed, the shape of the signals was preserved, and the analysis demonstrates the absence of information lost.

3.2 - Experimental Conditions

The selected machine learning methods to be applied in MATLAB only accept a unique file with all the input data to train the model. For each built model, the prediction was performed to a unique file containing the data of several patients. The approach to create the file with all the data for the training process consisted in the combination of various files into a unique one. The MATLAB script can read the specified set of csv files to be merged and combine them considering only ten of the thirty-one signals exported from ICM+. This data reduction resulted in a lower computational time and hence computational cost. To simplify, the variable that contains the information about the date and time of each instance, that will be further designated as *dateTime* variable, was converted to a sequence of the nonzero natural numbers, as a count of instances, because although the evolution of time is important to assess patient's health, the specific time and the date of each record can be rejected.

The reduction for the 10 variables also allows a signals equalization, so that the selected signals to be involved in the training process were added to the file with the same order. The selected signals

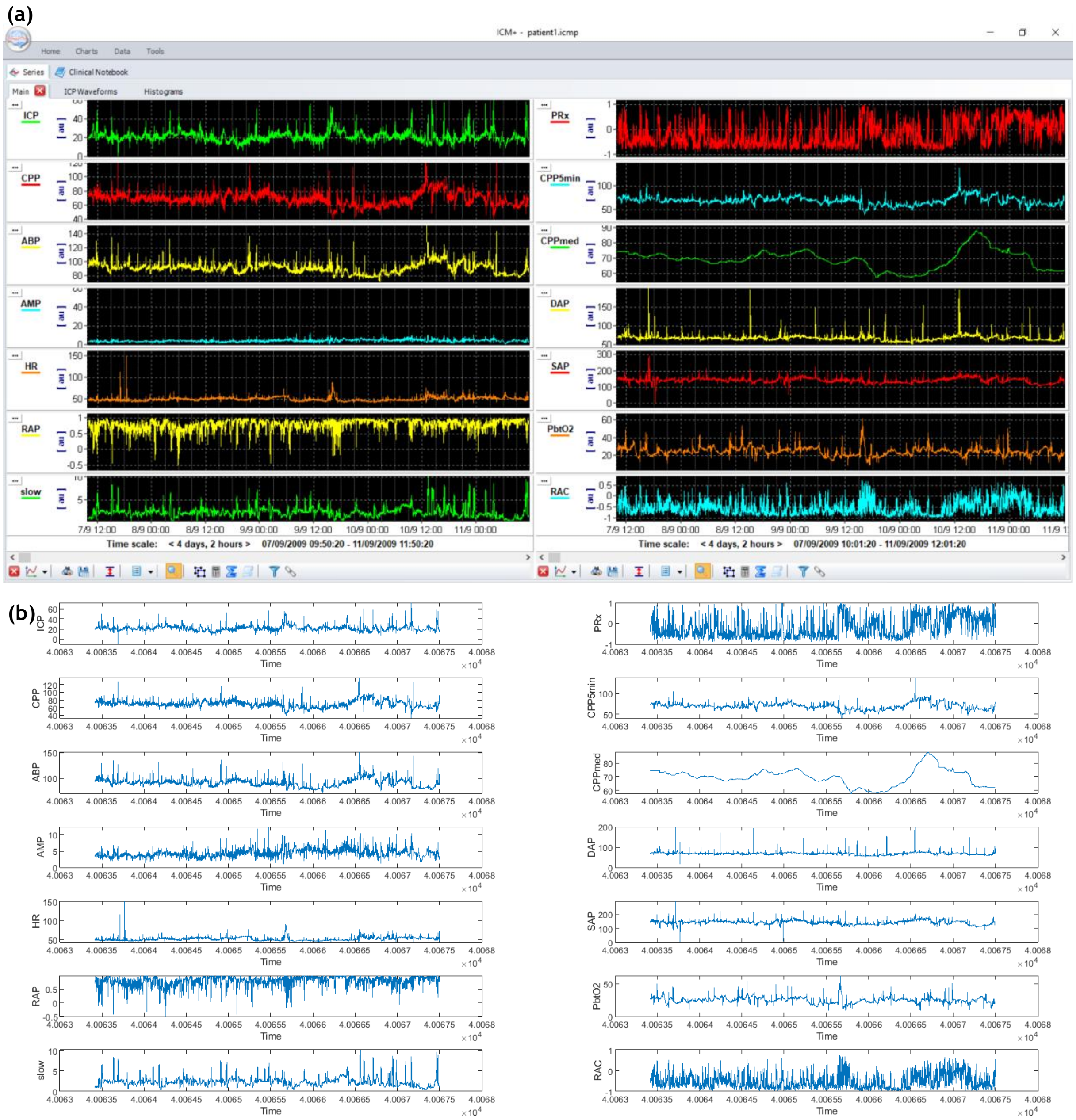


Figure 3.1- (a) – Visualisation of several monitored signals (ICP, CPP, ABP, AMP, HR, RAP, slow, PRx, CPP5min, CPPmed, DAP, SAP, pbtO$_2$ and RAC) of a patient in ICM+. (b) Representation of the same monitored signals of the same patient in MATLAB.

and variables for this first approach of train file were *dateTime*, ICP, CPP, ABP AMP, EEG, RAC, PRx and HR. The units presented in the original file and associated with the header of each signal were discarded in all posterior versions of files.

To train the machine learning methods, and as response to the emerging needs, different files to train and test were being created. The first file for training combined the information of the thirty-tree patients. Besides, the total of twenty-eight files allocated to test created the test data. In both, the files merged without any kind of separation between two subsequent files, and the lines containing missing values were removed.

In order to include the EEG features computed to create models that can scrutinise the influence of EEG signal before the beginning of a plateau wave, the features were added to the existing file, creating a second training file. Another change occurred in labels used to train and test, because the prediction considered ICP and ABP variations instead of the changes observed in ICP and CPP signals. The second testing file included the information of the twenty-eight patient files, as the first testing file, with the merge of the computed EEG features.

Further, a third training file was generated, because there are changes in the behaviour of the signals from patient to patient and a signal may end with a lower value in a patient and start with a high value in the file of the next patient. If these files were merged, there would be a sudden transition with none explanation that could confuse the learning process of the model. Hence, between the information of two patient files, five lines of data were added, that can be interpreted as five minutes of recording considering the acquisition frequency. In these lines, only the *dateTime* variable did not contain missing values and continued the time sequence, since the purpose was to reproduce the idea of time evolution with missing information to substantiate the mentioned changes. Moreover, this new file only contains five of the original thirty-one columns, because it was applied in the set of machine learning methods focused to evaluate the existence of a relationship between the EEG signal and the beginning of plateau waves. Therefore, it included *dateTime* variable, ICP, ABP and EEG signals, and the total power of the EEG signal.

This last version merged thirty-three files as in the first version. However, to reduce the computational cost and to transform the models into feasible training models, it was reduced to ten randomly selected patient information files. In the case of the test file, it is composed by the five mentioned signals of the twenty-eight patient files the data of the twenty-eight patients selected for testing. Moreover, the five lines introduced between two patient files were inserted and the missing values were not eliminated.

The missing values existing in the ICM+ data file, represented as *NaN* values, can be originated due to an accidental disconnection, such the disconnection of a sensor, or an intentional disconnection when the patients have a medical examination. Alternatively, it can be related with the capacity of ICM+ to save data that stops after a certain size of data has been collected, electrical interferences or movements of the patient. These missing values can be found in one or more signals at the same time and they were maintained as data because their elimination would represent a loss of information that could result in an abrupt change of signal values in two consecutive instances. However, these missing values were not considered for the calculation of means and standard deviations.

3.3 - Algorithm for Automatic Detection of Plateau Waves

Before the design of the algorithm, it is important to understand the evolution of ICP signal before, during and after a plateau wave. As mentioned in section 2.1.2, ICP signal varies between 0 and 15 mmHg, whereas values above 40 mmHg may reveal the presence of a plateau wave. Although the occurrence is the focus, it is important to understand the typical delineation of the mentioned wave

(Figure 3.2). Besides the period marked by the increased ICP values, the mean of ICP for the 30 minutes before the occurrence, characterized by values below 20 mmHg, is designated as baseline. On the other hand, the mean values of the first 30 minutes after the decrease of ICP establishes the 30 minutes after A-wave. If the second 30 minutes after the occurrence of the plateau wave are considered, it composes the 60 minutes after A-wave. The oscillations that occur before and after the episode, characterised by the variation between 20 and 40 mmHg, correspond to a change in the stationarity of the ICP signal. Thus, before plateau waves, this change has an exponential behaviour, while after the events the slope of the change is slower.

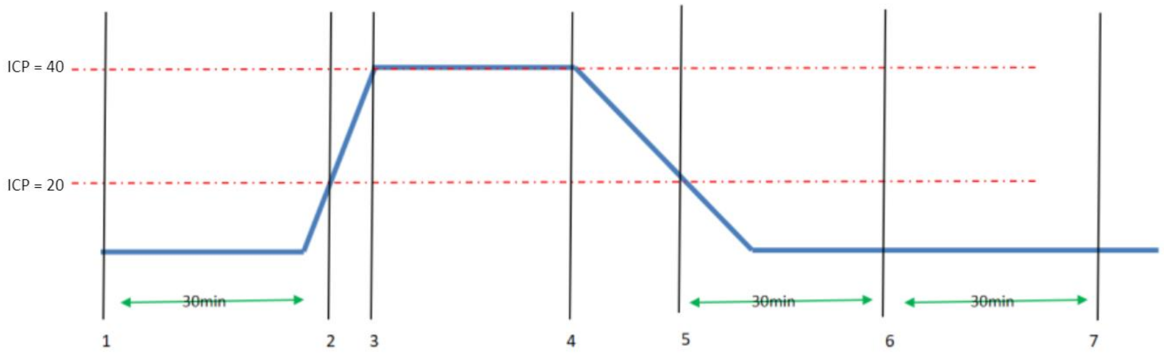


Figure 3.2 - Typical plateau wave delineation. The ICP values below 20 mmHg define the baseline of the ICP, represented in 1-2, 5-6 and 6-7 intervals. The periods of 30 minutes existing before and after the event are used to calculate the baseline. Plateau waves are commonly defined as the rises in ICP to value above 40 mmHg. The zones between the plateau and baseline values represent a change in the stationarity of the signal, which is associated with an exponential increase before the plateau wave (2-3 interval), or a slower variation in the decrease verified after the episode (4-5 interval). The two 30 minutes baseline verified after the occurrence of the plateau wave.

There are two main changes in the monitored signals that can be observed during a plateau wave – an increasing in ICP to values above 40 mmHg and a relevant decrease in CPP values. During the all period of the plateau waves phenomena ABP remains almost constanst. Thereby, a function to automatically detect plateau waves in patient files using offline data was developed. The labels computed trained and tested the accuracy of several machine learning models developed. However, the algorithm had not a good performance, since some of the instances were misclassified as plateau waves. This was observed in an analysis performed to the labels of a specific file, in which it was noticed that CPP signal variations were the origin of several false positives.

Therefore, since in plateau waves the increase of ICP is accomplished with a decrease of CPP signal and ABP can be defined as the sum of ICP with CPP (equation 3.1), a small variation in ABP signal can be considered for the detection of the episodes instead of the CPP decrease.

$$ABP = ICP + CPP \tag{3.1}$$

In this way, the final algorithm for the automatic detection of plateau waves creates a set with all the instances where an increase of ICP above 40 mmHg values was maintained for a minimum period of two minutes, represented as two data points. As result, several groups of instances revealing potential plateau waves have been identified over time. Moreover, the standard deviation of each group of potential instances was computed, and they were classified as a plateau wave if its value was lower than the standard deviation of the ABP signal.

3.4 - EEG Features Extraction

The study of the influence of EEG signal in the beginning of the plateau waves required the extraction of some EEG features, namely its amplitude, the amplitude difference established between the peak of the plateau wave and the basal values observed immediately before and after the episode. Besides, a frequency analysis of the signal was performed and the ratio between alpha and theta brain waves calculated.

The amplitude of the signal resulted from the absolute value of the difference established between each instance and the mean value of the entire signal. The values were inserted in a vector with the same size that the number of instances presented in EEG signal. The analysis of the difference between peak and basal values was concentrated in the instances classified as plateau waves, starting two instances before its beginning and finishing after two instances. Between the first and the penultimate instances, the difference in the amplitude between each data point and the next is calculated and added to a vector. The instances not involved in these calculations had their values set to 0, because the purpose was to study the variations that would occur near to the plateau waves, and thus this value did not represent variation.

To calculate the ratio between alpha and theta waves, the original signal was decomposed into five wavelets — noisy, gamma, beta, alpha and delta —, using the *wavedec* and *wrcoef* functions [41]. The division resulting from the alpha and theta wavelets computed the ratio.

The fast Fourier transform method converts the signal representation in the time domain to the frequency domain. Consequently, to extract some frequency features, the EEG signal was converted to the frequency domain and its phase and magnitude extracted.

These features were calculated for the EEG signal of a specific file, that could include the data of one or more patients. This information contained in the file was imported to an array and the EEG features attached to its end. The final array corresponded to the input data used to train or to test some of the machine learning methods applied.

3.5 - Machine Learning Techniques Applied

3.5.1 - Unsupervised Learning Algorithms

Since the provided data was not labelled to identify the plateau waves, the first approach applied to the data consisted in the training of unsupervised learning techniques to try to categorize and label the data as a plateau wave instance or not.

Although PCA is widely used for dimensionality reduction of the feature space, it is an unsupervised learning technique. For this reason, it trained using the first version of training file. However, since this method replaced the representation of data in x-axis for the principal axis of variation, the results as classifiers are inconclusive. Besides, due to the 3D scatter property the representation was focused on the three variables with most influence in the outcome, discarding the other.

A new attempt to train unsupervised learning techniques caused the implementation of k-means. This method, defined by *kmeans* function in MATLAB [42], only requires the input data and the definition of the number of clusters that can be interpreted as the number of classes to group the data, defined as 2 for this application. Four models were trained as the results of the application of one additional parameter in the function, which is the distance metric [42]. It can be the squared Euclidean, set as the default value, correlation, cosine and cityblock. There is other available metric, the hamming distance, that was not computed since its application resulted in an error. The input data corresponded to the data contained in the first training file created.

However, the computed labels presented an unexpected behaviour since the detection of the plateau waves was concentrated in the intermediate data points and the expected result consisted in a more distributed presence of plateau waves over time, since the file contained information of several patient with various plateau waves.

3.5.2 - Supervised Learning Algorithms

Once the application of the unsupervised learning techniques did not result in satisfying models for the classification of the data as plateau wave or not, the need for the development of an algorithm capable to automatically detect the plateau waves has arisen. As solution, a first algorithm was developed and later improved to its final version, as mentioned in section 3.3. The computed labels operated as real labels to train the supervised learning methods applied or to test their accuracy. The initial purpose of the application of these methods was to develop an algorithm that could predict plateau waves at the beginning. For that, the data of the first and second training files were imported as input data. Later, the purpose changed to evaluate the existence of a possible relationship between the EEG signal and the beginning of a plateau wave. In this case, the models used the third training file with the EEG features as input data.

3.5.2.1 - Logistic Regression

Linear and logistic regressions are techniques that place a hyperplane to separate data point into classes. Although the linear regression showed a good performance in Weka software, the logistic regression can design non-linear boundaries for two-dimensional space problems for a better separation of data. Thus, its output assumes any value between 0 and 1 since it represents the probability of the occurrence of an event based on a logistic sigmoid function.

This model trained only once using the data of ICP and CCP signal imported from the first training file created as input data. Being a supervised learning technique, the first version of the algorithm for the automatic detection of plateau waves computed the label. The cost function and the gradient were calculated and used as parameters to compute the optimization function. At the end, using the developed model, the label of the data imported from the first testing file and the model accuracy were calculated.

3.5.2.2 – Long-Short Term Memory (LSTM) Model

Long-short term memory (LSTM) is a type of recurrent neural network convenient for the study of time series and sequences. The algorithm learns from the dependencies existing between the time instances of the series or sequence. If the information used to train is in the forward direction, it is a LSTM network. Otherwise, if the information in both directions is simultaneously considered, the model is designated as bidirectional LSTM network.

In MATLAB, the *trainNetwork* function can be used to train an LSTM model, a bidirectional LSTM model or a neural network [43]. As parameters, the models require a cell array with a time series or a sequence, the label of that data, and the definition of the options and the layers of the network. The type of the label varies according to the problem. In case of classification problems, the output can be a categorical vector or a cell array of categorical sequence, while the output of regression problems can be a matrix of targets or a cell array of numeric sequence. Since this is a classification problem, the label was provided to the model as a categorical vector. The training options, such the maximum number of epochs, the initial learning rate and the gradient threshold, and the specifications of the network layers, including its architecture and the number of hidden units, are determined with options and layer [44] parameters, respectively.

The time series was imported from the first training file created and its responses was calculated using the first version of the algorithm to the automatic detection plateau waves. To create the cell array with the input data, the transpose matrix containing the data imported from the training file was calculated, so that each row represented one feature and not an instance as in time series of the original matrix. In layers, 10 and 100 were defined as different numbers of hidden units to train.

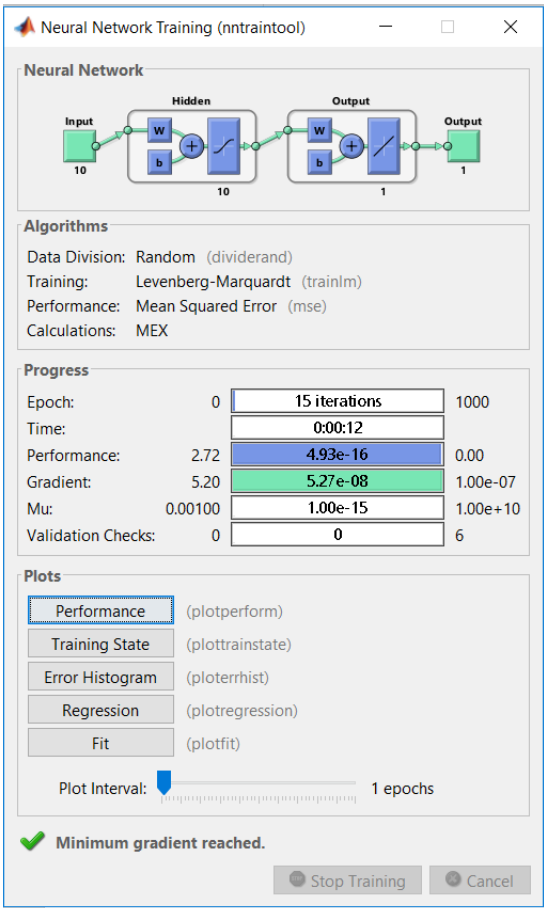
3.5.2.3 - Neural Networks Tool

MATLAB has a toolbox to train and test neural network models for different applications, that can be launched by *nnstart* command line [45]. When it is started, there is a panel containing four types of solving problems where neural networks work well, which are fitting tool, pattern recognition tool, clustering tool and the time series tool. Alternatively, each tool can be launched using *nftool*, *nprtool*, *nctool* and *ntstool* command lines, respectively.

Regarding the fitting tool, after the selection of the input data and the targets, a panel is presented to select the percentage of data that must be used for the validation and test phases. A fixed value of 70 is defined as the percentage of the input data to be used to train the model. However, this data partition is randomly executed [46], which means that the concept of the evolution of the signals over time is missed. This can interfere negatively in the learning process because the ICP values of plateau waves are mixed with the baseline values and the changes that may occur in the different monitored signals before the triggering of a plateau wave are not measured. The following panel allows the definition of the number of hidden neurons that must be considered in the learning process. It is important to emphasise that the higher number of hidden units requires a higher computational cost, since more convolutional operations are performed to produce intermediate output as a result of the weighted inputs. After that, the model can be trained using Levenberg-Marquardt, Bayesian Regularization and Scaled Conjugate Gradient backpropagations as training algorithms. The Levenberg-Marquardt backpropagation algorithm was designed to approach second-order training speed without the computation of a matrix of second-order partial derivatives of a function, designated as Hessian matrix [47]. The Bayesian Regularization backpropagation minimizes the linear combination of squared errors and weights. The weight and the bias values are updated according to Levenberg-Marquardt optimization [48]. The Scaled Conjugate Gradient backpropagation updates the weight and the bias values using the scaled conjugate gradient method, which applies the Levenberg-Marquardt in step size scaling [49, 50].

During the training, a window with the neural networks details appears (Figure 3.3). This window contains the summary of the training method, containing information regarding the network accuracy, the number of iterations performed, and the time elapsed until the training phase is completed. Besides, it provides useful plots, such the performance plot that presents the evolution of the performance measure over time. The fit plot shows how the network fits in the data throughout a representation of the error over the several input data points.

If the results in the training are not satisfactory, the network can be retrained using the same initial conditions, resulting in a different performance accuracy. The network and the labels of the test set resulting from this training can be exported and saved in the workspace of MATLAB. Furthermore, the network code can be edited to be more customized to the current application. These features related to the saving and editing of the networks are available in the four tools.



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	15 iterations	1000
Time:		0:00:12	
Performance:	2.72	4.93e-16	0.00
Gradient:	5.20	5.27e-08	1.00e-07
Mu:	0.00100	1.00e-15	1.00e+10
Validation Checks:	0	0	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Fit (plotfit)

Plot Interval: 1 epochs

✓ Minimum gradient reached.

Figure 3.3 – Summary window of the training process using the neural networks tool. Between the information available, there is the definition of the performance evaluator, the type division performed in data and the selected training algorithm.

The pattern recognition tool is suitable for pattern recognition and classification problems. Similarly to the previous tool, the input data and the labels must be imported, followed by the definition of the percentage of the input data for validation and test, with the training value fixed at 70%, and the definition of the number of hidden units. However, this tool only uses the Scaled Conjugate Gradient backpropagation. During the training process, a window is displayed with a summary of the process, as in the previous tool.

Regarding the clustering tool, it is suitable for feature extraction and data dimensionality reduction. Only the input data is provided, and thus it can be considered as an unsupervised learning method. The input data should not contain missing values, and the number of the hidden units is replaced by the number of neurons. After that, the model can train using a batch self-organizing map algorithm [51]. This algorithm groups the data accordingly to the way that they are grouped in the input space and it learns from the distribution and the topology of the input vectors.

Finally, the time series tool solves nonlinear time series problems using dynamic neural networks, since it uses past prediction to compute the new. Three types of nonlinear series problems can be solved using this tool. The first type is the nonlinear autoregressive with external (exogeneous) input (NARX) where the prediction is calculated based on some past predicted values and in the time series.

The second kind is the nonlinear autoregressive (NAR) problems, where the prediction is only based on a specified number of the last predictions. The third problem is the nonlinear input-output that only considers the series to compute the prediction, and thus it does not consider any predicted value [52].

While the NARX and the nonlinear input-output require the input data and the labels, the NAR only requires the targets, since the new predictions are based on some of the previous computed predictors. However, the three approaches require the selection of percentages for validation and testing, the number of hidden neurons and the number of delays. Besides, Levenberg-Marquardt, Bayesian Regularization and Scaled Conjugate Gradient are available training algorithm for the three approaches. The time delay concept in this context represents the number of input data that feds the neural network in time dimension, instead of the spatial dimension observed in most of the machine learning processes.

Due to the randomly split of data, training multiple times will generate different results since the initial sampling is different. Three sets of input data and the two labels of plateau waves were applied.

Regarding the fitting tool, the first training file created two subsets of data: the first considered all the data imported from the file and the labels were computed using the first version of the plateau waves detection algorithm, while the second considered the same data except the ICP values that were implemented as target. The pattern recognition tool trained with the input data coming from the first training file and the labels computed from the first version of the algorithm for the automatic detection of plateau waves. These labels were also implemented in the clustering tool. However, the summary about the training process of the model was inconclusive, because the performance evaluator was absent. Regarding the time series tool, it was widely trained, because, in addition to the tree training algorithm and different number of delays trained, several pairs of input data and labels were also applied. To simplify the comprehension of the combinations trained using the time series tool, they are all summarized in Table 3.1. The designation of all data represents the data derived from the first training file, while the same data without the ICP values is designated as all data except ICP values. The computed labels designation represents the label computed using the first version of the algorithm for the automatic detection of plateau waves.

In addition, the second training file were also used to train models in time series app. Since, the size of the input data increased, the models were initially trained using only the information of three files randomly selected.

In time series tool, the NARX and the nonlinear input-output problems were applied with the three training algorithms. The number of hidden neurons defined as 10 and the number of delays set in 2. The NAR problem was not trained because it only considers the labels to predict new instances and the purpose of this study is to predict plateau waves based on the monitored signals.

Table 3.1 — Summary of the trained models in the time series application to solve NARX problems, using the first training file.

Training Algorithm	Input Data	Label of the Data	Number of delays
Levenberg-Marquardt	All data	Computed Labels	10
	All data except ICP values	ICP values	10
Bayesian Regularization	All data except ICP values	ICP values	5,10
	All data	Computed Labels	100
Scaled Conjugate Gradient	All data	Computed Labels	10, 50, 100
	All data except ICP values	ICP values	10, 50, 100

Moreover, using the information of the three patients, the fitting tool trained with the three training algorithms with 10 hidden units. Besides, the Levenberg-Marquardt also trained with 4 hidden units and the Scaled Conjugate Gradient with 100. The pattern recognition tool trained with 10 and 100 hidden units, and the clustering app only used the input data. However, the computational time was relatively lower, and all data were used later. Thus, using all patient files, the fitting tool and pattern recognition were trained with 10 hidden units and the time series tool trained with 10 hidden neurons and 2 time delays. In both cases, the labels were predicted based on the final version of the plateau waves detection algorithm.

This tool was not used with the final version of data where the focus is in the EEG features, since the division of the data using this toolbox is random [46]. Thus, the results that could be obtained would not be meaningful for the present study.

Alternatively to the tool, the *train* function can also be applied to train a neural network model and the computation of the data test labels using the trained model can be achieved using the *predict* function [53].

3.5.2.4 - Support vector Machine (SVM) Models

Considering the medical applications using SVM models, it seems to be a suitable method to be applied. To train it in MATLAB, with *fitcsvm* function [54], the input data and labels arrays must be provided. Optionally, other parameters can be defined, such as the kernel function and the name of classes to use in training.

Two groups of SVM models were trained. Both use the labels computed by the final version of plateau waves detection. However, the set of input data was different. While one set used the data available in the second training file, characterised by the presence of the 10 variables and the EEG features, the second group were trained using the third version of data, which contains only 5 signals collected from patients and the EEG features. Besides, each input data has been trained for all the available kernel function, which are gaussian, linear and polynomial kernels, as an additional parameter.

Alternatively to the mentioned function, there are other two designed for specific SVM problems. For binary classification problem for high-dimensional data, the *fitclinear* function can be used instead. In case of regression problems, there is the *fitrlinear* function for high-dimensional data and *fitrsvm* function for the other data sets.

3.5.2.5 - Gaussian Process Models

The gaussian process regression model are trained in MATLAB using the *fitrgp* function [55], which requires as parameters the input data and its labels. Optionally, there are other parameters that can be redefined. One of them is the kernel function responsible for defining the covariance function form to compute the gaussian process. Due to the variety that this parameter might produce in the results, all its values were trained. Besides the default kernel, *squaredexponential*, there are other nine functions: *exponential*, *matern 32*, *matern 52*, *rationalquadratic*, *ardexponential*, *ardsquaredexponential*, *ardmatern32*, *ardmatern52* and *ardrationalquadratic*. The purpose of this approach was to evaluate if one of the kernel functions would result in an algorithm with better performance. This method was the most trained because, besides the kernel functions, three sets of input data were trained.

The first training data imported the data available in the first training file. The labels of these data were calculated using the first version of the algorithm developed for the automatic detection of plateau waves. The second input data picked up the data coming from the second training file. In this case, the labels were computed through the final algorithm for the automatic detection of plateau waves. Finally, the third input data set contained the data of the third training file and its labels were also computed using the final version of the algorithm for plateau waves detection. Since, the integration of the EEG features increases the computational cost of the models, the second and third groups of models only considers the information of four and ten patient files, respectively. Even so, the models still require a medium to high computational time.

Chapter 4

Results and Discussion

After the training of each machine learning model presented and explained in the previous chapter, the labels of the test data were computed. These labels predicted by the models were compared with the labels obtained using the two versions of the algorithm for the automatic detection of plateau waves, in different situations. The purpose of the comparison relies on the calculation of the accuracy of each model, since this is one of the most important performance measurements of a model. However, some models present their mean squared error instead of accuracy value. Hence, based on the built models, the performance evaluators were calculated to conclude which models may have a better behaviour for the current application. In addition, for a set of developed models, their confusion matrices were calculated for a better analysis of the predictions performed by those models.

To accomplish the results, a discussion will be presented to perceive the reasons that may cause those results, since the input data, the labels, the parameters defined for each model and the working principle of the different machine learning technique applied can be in their origin.

4.1 - Machine Learning Techniques Applied

4.1.1- Methodology Results

As mentioned in the previous chapter, three training files were created to train several machine learning methods. Associated with each training file, a testing file was also created with the same conditions. Besides the same variables were presented in the same order, the existence or absence of the missing values and the EEG features depended on the structure of the training file. However, the data used to test consisted in a new dataset unseen by the models.

Using each of the developed models, the prediction of the new labels was performed with the model and the test data as input arguments. After the computation of the predictions, the accuracy of the model was calculated based on the labels calculated using one of the versions of the algorithm for the automatic detection of plateau waves, which was considered in each application as the ground truth, i.e. as a real classification of data points.

Since it is a binary classification problem, the data points can assume one of the two available values for labelling, which are 0 and 1 that correspond to the absence and presence of plateau wave, respectively. Therefore, the label that will be referred as class 0 identifies the instances where there are no plateau waves, whereas the class or label 1 indicates the instances that reveal the presence of a plateau wave.

The accuracy is the ratio calculated between the quantity of instances well classified and the total number of data points. This was the most used formula to calculate the accuracy of the models; however, there are other evaluators.

The mean squared error, mainly used in neural networks tool, corresponds to the average squared difference between the outputs and targets. For this reason, lower values indicate better performance of the models, and a value equal to 0 means the absence of errors. The cross-entropy measurement presents a similar behaviour. Thus, lower values indicate good classification performed by the model. In some of training models, there is the regression value that measures the correlation between the outputs and the targets. If its value is 1, it indicates that there is a close relationship. Instead, a value of 0 indicates a random relationship. The mentioned outputs correspond to the labels predicted by the training models, while the targets correspond to the labels computed using one of the versions of the algorithm designed for the automatic detection of plateau waves.

The confusion matrix is also a common measurement to evaluate the performance of models and it can be obtained in MATLAB using *confusionmat* function [56]. To obtain the confusion matrix, it is necessary to provide the predicted and the real labels. As results, a summary of the existing classes and the confusion matrix are presented. The matrix contains the quantity of TP, TN, FP and FN instances [57]. The positive designation is related with the presence of an abnormality, that could be an event or pathology, whereas the term negative reveals the absence of the abnormality. On the other hand, the true term indicates that the instance was correctly predicted, while the false designation indicates the misclassification of the instance. In this case, TP represents the number of instances that were correctly predicted as plateau waves by the model and TN reveals the amount of instances where the absence of plateau waves was also predicted properly. Regarding FP, it indicates the number of instances incorrectly predicted as plateau waves instances. This means that the model predicted the instance with a label 1, but the label considered as ground truth was 0. Similarly, the FN number presents the quantity of instances not predicted as plateau waves when they are instances that present such episodes. Thus, the label coming from the ground truth is 1 and the model predicted as 0.

Using the results of the confusion matrix, it is possible to calculate the accuracy (equation 4.1) and obtain a value equal to the one obtained using the mentioned formula. In addition, other performance metrics can be evaluated, such as the sensitivity (equation 4.2), also designated as recall or true positive fraction, the specificity (equation 4.3), the precision (equation 4.4) and the false positive fraction (equation 4.5) [57].

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FN+FP}} \tag{4.1}$$

$$Sensitivity = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{4.2}$$

$$Specificity = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{4.3}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{4.4}$$

$$False\ Positive\ Fraction = 1 - Specifcity = \frac{\mathrm{FP}}{\mathrm{TN+FP}} \tag{4.5}$$

4.1.2- Compute Labels

After some machine learning techniques have trained with the labels computed by the first version of the algorithm developed for the automatic detection of plateau waves, it was noticed that their results were not satisfactory. Since those labels were not the real prediction, i.e. they were not classified by a physician in order to have a ground truth to be used to train and test the models, it was thought that the reason behind the unsatisfactory results could be in the automatic computation of the labels. For that reason, some patient had their labels computed and an analysis of the potential plateau waves in ICM+ was performed. The files were randomly selected and opened in ICM+ to observe the various signals in simultaneous and confirm or eliminate the presence of a plateau wave for the predicted instances. As result, some of the instances detected by the algorithm as plateau waves were false positive, which indicates the detection of instances as plateau waves higher than the number of real plateau waves. However, all true plateau waves were correctly identified.

The development of an improved version of the algorithm for the detection of the plateau waves arose. This new version results in a more accurate model for the detection of the plateau waves. Figure 4.1 shows the prediction computed by the algorithm and the physician identification are shown. Figure 4.1(a) display the ICP and ABP signals of a patient in ICM+ software. The red lines placed over the ICP signal identifies the beginning of the plateau waves identified by the physician. On the other hand, Figure 4.1(b) shows the detection of the plateau waves computed by the algorithm, since it contains the number of instances in the x-axis and the prediction of each one in y-axis. Apart from the fact that the images are not completely aligned, the plateau waves identified by the physician correspond to the plateau waves detected by the algorithm.

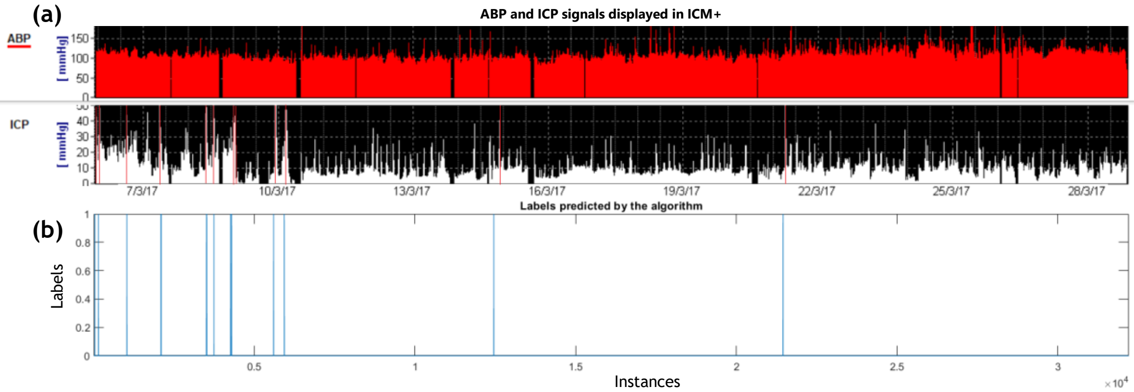


Figure 4.1 - (a) ABP and ICP signals of a patient displayed in ICM+ software. ICP signal contains red lines that represent the beginning of the plateau waves. (b) Representation of the plateau waves detected by the final version of the algorithm for the automatic detection of the plateau waves in MATLAB.

Thus, the algorithm shows a good performance and it seems to be a suitable tool for the automatic detection of plateau waves.

4.1.3- Unsupervised Learning Algorithms

As already mentioned in section 3.5.1, PCA model produce inconclusive results, because this machine learning technique transforms the representation of the data from the original axis to the principal axis of variation, and therefore the results are not appropriate as classifiers. For that reason, this method was only used with the first training file and not in posterior models as it happens with some supervised learning techniques.

Regarding the k-means models, four models were trained as the result of the application of four available distance metrics. These models use the information coming from the first training file as input data with the number of classes to classify the data points defined as two. Since it is an unsupervised learning technique, the labels do not have to be provided to supervise the learning process. However, they were computed using the first version of the algorithm for the detection of the plateau waves. The purpose of this computation was to obtain labels to be considered as the real labelling of the data to calculate the accuracy, because each developed k-means model predicted the instances for the training data. Therefore, the accuracy presented the amount of instances in the training data whose label of the algorithm and the model matched.

To assess whether the accuracy values were biased due to the employment of the same data in training and testing processes of the models, the data of the first testing file created was tested in each of the four developed models. Therefore, eight accuracy values were calculated, and they are presented in Table 4.1. The accuracy values vary between 45% and 98%. As it can be noticed, most of the models present better accuracy in the testing set of data than in training data. With this, it is possible to conclude that the values obtained in the training set were not biased. This could be a positive feature of the developed models.

Table 4.1 — Accuracy values obtained for the models trained with each distance metric.

Distance Metric	**File with the Input Data**	**Accuracy** (%)
Squared Euclidean	Training	46.0560
	Testing	83.3957
Cityblock	Training	57.1576
	Testing	58.7444
Cosine	Training	57.3380
	Testing	83.3858
Correlation	Training	98.0821
	Testing	83.3864

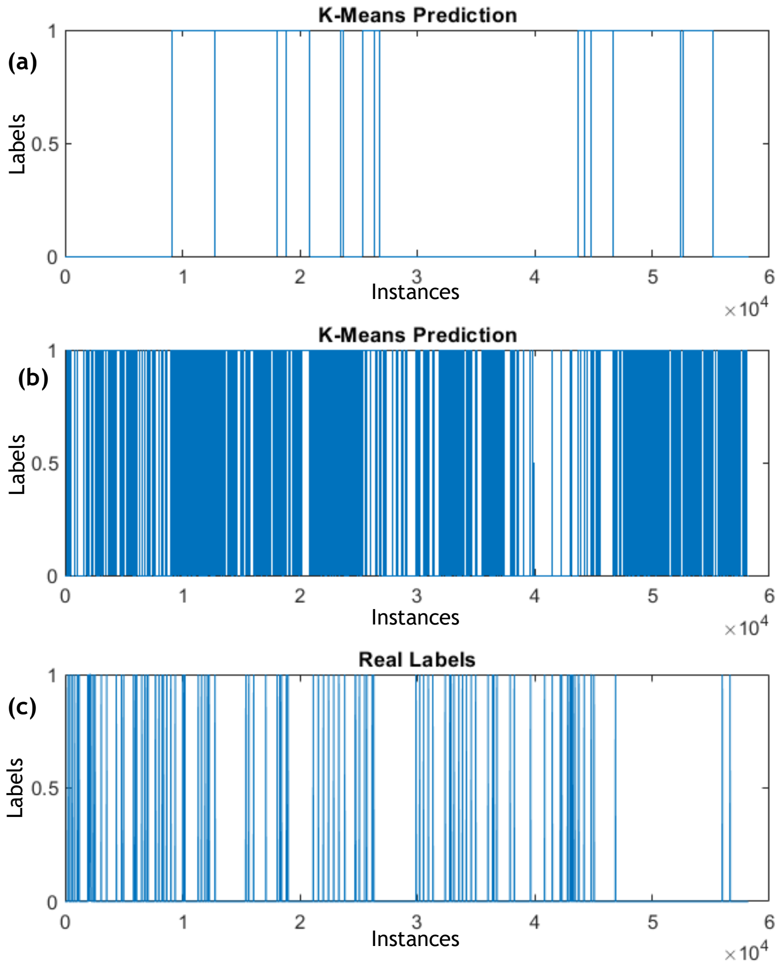


Figure 4.2 – (a)-(b) Labels predicted by the k-means models designed, using cityblock distance metric, in two subsequent iterations, and (c) represents the labels considered as real label to calculate the accuracy of the models.

However, when the models are trained multiple times, the accuracy values obtained are different. Therefore, the resulted presented for the training data were calculated when the figures were saved. Figure 4.2 represents this behaviour in two consecutive trainings using cityblock as distance metric and it also contains the labels that were considered as ground truth to compute the accuracy. As it is possible to notice, the two models resulted in predictions very different, which represents the diversity mentioned. The x-axis presents the number of instances, while the y-axis represents the prediction of each sample, which corresponds to the configuration defined for all the plots that shows the prediction of models. The prediction present in Figure 4.2 (a) has the accuracy value of 57%, whereas 47% is the accuracy value associated with the predicted observed in Figure 4.2 (b). In addition, Figure 4.3 (a) represents the computation of the labels with the model that presented better accuracy using the training data, which corresponds to the application of correlation as distance metric. Although the elevated accuracy value, there are a lot of plateau waves that are not detected. Besides, some prediction that should belong to the class 0 where classified as members of the class 1. Conseqeuntly, it is possible to conclude that the models are not suitable to build a model to predict plateau waves because of the diversity of models originated in different iterations and the misclassification of several points.

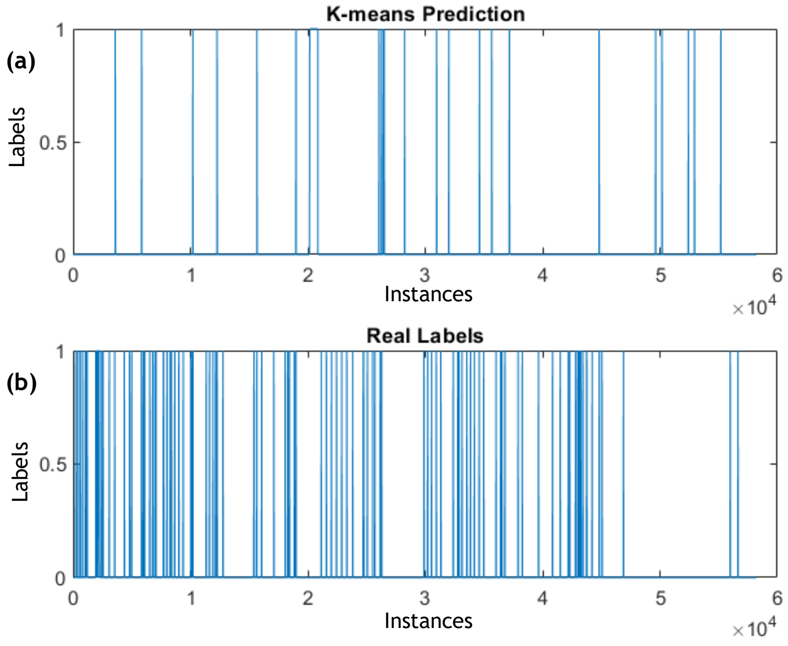


Figure 4.3 – (a) Labels predicted by the k-means model trained with the training data and correlation as the distance metric. (b) Labels computed using the first version of the algorithm for the detection of plateau waves, considered as real labels.

4.1.4 - Supervised Learning Algorithms

4.1.4.1 – Logistic Regression

As mentioned, the logistic regression method was trained only once with the first training file. The developed model predicted the instances of the first testing file. The resulting predictions were compared with the ones computed using the first version of the algorithm developed for the automatic detection of plateau waves. Therefore, the accuracy value was 99.3848%, which corresponds to a high value. Figure 4.4 presents the plot resulting from the predicted labels and the ones considered as real. As it can be observed by the similarities between the two plots, most of the plateau wave instances were correctly predicted, although some of them were not detected by the logistic regression model. However, this method was not applied in the posterior training and testing files, due to the increase of the computational cost caused by the increase in the input data size. Besides, the final training file contains missing values that are not supported in this supervised learning technique.

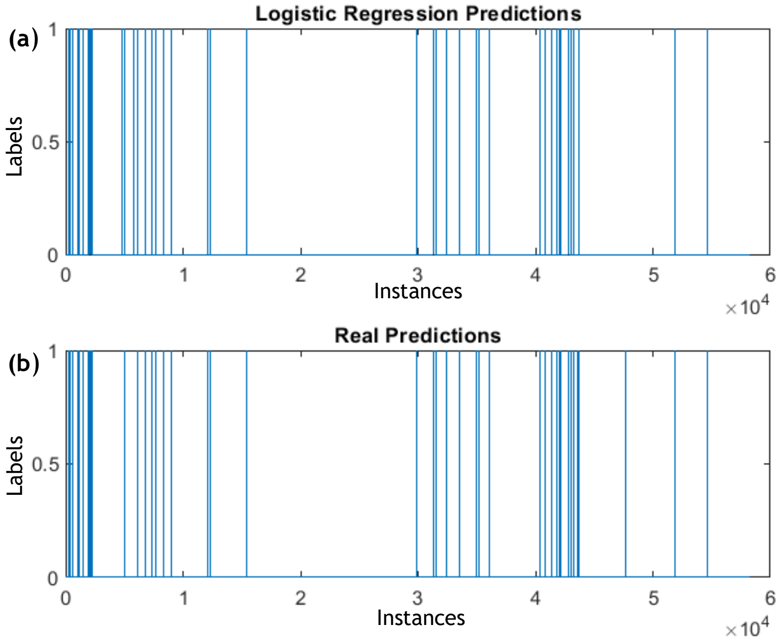


Figure 4.4 – Representation of the labels computed with the logistic regression model developed, and the labels computed by the first version of the algorithm for the detection of plateau waves, considered as the real labels.

4.1.4.2 – Long-Short Term Memory (LSTM)

Regarding LSTM method, two models were developed as result of the two different numbers of hidden units trained. The network with 100 hidden units took more than 130 minutes to compute 496 iterations, whereas the network with 10 hidden units required 15 minutes to compute the same number of iterations and the 36 minutes to compute the 1000 iterations that corresponds to the maximum number of iterations.

In both cases, the developed models predicted the labels for the first training file, which were compared with the ones obtained using the first version of the algorithm for the automatic detection of plateau waves. The calculation of accuracy for both models resulted in a value between 50 and 60%, and the loss function rounds the value of 0.7. Although the labels were not the correct ones, these models did not show a good performance. For that reason they were discarded to train with the other training files created later, because just almost half of the instances were correctly predicted. This revealed a bad performance for the prediction of plateau waves.

4.1.4.3 - Neural Networks Tool

Regarding neural networks tool, a set of models were trained using the first training file and another set using the second training file. Since there is a part dedicated to the definition of the percentages that must be used to partition the input data into training, validation and testing subsets, the first and second testing file were not used in this tool. The labels for the first training file were computed using the first version of the algorithm for the automatic detection of plateau waves, whereas the labels for the second file were computed with the final version. The performance evaluator used in this tool is the mean squared error, where a lower value represents a better model performance.

Using the first training file, the performance of the models trained, validated and tested in fitting and pattern recognition tools are presented in Table 4.2. The performance of the clustering model is not presented, because the results of the training were inconclusive. As mentioned, some models were trained with all the data contained in the training file, while other ignored the ICP values in the input data and used it as labels of the data to evaluate if the other signals could be predictor of the ICP values, and thus with the presence of plateau waves. The models trained with the entire file exhibit a better performance value, which can be due to importance that ICP values have in the prediction of plateau waves. In addition, those models require less computational time. To illustrate it, the Bayesian Regularization training algorithm takes less than 4 minutes to compute 1000 iterations, that corresponds to the maximum number of iterations, while the same computation using the input data that ignored the ICP values took longer than four minutes. So, the application of ICP as target instead of an additional signal in the input data is not a good approach for the first goal of this study, at least with these experimental conditions.

Regarding time series results obtained from NARX problem solving, presented in Table 4.3, the increased number of hidden units applied for the same input data exhibits an improvement of the mean squared error value, since this value decreases. It is an expected behaviour, because the higher number of hidden units requires more intermediate computations between the input and the output layers, and thus the outcome is more adaptable. Once again, the models in which ICP data were used as label had worst overall performance results, and the computational time is higher, even if the difference is small.

Table 4.2 — Performance of several models trained with different input data in Fitting and Pattern Recognition Tools.

Neural Network Application	Training Algorithm	Input Data contains ICP values	Mean Squared Error
Fitting Tool	Levenberg-Marquardt	Yes	0.00111
		No	0.0293
	Bayesian Regularization	Yes	0.00105
		No	0.0271
	Scaled Conjugate Gradient	Yes	0.00393
		No	0.223
Pattern Recognition Tool	Scaled Conjugate Gradient	Yes	0.00307

Table 4.3 — Performance values of the models trained in Time Series Tool to solve NARX problems, according to the input data and number of delays.

Training Algorithm	Input Data contains ICP values	Number of delays	Mean Squared Error
Levenberg-Marquardt	Yes	10	0.00297
	No	10	2.75
Bayesian Regularization	No	5	2.41
	No	10	2.20
	Yes	100	0.00364
Scaled Conjugate Gradient	Yes	10	0.00414
	Yes	50	0.00417
	Yes	100	0.00405
	No	10	3.01
	No	50	4.82
	No	100	3.05

In this context, it is possible to assert that the approach of use the ICP values as prediction, in order to try to predict the future ICP values based on the changes and the values observed in the other signals is not a good methodology. Thus, the ICP signals must be always used to train the different models. Since this data approach was applied as one of the first approaches to develop a model, as well as the k-means method and logistic regression, it has not been further trained or tested in this tool, SVM or gaussian process models.

Due to the insignificance improvement in the performance as result of the increased number of hidden units, the second training file trained with few variations in the number of hidden units. The labels used were computed by the final version of the algorithm for the detection of plateau waves. As initial approach, some models trained with the information imported from only three patient files, randomly selected. The purpose was to reduce the computational cost; however, it was verified that the addition of EEG features did not significantly increased the cost. Thus, the same models were training with all the patient data of the second training file. Both sets of data were trained for fitting, pattern recognition, clustering and time series tool.

The results regarding the fitting app with these experimental conditions can be consulted in Table 4.4. The application of the reduced subset of data results in higher performance, because the model had less heterogeneous information to compute and a more adjustable model is developed. Besides, the test data is very similar to the data used to train and validate. However, the full set of data results in good mean squared error values and they are very similar for Levenberg-Marquardt and Bayesian Regularization training algorithms.

Table 4.4 — Performance results obtained from the application of the different training algorithms in Fitting App.

Training Algorithm	Quantity of patient files used to train	Number of delays	Mean Squared Error
Levenberg-Marquardt	3	4	1.03×10^{-10}
	3	10	9.46×10^{-12}
	33	10	0.0560
Bayesian Regularization	3	10	5.08×10^{-13}
	33	10	0.0558
Scaled Conjugate Gradient	3	10	0.000515
	3	100	0.000534
	33	10	1.12

Regarding pattern recognition tool, the cross-entropy is the measure used to evaluate the performance of the models trained. Therefore, and as in the previous tool, training with the information of three patients and 10 hidden units results in a performance value of 0.000113, while 100 hidden units improves the value to 8.58×10^{-5}. Using all the data of the second training file and 10 hidden units, the cross-entropy value is 0.258.

As mentioned in section 3.5.2.3, eleven models were trained using time series tool, because, in addition to three types of problems and the training algorithms, two sets of data were provided to train. One of them uses all the information presented in the second training file, while the other uses the information of three patients presented there. The first set was not used in NAR problem, because it uses the past predictions to compute the new prediction, which is not the purpose of this study.

Table 4.5 shows the results computed from NARX and nonlinear input-output problems for the algorithms using all the data of the second training file. In general, the NARX approach has better mean squared error values. However, regarding training algorithms, Levenberg-Marquardt generated the models with higher performance. These results were obtained in a specific training iteration; thus, the same data and the number of hidden units and delays will probably change the definition of the best model. This is because the partition of data that originates different subsets in each iteration, which is an acceptable reason to reject the development of a model with this tool because the behaviour of the model computed in the next iteration is unknown, as it can be noticed in Figure 4.5. Although, the predictions of the developed models are similar between them and to the predictions computed by the final algorithm developed for the automatic detection of plateau waves, there are some instances that can be distinguished because they were not predicted.

Table 4.5 — Performance results obtained from the application of the different training algorithms in each type of problem available in the Time Series Tool.

Type of Problem	Training Algorithm	Mean Squared Error
NARX	Levenberg-Marquardt	0.271
	Bayesian Regularization	0.768
	Scaled Conjugate Gradient	0.152
Nonlinear input-output	Levenberg-Marquardt	0.212
	Bayesian Regularization	0.262
	Scaled Conjugate Gradient	1.69

As the results of the fitting app, the models trained with the small set of data (Appendix A – Table A.1) present better performance, because the model is more customized to the input data. Therefore, the computation of a new set of different instances can result in an unexpected outcome.

Since the data used to train, validate and test the models were randomly selected, this approach was not applied to be trained and tested with the third training and testing files created, respectively.

The changes that occur in all the signals over time are important for a correct prediction of future instances.

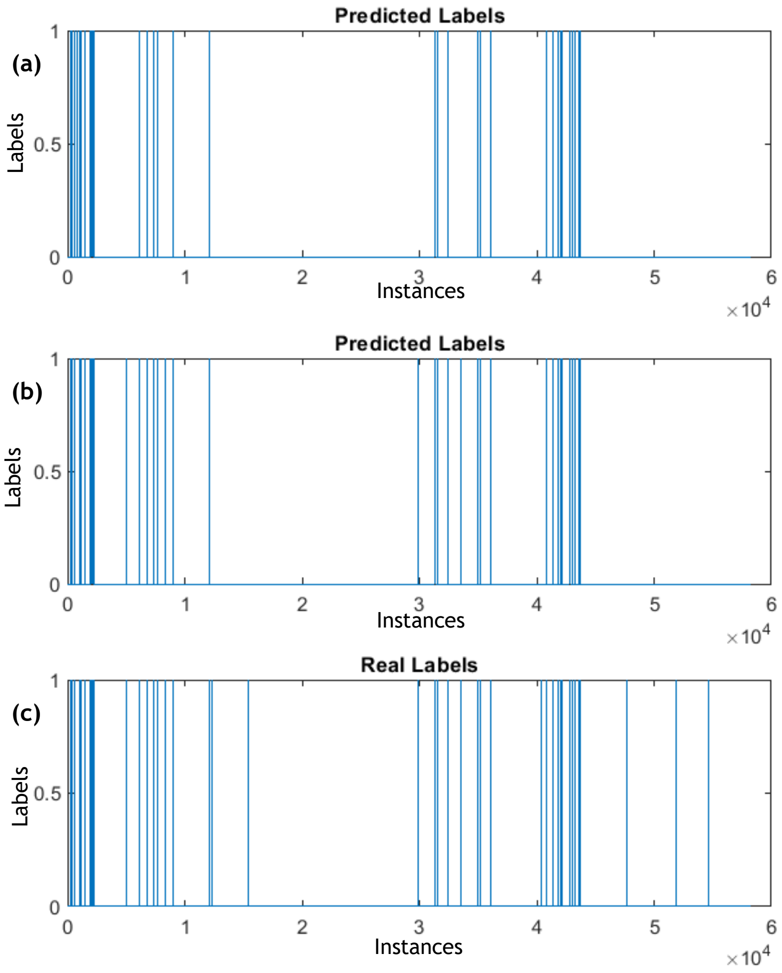


Figure 4.5 – (a)-(b) Labels predicted by models resulting from the time series app to solve NARX problem, using Levenberg-Marquardt as training algorithm. Both models were trained with the data contained in the second training file. (c) Labels computed using the final version of the algorithm for the detection of plateau waves, considered as real labels.

4.1.4.4 – Support Vector Machine (SVM)

The two training files and the three kernel function available resulted in the training of six SVM models. The labels computed using the developed models can assume any value between 0 and 1, instead of assuming only the 0 and 1 values as the labels obtained from the second version of the algorithm for the automatic detection of plateau waves, which was used in SVM models. For that reason, the labels were normalized to guarantee that they really were between 0 and 1, and then two approach were applied to classify each value as a member of the class 0 or class 1.

The first approach considers a threshold value that was manually selected. This must be a value that does not influence the detection of the plateau events nor the instances where plateau waves are not presented. Thus, the selected value was 0.5 since it is the mean value in the range for prediction and it seems appropriate to assume that the instances without the event present lower prediction values, while the plateau wave instances present the highest values. The second approach uses a value selected by the algorithm. Such value corresponded to the mean value of the predictions, which can be higher or lower than the previous threshold value, according to the amount of plateau waves that the file contains. The purpose was to evaluate if the threshold value was properly chosen or if could affect the results. However, all the three models for each training file result in the same accuracy. This means that the models trained with the gaussian, linear and polynomial kernel training and the data of the second training file had an accuracy of 99.8571%. This value reflects the quantity of instances correctly predicted as 0, because the models predicted all the samples as 0 (Figure 4.6).

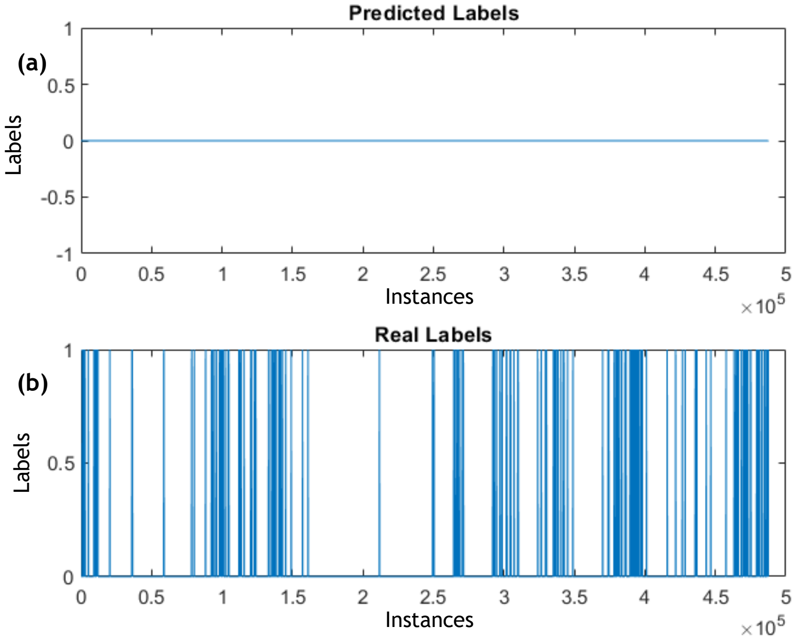


Figure 4.6 – (a) Labels predicted from the SVM model trained with the data of the second training file and gaussian kernel. These predictions were obtained after the recalculation of the labels to belong to class 0 or 1. (b) Labels computed from the final version of the algorithm for the detection of the plateau waves.

Besides, the accuracy value for the three kernel functions models trained with the third training file was equal to 99.9693%. Once again, all the instances were predicted as 0 by the developed models (Figure 4.7). Since it does not vary according to the kernel function, it is possible to assert that neither the kernel nor the approach used for the final assign did not influence the final predictions.

Although the prediction did not detect plateau waves, the accuracy values are elevated, because the classification of most of the total of 487648 instances using the second version of the algorithm for the automatic detection of plateau waves, used to compute the real labels of SVM models, were correctly classified as instances of class 0. Since the amount of instances related with plateau waves is reduced, the accuracy value is high.

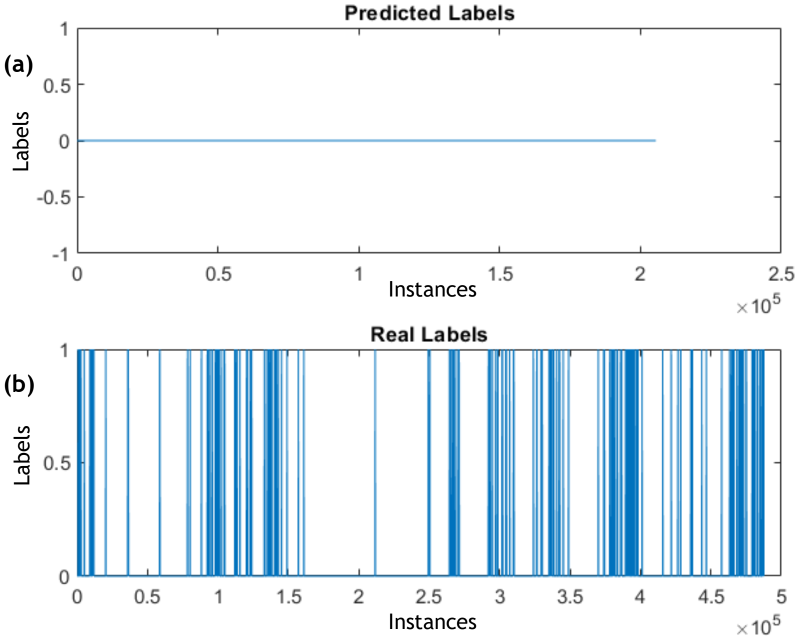


Figure 4.7 - (a) Labels predicted from the SVM model trained with the data of the third training file and gaussian kernel. These predictions were obtained after the recalculation of the labels to belong to class 0 or 1, which presents an equal behaviour for both approaches. (b) Labels computed form the final version of the algorithm for the detection of plateau waves.

4.1.4.5 – Gaussian Process Models

Due to the three training files and the ten available kernel function to define the covariance function form, a total of thirty models were trained. All the results can be observed in Appendix A – Table A.2. Since the results related with the first and second training and testing files were not the most important, a summary of them will be presented. Only the results of the models developed using the third training and testing with the third testing files will be completely presented, in addition to their confusion matrices.

Once again, the labels computed using the developed models assumed any value between 0 and 1. The two approach applied in SVM models to convert those labels for the values of 0 and 1 were also applied.

For the first training file, the models that used the threshold value of 0.5 for the classification of the labels present a better overall accuracy than the set of models that used the mean of the predictions as threshold value. However, the *rationalquadratic* and *ardrationalquadratic* are kernel functions whose models present similar accuracy values for the two mentioned approaches. In addition, they present the two highest accuracies using the threshold value for the final classification, which are 99.7321% and 99.5025%, respectively.

The models trained with the second training file and tested with the second testing file present similar accuracy values for both approaches (Appendix A – Table A.2), which can be observed in Figure 4.8 (a)-(b). Although, there is a slightly higher values for the approach using the threshold value of 0.5. This is understandable because this value is not influenced by the amount of instances presented potentially classified as plateau waves. In this context, the further results will only focus on the threshold approach and not in the mean approach, although both results are provided.

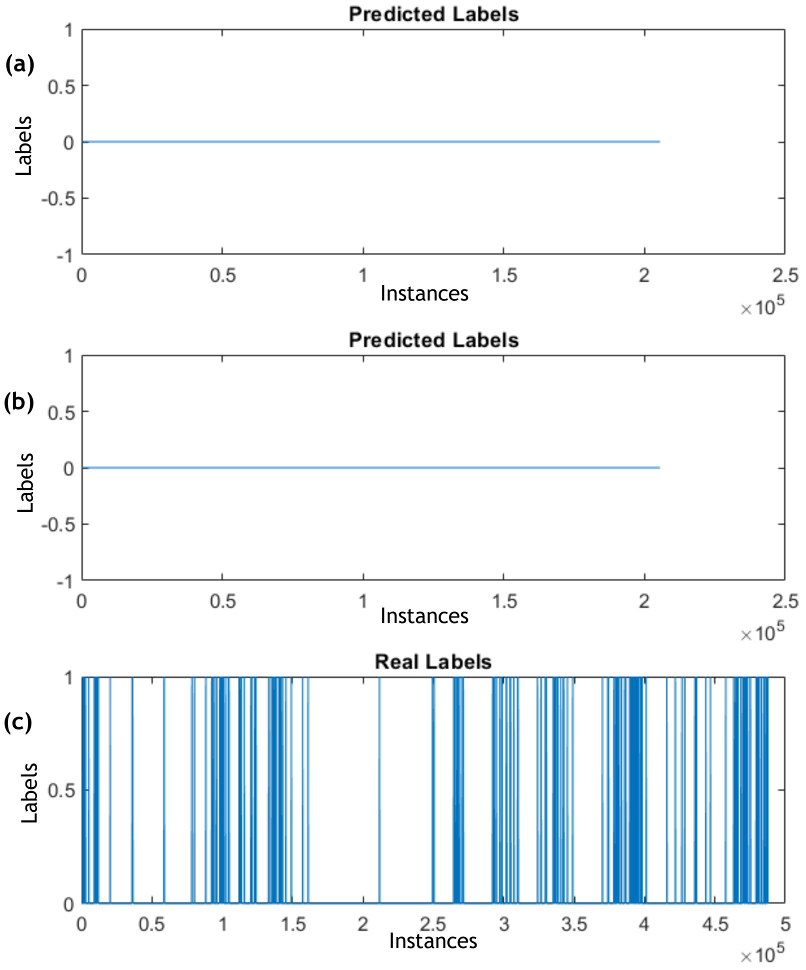


Figure 4.8 – (a) Label predicted by the gaussian process model, applying the threshold value to assign the prediction to classes 0 and 1. (b) Labels computed using the same model but using the mean of the predictions as a threshold for the same computation. (c) Labels computed using the final version of the algorithm developed for the automatic detection of plateau waves, used as the ground truth.

The final set of models trained using the gaussian process had as input data the third training file and they were tested with the third testing file (Appendix A – Table A.2). The threshold value approach resulted in better accuracy values, with the *ardsquaredexponential*, *squaredexponentialand* and *ardrationalquadratic* having the best values. After the plotting of those models, it was noticed that all the instances were predicted as 0. Once again, the high accuracy value reflects the huge amount of the 487648 instances correctly predicted as members of the class 0. Since a small amount of them is predicted as instances that belongs to class 1 by the final version of the algorithm for the automatic detection of plateau waves, the ratio established between the quantity of instances correctly predicted and the number of all the data points presents an elevated value. Similarly to the predictions obtained using the second training file, some of the trained models with the third training file results in predictions similar to Figure 4.8 (a)-(b), instead of a prediction representation similar to Figure 4.8 (c).

The model using the *exponential* kernel function presented the worst accuracy value. When plotting the resulting labels (Figure 4.9), it is noticed that most of the instances are predicted as plateau waves, which results in an elevated number of FP. This is not the desirable outcome since the presence of several FP instances may result in various false alarms. As in the first set of gaussian process models developed, it was the kernel function with the weakest performance results. Thus, it is possible to conclude that this kernel function is not suitable for the development of gaussian process models for the type of data used in this study.

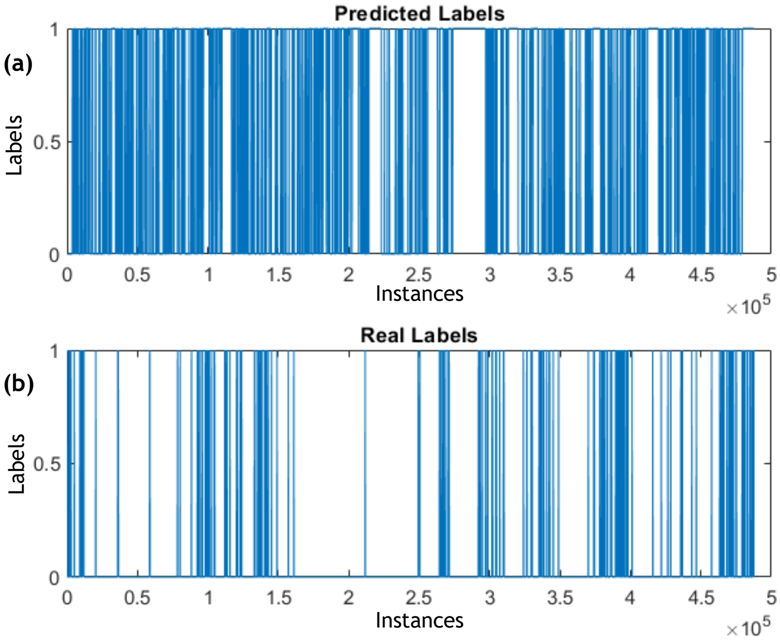


Figure 4.9– (a) Label predicted by the gaussian process model trained with *exponential* kernel function and with the data of the third training file. (b) Labels computed using the same model but using the mean of the predictions as a threshold for the same computation.

As an extra to evaluate the performance of the models trained and testing with their respective third files, the confusion matrix was computed for each one and their results are presented in Appendix A – Table A.3. Due to the importance associated with the quantity of TP, TN, FP and FN samples, the model with the best performance is the one considering the *ardexpoential* kernel function, although it is not the model with the highest accuracy value. However, the model does not detect FN instances, which means that all the plateau waves were correctly predicted. Thus, it classified 1289 as plateau waves instances and the remaining 486359 instances as members of class 0. According to the final version of the algorithm used for the detection of plateau waves, a total of 697 instances were classified as plateau waves. This value of instances can be founded in the number of true positive instances. In comparison, the amount determined by the gaussian process model is higher. Thus, the sensitivity of the model corresponds to 1, the specificity has a value of 0.9988 and a precision of 0.5407. These evaluators values for the other gaussian process models trained with the third training file can be consulted in Appendix A – Table A.4. The absence of FN cases in this model is an important feature because, if only the results provided by the model was used to look for the plateau waves, it would not misclassify instance as a normal value when it corresponds to a plateau wave value. Thus, the plotting of the predictions computed by this gaussian process model (Figure 4.10) shows similarities, although it is possible to detect that Figure 4.10 (a) contains more instances of the class 1 than the quantity of instances presented in Figure 4.10 (b).

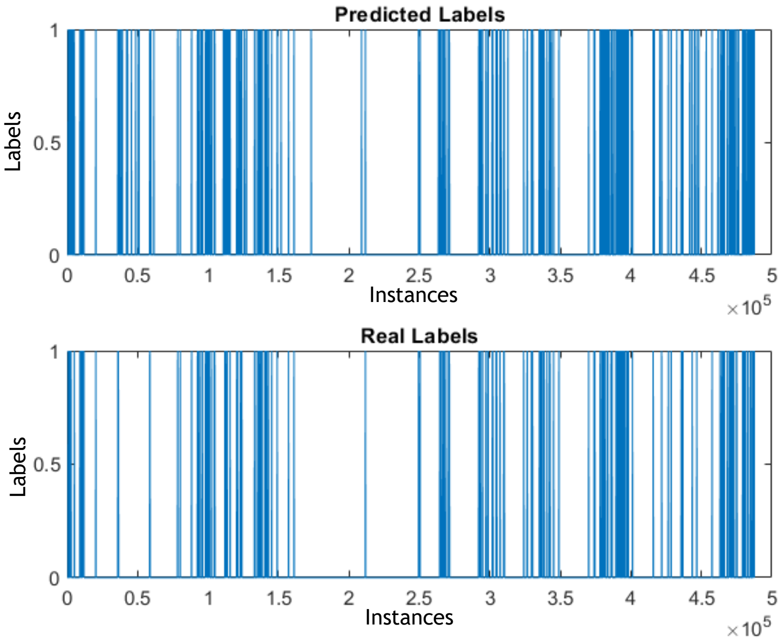


Figure 4.10 – (a) Label predicted by the gaussian process model trained with *ardexponential* kernel function and with the data of the third training file. (b) Labels computed using the same model but using the mean of the predictions as a threshold for the same computation.

4.2 - Discussion of the Results

4.2.1– Computation of the Labels

Regarding the results obtained for the computation of the labels, the algorithm apparently has a good performance in the detection of plateau waves, as it can be observed in Figure 4.1. In a further analysis, the date and time of the physician and algorithm predictions were compared, and the results are presented in Table 4.6.

Observing the obtained results, it is possible to conclude that the date and time of predictions detected by the algorithm did not match with the episodes identified in ICM+. The reason for this is that the data displayed in ICM+ contains more than one data point collected from the patient every second. However, when the patient data is exported as csv file, it was notice that between two consecutive data point there is a one minute interval. Thus, the data presented in the file corresponds to a part of all the data available. Hence, the algorithm detects each plateau wave in the first instance that fulfil the conditions, instead of the instance when the plateau wave starts, as it happens in the predictions in ICM+. In this context, the similarity between the data points should be considered rather than their equalities; thus, the similarities in the data points demonstrate a correct detection of the plateau waves.

Table 4.6 — Comparison between the physician and the algorithm prediction of plateau wave.

Number of Plateau Wave	Physician Identification	Algorithm Detection
PW1	05/03/2017 23:05:51	05/03/2017 23:06:01
PW 2	06/03/2017 00:50:01	06/03/2017 00:51:01
PW 3	06/03/2017 15:41:01	06/03/2017 15:42:00
PW 4	07/03/2017 09:18:01	07/03/2017 09:19:00 07/03/2017 09:27:01 07/03/2017 09:29:01
PW 5	Not detected	08/03/2017 09:39:24
PW 6	08/03/2017 09:45:24	08/03/2017 09:46:24
PW 7	08/03/2017 13:35:24	08/03/2017 13:35:24
PW 8	09/03/2017 00:18:38	09/03/2017 00:19:38
PW 9	09/03/2017 00:53:05	09/03/2017 00:52:00
PW 10	09/03/2017 22:47:07	09/03/2017 22:44:00
PW 11	10/03/2017 04:18:33	10/03/2017 04:18:07
PW 12	14/03/2017 22:11:16	14/03/2017 22:10:38
PW 13	21/03/2017 06:54:18	21/03/2017 06:54:00

PW – Plateau Wave

However, there are two interesting results that must be clarified. The first is related with the PW4. The oscillations presented in ICP values during the 10 minutes of data collected results in the detection of three plateau waves by the algorithm. Indeed, they can be considered as a unique wave because of the general behaviour of ICP and ABP signals. Consequently, although the prediction is not incorrect, the algorithm can be improved by adding a step responsible for ensuring that the episodes closest in time, similar to this case, are predicted as a unique wave. The second result is associated with the detection of PW5 by the algorithm. This wave was not identified in ICM+, because, clinically, it seems to be a unique episode with a long duration, associated with PW6, that presents some ICP fluctuations between 30 and 40 mmHg. Thus, it is more correct that PW5 is not considered as plateau wave, but it can be an indicator of the presence of a plateau wave occurring soon.

Besides, there are two situations that can induce the incorrect detection of plateau waves due to behaviour that the signals present. However, a deeper analysis results in the elimination the presence of a plateau wave. These two episodes were not detected by the algorithm, which means that the established definition of plateau wave avoids the detection of false positive events.

Despite the misclassification of PW5, whose detection was based on the fulfilment of the conditions presented in the algorithm, it is possible to conclude that the algorithm shows a good performance in the detection of plateau waves. It is important to notice that there are no plateau waves missing, which may be considered a good recall of the algorithm. As a second opinion tool, it would never invalidate the diagnosis made by the physicians and the misclassification of instances as plateau waves can be undone.

4.2.2 - Unsupervised Learning Techniques

The purpose of the application of the unsupervised learning techniques consisted in the creation of a model that can correctly classify the data points as instances of one of the two classes to label the data. However and as discussed in sections 2.2.3.1 and 3.5.1, the application of PCA technique did not result in reasonable results for the current application, since this technique applied a transformation of the axis of the original data. In this sense, it is not suitable as unsupervised learning technique.

Using k-means method, several models were trained. Their accuracies were calculated for the data used to train and for a new set of data. Although these accuracy values vary between two subsequent trainings, some of the models present good accuracy. For those models, after the plotting of the predicted labels, it was noticed that all the instances were predicted as data points of the class 0. Thus, the high accuracy value reflects the elevated quantity of instances correctly labelled as 0 expressing the absence of plateau waves. Therefore, the non-prediction of plateau waves combined with the fact that the performance evaluator can vary between two iterations for the same experimental conditions were reasons to not develop an algorithm capable to detect the plateau waves with this method. Besides, due to the results of the other technique, the idea of the employment of unsupervised learning techniques to create an initial label of the data was discarded. Thus, the algorithm for the detection of the plateau waves was developed.

4.2.3 - Supervised Learning Techniques

Considering only the accuracy value with the plot of label predicted by the logistic regression model, it could be asserted that the model could be present a very good performance to predict the label for new data. However, the training was performed only using the ICP and CPP values since it was applied as one of the first techniques. Thus, the data of the third testing file cannot be used to predict the label of the instances, since it contains the EEG features that must be considered to train models. Moreover, the third file was used to try to train a logistic regression model, but the computation was not concluded because it required an elevated computational time. Besides, logistic regression is a method that does not support missing values in the data inputted to train and the file contains such values.

Although it is an appropriate method to be applied for the study of time series, the developed long-short term memory model presented weak performance, expressed by the lower accuracy values. The application of the incorrect labelling of the data and the first training file could be the origin for those results. However, other techniques trained with the same data showed better performance, while the application of the third testing file resulted in a slight improvement of the performance evaluators. Thus, it is possible to conclude that this recurrent neural network is not suitable to be trained and tested with the other files to predict the labels for new data.

The performance of the models trained in neural networks tool show a better performance if the ICP values are used as an input data, instead of being used as labels. With this, and as expected, the ICP values must be always provided as input data for the prediction of the instances. Besides, the increased number of hidden units shows an improvement in accuracy values, because more intermediate computations are performed which means a better balance of the inputs to obtain the new labels.

Regarding the available results for the first set of trained models, the Levenberg-Marquardt presents the better performance in fitting app. On the other hand, the Scaled Conjugate Gradient is the algorithm associated with the models that present the worst performance values in all the applications, since their mean squared error values are higher. However, it is difficult to comment the results because different iterations of training result in different accuracy values. Such variation can be observed in Figure 4.11, that shows the prediction of the labels for the same file using two different trained model with the Levenberg-Marquardt training algorithm in the fitting tool. This means that if all the models were retrained, the model that present the better performance would probably change. The reason that explains this behaviour is the random partition of the input data that each time creates different subsets of data to train, validate and test the model. In addition, this division of the data eliminates the idea associated with the evolution of the signals over time which is not convenient for the development of models that must predict the existence of new plateau waves.

In general, the application of the second training file did not result in improved models' performance, comparing with the results obtained in the previous set of models. Once again, there are models that show a lowest mean squared or cross-entropy value results, representing higher performance. However, they were obtained for that training iteration and another iteration would result in different values, because of the mentioned division of the data into small subsets to be used on the development of the model. Thus, this neural network tool cannot be used to develop models to predict plateau waves, due to the variety that iteration can originate in the development of models.

The SVM models exhibit high performance values. The three models trained for each input file, as a result of the application of the different kernel functions, present the same accuracy. However, such values derived from the large quantity of instances well classified as instances of the class where

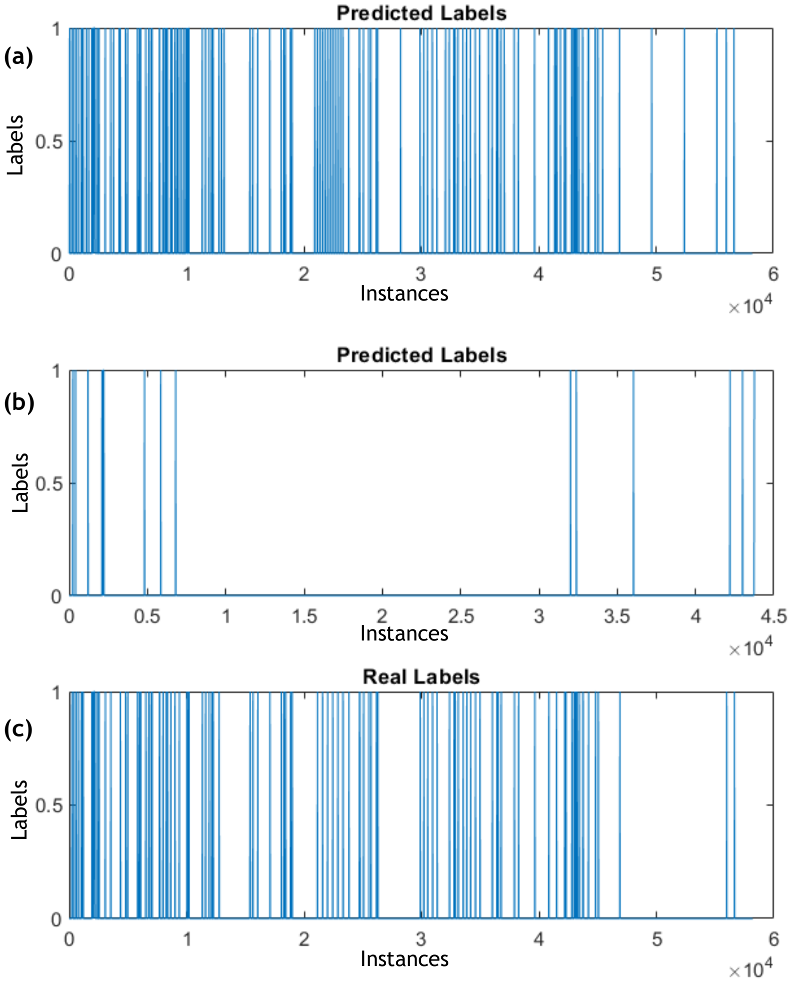


Figure 4.11 - (a)-(b) Labels predicted by models resulting from the fitting app of time series tool, using the first training file as input data and the labels computed from distance metric, in two subsequent iterations, and (c) represents the labels considered as real to calculate the accuracy of the models.

plateau waves are absent, because all the points were predicted as members of the class 0. Since 697 of 487648 instances classified as plateau waves is a small number, the value of the accuracy is slightly affected, and thus is presents a value close to the perfect prediction.

The gaussian process method presents overall models with high accuracy values, revealing their good performance for the purpose of the present study. Although the training and testing steps performed with different files, algorithms and kernel functions, the models tested with the last testing file are the most important because there are related with the study of a possible relationship between the beginning of plateau waves and the EEG signal. The model trained with the *ardexponential* kernel function exhibits the better performance and it can indicate that the employment of the EEG features, accomplished with the signal itself and the two of the most important signals for the detection of plateau waves, can be represent some relationship.

For an intensive analysis of the model performance, the confusion matrix was computed. The *ardexponential* kernel function model present the most interesting results in the matrix, since it does not predict any plateau wave as a member of class 0, although the plateau wave prediction exceeds the number of actual instances. Providing a higher number of instances predicted as plateau waves is a preferable outcome than providing a smaller number of predictions, because a physician can detect the misclassification associated with the additional episodes. However, the absence of the detection of the pathologies or, as in this case, the critical events can compromise patient's life, if the algorithm is the only method used to detect the plateau waves and evaluate the state of the patient over time. Notwithstanding, the physicians' prognosis is still the more important and appropriate, because of their vast knowledge and expertise on neurocritical field.

To conclude, a lot of machine learning techniques were trained and tested as shown in the first section of this chapter. In addition to the algorithm developed for the automatic detection of plateau waves that seems to be very accurate in the detection, the gaussian process model using the *ardexponential* kernel function is the model that, in despite of not presenting the best accuracy value, the confusion matrix shows the best results. Since the model does not detect FN instances, which could influence the treatment of the patient, it is possible to conclude that it maybe could be used for further application.

Although there are some other machine learning models developed with high accuracies, these values do not represent good models because they are associated with the high quantity of instances well classified as instances of class 0, since just a small subset of them are classified as plateau waves and therefore it generates a minimum impact in the accuracy values. While, the neural networks models, developed to predict plateau waves, presented also very good performance evaluators, they have the problem of the random division of the data and thus the evolution of the signals is lost, since the purpose was to create a model that can predict a new episode based on the changes that happen before the occurrence of the plateau wave.

Chapter 5

Conclusion

5.1 – Conclusion

Triggered by a vasodilation cascade loop, plateau waves are characterised by an increase of ICP to values above 40 mmHg, during a minimal period of two minutes, accomplished with changes in other signals, such as an almost constanst ABP and consequently a decrease in CPP. Such episodes may end spontaneously by the passive activation of the vasoconstriction mechanism or in response to active treatment. Plateau waves are recurrent phenomena observed in TBI patients that have their pressure-volume compensatory reserve system exhausted with brain autoregulation preserved. Severe TBI patients are admitted in the NCCU to be managed under continuous multimodal monitoring to avoid secondary insults, such as long periods of intracranial hypertension and ischemia and further development of secondary brain injury. Multimodal monitoring includes, besides from the three above-mentioned signals, other waveforms, such as HR, CO_2, EEG, RSO_2 and $pbtO_2$, collected from patients.

Resulting from the computation of some collected signals, additional variables are calculated, which can reveal some further information about the patient health status. RAP coefficient is the Pearson correlation coefficient between ICP and its amplitude and describes the state of the pressure-volume compensatory reserve, also called brain compliance. PRx index is the Pearson correlation coefficient between ICP and CPP and describes the cerebrovascular reactivity to pressure as an estimation of the autoregulation status of the brain.

Based on the multimodal monitoring, a tool capable to automatic detect the episodes of increased ICP could improve the patient's prognosis. Since there are a lot of applications in the medical field using machine learning techniques and some studies already developed involving ICP signal with positive contributions to treat neurocritical patients, the algorithm should be developed based on machine learning techniques. Machine learning consists in a collection of techniques capable to design models based on artificial intelligence. This implies that based on the input data, the model can extract information and use it to group the data according to the similarities, as an unsupervised learning technique, or it can learn using a combination of the input data with their labels, as a supervised learning technique.

Due to the extensive quantity of methods available, the results obtained in the first part of this study was the starting point. The first step consisted in the collection of patient data to create the files that must be used as training and as testing data. After that, k-means, logistic regression, long-short term memory and neural network techniques were trained to evaluate if one of them could result in a model with a good performance to detect the plateau waves in the collected data. However, although some of the models could be promising, when plotting the labels predicted by the model, most of them did not detect any plateau wave. Thus, the high performance values reflect the large amount of instances well classified as members of class 0.

At the beginning, one initial problem was encountered, because the supervised learning methods required the label of the data as input. In this context, a first version of the algorithm for the detection of plateau waves was developed. Due to the poor results obtained, the characterisation of plateau waves was changed, and the algorithm improved.

A new objective had arisen to study the relationship of the EEG signal at the beginning of plateau waves. For that, features related with the amplitude and the frequency content of the signal were extracted and used as input data. Thus, the structure of the input file data had changes to include those features. Most of the models trained with this data were based on gaussian process method and SVM. The gaussian process models were trained and tested for various kernel functions, but *ardexponential* is the one that performs better. Although, its accuracy value is not the highest, the confusion matrix outcome reveals that apparently it detects all the plateau waves and some additional episodes. Confusion matrix is an important evaluator because the study of the quantity of FN suggests if the algorithm did not detect plateau waves. Besides, it provides an overview of the misclassification of the instances.

In this regard, based on the overall performance of the models trained with gaussian process method, and in particular the one trained with *ardexponential*, it is possible to conclude that the EEG signal may present some changes that can be related with the beginning of the plateau waves. However, it is important to notice that EEG is a stochastic signal, which may justify the occurrence of an unexpected behaviour that may occur at any time. However, since signals such ICP and ABP are presented in the input data provided to train the models, they can have a considerable importance in the computation of the final predictions.

5.2 – Suggestions for Future Work

The final version of the algorithm developed for the detection of plateau waves shows a good overall performance. However, an additional step can be included to algorithm to ensure that the detection of several plateau waves near in time corresponds to various episodes or if they can be related with changes of a unique plateau wave. Despite this improvement, the algorithm can be implemented as a plugin in ICM+. Such plugin would allow the detection of plateau waves using the online data, i.e. the predict would be performed simultaneously with the collection of the signals in real time. Besides, it could be used at offline mode, with the existing patient files, for a further analysis.

The input data used to train the gaussian process models that demonstrates the better performance contained missing values. However, some of the machine learning methods applied does not support the presence of missing values in the input data. In this sense, other approaches to handle with the missing data and to separate the data of two different patients can be studied and implemented. The data may be applied to all the techniques to conclude if one of them result in a model with higher global performance than the gaussian process model presents using *ardexponential* as kernel function.

Moreover, the extraction of EEG features to study the relationship between EEG and plateau episodes resulted in the computation of models with good accuracy. Alternatively, a new set of EEG features namely related to processed EEG may be explored and improved. Besides, other methods for the transformation of the data, apart from the fast Fourier transform may be used, to transform the EEG signal from time domain to frequency domain, such as the wavelet transform. In addition, this method allows the employment of both domains simultaneously for the extraction of features.

Appendix A

Table A.1 — Performance results obtained from the training in Time Series Tool for each problem, using the data of three patients.

Type of Problem	Training Algorithm	Mean Squared Error
NARX	Levenberg-Marquardt	1.57×10^{-5}
	Bayesian Regularization	1.33×10^{-10}
	Scaled Conjugate Gradient	0.000587
NAR	Levenberg-Marquardt	0.000448
	Bayesian Regularization	0.000481
	Scaled Conjugate Gradient	0.000639
Nonlinear input-output	Levenberg-Marquardt	0.371
	Bayesian Regularization	0.879
	Scaled Conjugate Gradient	0.706

Table A.2 — Performance values obtained from the training of different kernel functions, for each training file developed. The threshold and the mean designations represent the two approaches used to reassign the data points to the classes.

Version of Input Data File	**Kernel Function**	**Accuracy (%)**	
		Threshold	**Mean**
First	*Ardsquaredexponential*	98.7353	98.7353
	Exponential	0.8419	3.7122
	Squaredexponential	97.4354	0.2070
	Matern 32	99.3877	1.5691
	Matern 52	98.9093	0.3253
	rationalquadratic	99.7321	93.1879
	ardexponential	97.5890	88.0338
	ardmatern32	97.9908	89.8775
	ardmatern52	97.8951	89.4374
	ardrationalquadratic	99.5025	95.1971
Second	*Ardsquaredexponential*	99,8571	99.3303
	Exponential	99.3303	97.1106
	Squaredexponential	3.1190	99.3303
	Matern 32	97.2063	99.3303
	Matern 52	97.2063	99.3303
	rationalquadratic	99.3303	99.3303
	ardexponential	99.3303	88.5381
	ardmatern32	99.9225	99.3303
	ardmatern52	99.7321	99.3303
	ardrationalquadratic	99.8661	99.3303
Third	*Ardsquaredexponential*	99.8688	99.8688
	Exponential	56.3574	56.3585
	Squaredexponential	99.8669	99.8688
	Matern 32	56.3585	56.3585
	Matern 52	56.3585	56.3585
	rationalquadratic	99.8669	99.8688
	ardexponential	99.8651	7.3518
	ardmatern32	99.8669	53.4537
	ardmatern52	7.3518	7.3518
	ardrationalquadratic	99.8688	99.8688

Table A.3 — Confusion Matrix of each gaussian process model trained with the data of the third training file developed. The threshold and mean designations represent the two approaches used to reassign the data points to the classes.

Kernel Function	**Approach to recalculate labels**	**Confusion Matrix**	
Ardsquaredexponential	Threshold	TN = 486951	FP = 0
		FN = 697	TP = 0
	Mean	TN = 486951	FP = 0
		FN = 697	TP = 0
Exponential	Threshold	TN = 274515	FP = 212436
		FN = 408	TP = 289
	Mean	TN = 274515	FP = 212436
		FN = 408	TP = 289
Squaredexponential	Threshold	TN = 486951	FP = 0
		FN = 697	TP = 0
	Mean	TN = 486951	FP = 0
		FN = 697	TP = 0
Matern 32	Threshold	TN = 274515	FP = 212436
		FN = 408	TP = 289
	Mean	TN = 274515	FP = 212436
		FN = 408	TP = 289
Matern 52	Threshold	TN = 274515	FP = 212436
		FN = 408	TP = 289
	Mean	TN = 274515	FP = 212436
		FN = 408	TP = 289
Rationalquadratic	Threshold	TN = 486951	FP = 0
		FN = 697	TP = 0
	Mean	TN = 486951	FP = 0
		FN = 697	TP = 0
Ardexponential	Threshold	TN = 486359	FP = 592
		FN = 0	TP = 697
	Mean	TN = 35211	FP = 451740
		FN = 0	TP = 697

Ardmatern32	Threshold	TN = 486951
		FN = 697
			FP = 0
			TP = 0

	Mean	TN = 260246
		FN = 262
			FP = 226705
			TP = 435

Ardmatern52	Threshold	TN = 35211
		FN = 0
			FP = 451740
			TP = 697

	Mean	TN = 35211
		FN = 0
			FP = 451740
			TP = 697

Ardrationalquadratic	Threshold	TN = 486951
		FN = 697
			FP = 0
			TP = 0

	Mean	TN = 486951
		FN = 697
			FP = 0
			TP = 0

Table A.4 — Specificity, sensitivity and precision values for each gaussian process model trained with the third training file. The threshold and mean designations represent the two approaches used to reassign the data points to the classes.

Kernel Function	Threshold	Mean

Ardsquaredexponential	specificity = 1.0000
	sensitivity = 0.0000
	precision = NaN
		specificity = 1.0000
		sensitivity = 0.0000
		precision = NaN

Exponential	specificity = 0.5637
	sensitivity = 0.4146
	precision = 0.0014
		specificity = 0.5637
		sensitivity = 0.4146
		precision = 0.0014

Squaredexponential	specificity = 1.0000
	sensitivity = 0.0000
	precision = NaN
		specificity = 1.0000
		sensitivity = 0.0000
		precision = NaN

Matern 32	specificity = 0.5637
	sensitivity = 0.4146
	precision = 0.0014
		specificity = 0.5637
		sensitivity = 0.4146
		precision = 0.0014

Matern 52	specificity = 0.5637
	sensitivity = 0.4146
	precision = 0.0014
		specificity = 0.5637
		sensitivity = 0.4146
		precision = 0.0014

Rationalquadratic	specificity = 1.0000
	sensitivity = 0.0000
	precision = NaN
		specificity = 1.0000
		sensitivity = 0.0000
		precision = NaN

Ardexponential	specificity = 0.9988
	sensitivity = 1.0000
	precision = 0.5407
		specificity = 0.0723
		sensitivity = 1.0000
		precision = 0.0015

Ardmatern32	specificity = 1.0000
	sensitivity = 0.0000
	precision = NaN
		specificity = 0.5344
		sensitivity = 0.6241
		precision = 0.0019

Ardmatern52	specificity = 0.0723
	sensitivity = 1.0000
	precision = 0.0015
		specificity = 0.0723
		sensitivity = 1.0000
		precision = 0.0015

Ardrationalquadratic	specificity = 1.0000
	sensitivity = 0.0000
	precision = NaN
		specificity = 1.0000
		sensitivity = 0.0000
		precision = NaN

NaN – value not available

References

[1] Waikato, T.U.o. *Weka 3 - Data Mining with Open Source Machine Learning Software in Java*. 2019 [cited 2019 January]; Available from: https://www.cs.waikato.ac.nz/ml/weka.

[2] Lusa, *Traumatismo crânio encefálico grave mata 600 pessoas por ano em Portugal*, in *Público*. 2016.

[3] OLIVEIRA, E., et al., *Traumatismo Crânio-Encefálico: Abordagem Integrada.* Acta Médica Portuguesa, 2012. 25(3).

[4] MathWorks. *MathWorks - Solutions - MATLAB & Simulink*. 2019 June 18]; Available from: https://www.mathworks.com/solutions.html?s_tid=gn_sol.

[5] Czosnyka, M., et al., *Significance of intracranial pressure waveform analysis after head injury.* Acta neurochirurgica, 1996. 138(5): p. 531-542.

[6] HAYASHI, M., et al., *Plateau-wave phenomenon (I) correlation between the appearance of plateau waves and CSF circulation in patients with intracranial hypertension.* Brain, 1991. 114(6): p. 2681-2691.

[7] Varsos, G.V., et al., *Critical closing pressure during intracranial pressure plateau waves.* Neurocritical care, 2013. 18(3): p. 341-348.

[8] Czosnyka, M., et al., *Hemodynamic characterization of intracranial pressure plateau waves in head-injured patients.* Journal of neurosurgery, 1999. 91(1): p. 11-19.

[9] Luis, A., et al. *Heart rate variability during plateau waves of intracranial pressure: a pilot descriptive study.* in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. 2015. IEEE.

[10] Czosnyka, M., *Association between arterial and intracranial pressures.* British journal of neurosurgery, 2000. 14(2): p. 127-128.

[11] Dias, C., et al., *Pressures, flow, and brain oxygenation during plateau waves of intracranial pressure.* Neurocritical care, 2014. 21(1): p. 124-132.

[12] Czosnyka, M., et al., *A synopsis of brain pressures: which? when? Are they all useful?* Neurological research, 2007. 29(7): p. 672-679.

[13] Czosnyka, M. and J.D. Pickard, *Monitoring and interpretation of intracranial pressure.* Journal of Neurology, Neurosurgery & Psychiatry, 2004. 75(6): p. 813-821.

[14] Donnelly, J., et al., *Regulation of the cerebral circulation: bedside assessment and clinical implications.* Critical care, 2016. 20(1): p. 129.

[15] Kim, M.O., et al., *Principles of cerebral hemodynamics when intracranial pressure is raised: lessons from the peripheral circulation.* Journal of hypertension, 2015. 33(6): p. 1233.

[16] Rosner, M.J. and D.P. Becker, *Origin and evolution of plateau waves: experimental observations and a theoretical model.* Journal of neurosurgery, 1984. 60(2): p. 312-324.

[17] Daley, M., et al., *Plateau waves: changes of cerebrovascular pressure transmission*, in *Intracranial Pressure and Brain Monitoring XII*. 2005, Springer. p. 327-332.

[18] Goodfellow, I., et al., *Deep learning*. Vol. 1. 2016: MIT press Cambridge.

[19] Maglogiannis, I.G., *Emerging artificial intelligence applications in computer engineering: real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies*. Vol. 160. 2007: Ios Press.

[20] Bishop, C.M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 2006: Springer-Verlag.

[21] Molnar, C., *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2019.

[22] Haykin, S.S., *Neural networks and learning machines*. 2009: New York: Prentice Hall.

[23] Ray, S., et al., *Understanding Support Vector Machine algorithm from examples (along with code)*. Analytics Vidhya, 2016.

[24] Rasmussen, C.E., *Gaussian Processes in Machine Learning*, in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, O. Bousquet, U. von Luxburg, and G. Rätsch, Editors. 2004, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 63-71.

[25] Rasmussen, C.E. and H. Nickisch, *Gaussian processes for machine learning (GPML) toolbox*. Journal of machine learning research, 2010. 11(Nov): p. 3011-3015.

[26] Liu, H., et al., *When Gaussian process meets big data: A review of scalable GPs*. arXiv preprint arXiv:1807.01065, 2018.

[27] Williams, C.K. and C.E. Rasmussen. *Gaussian processes for regression*. in *Advances in neural information processing systems*. 1996.

[28] MathWorks. *Kernel (Covariance) Functions Options - MATLAB & Simulink*. 2019 2019-05-28]; Available from: https://www.mathworks.com/help/stats/kernel-covariance-function-options.html#d117e19649.

[29] Hyvärinen, A., J. Karhunen, and E. Oja, *Independent Component Analysis*. 2001: Wiley.

[30] Shalizi, C., *Advanced data analysis from an elementary point of view*. 2013, Citeseer.

[31] Fielding, A.H., *Cluster and Classification Techniques for the Biosciences*. 2007: Cambridge University Press.

[32] Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern Classification (2nd Edition)*. 2000: Wiley-Interscience.

[33] Sivaganesan, A., G.T. Manley, and M.C. Huang, *Informatics for neurocritical care: challenges and opportunities*. Neurocritical care, 2014. 20(1): p. 132-141.

[34] Güiza, F., et al., *Novel methods to predict increased intracranial pressure during intensive care and long-term neurologic outcome after traumatic brain injury: development and validation in a multicenter dataset*. Critical care medicine, 2013. 41(2): p. 554-564.

[35] Chen, W., et al. *Intracranial pressure level prediction in traumatic brain injury by extracting features from multiple sources and using machine learning methods*. in *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2010. IEEE.

[36] Bishop, S.M. and A. Ercole, *Multi-Scale Peak and Trough Detection Optimised for Periodic and Quasi-Periodic Neuroscience Data*, in *Intracranial Pressure & Neuromonitoring XVI*. 2018, Springer. p. 189-195.

[37] Scalzo, F., D. Liebeskind, and X. Hu, *Reducing false intracranial pressure alarms using morphological waveform features*. IEEE Transactions on Biomedical Engineering, 2012. 60(1): p. 235-239.

[38] Meyfroidt, G., et al., *Computerized prediction of intensive care unit discharge after cardiac surgery: development and validation of a Gaussian processes model*. BMC medical informatics and decision making, 2011. 11(1): p. 64.

[39] Chacón, M., et al., *Comparing models of spontaneous variations, maneuvers and indexes to assess dynamic cerebral autoregulation*, in *Intracranial Pressure & Neuromonitoring XVI*. 2018, Springer. p. 159-162.

[40] UK, C.E., *ICM+ Standard of Procedures*. 2017.

[41] MathWorks. *Needed FRequency analysis of an EEG signal - MATLAB Answers - MATLAB Central*. 2019 June 5]; Available from: https://www.mathworks.com/matlabcentral/answers/32168-needed-frequency-analysis-of-an-eeg-signal.

[42] MathWorks. *kmeans clustering - MATLAB kmeans*. 2019 June 9]; Available from: https://www.mathworks.com/help/stats/kmeans.html.

[43] MathWorks. *Train neural network for deep learning - MATLAB trainNetwork*. 2019 June 9]; Available from: https://www.mathworks.com/help/deeplearning/ref/trainnetwork.html.

[44] MathWorks. *Options for training deep learning - MATLAB trainingOptions*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html.

[45] MathWorks. *Getting Started with Deep Learning Toolbox - Video*. 2019 June 8]; Available from: https://www.mathworks.com/videos/getting-started-with-neural-network-toolbox-68794.html.

[46] MathWorks. *Divide Data for Optimal Neural Network Training - MATLAB & Simulink*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ug/divide-data-for-optimal-neural-network-training.html.

[47] MathWorks. *Levenberg-Marquardt backpropagation - MATLAB trainlm*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ref/trainlm.html.

[48] MathWorks. *Bayesian regularization backpropagation - MATLAB trainbr*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ref/trainbr.html.

[49] Møller, M.F., *A scaled conjugate gradient algorithm for fast supervised learning*. Neural networks, 1993. 6(4): p. 525-533.

[50] MathWorks. *Scaled conjugate gradient backpropagation - MATLAB trainscg*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ref/trainscg.html.

[51] MathWorks. *Cluster with Self-Organizing Map Neural Network - MATLAB & Simulink*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html.

[52] MathWorks. *Shallow Neural Network Time-Series Prediction and Modeling - MATLAB & Simulink*. 2019 June 7]; Available from: https://www.mathworks.com/help/deeplearning/gs/neural-network-time-series-prediction-and-modeling.html.

[53] MathWorks. *Train shallow neural network - MATLAB train*. 2019 June 8]; Available from: https://www.mathworks.com/help/deeplearning/ref/train.html.

[54] MathWorks. *Train support vector machine (SVM) classifier for one-class and binary classification - MATLAB fitcsvm*. 2019 June 8]; Available from: https://www.mathworks.com/help/stats/fitcsvm.html?s_tid=doc_ta.

[55] MathWorks. *Fit a Gaussian process regression (GPR) model - MATLAB fitrgp*. 2019 June 8]; Available from: https://www.mathworks.com/help/stats/fitrgp.html.

[56] MathWorks. *Compute confusion matrix for classification problem - MATLAB confusionmat*. 2019 June 12]; Available from: https://www.mathworks.com/help/stats/confusionmat.html.

[57] Costaridou, L., *Medical image analysis methods*. 2005: CRC Press.