UNIVERSITAT DE BARCELONA

Final Degree Project
Biomedical Engineering Bachelor

“Predictive modelling of Loss Of Consciousness under general anaesthesia“

Barcelona, 15th of January of 2022

Author: Marc Palomer Cadenas

Director: Pedro L. Gambús

Acknowledgments

Thanks to Dr Pedro Luís Gambús and all the charming professionals from Operating Room number 4, which helped me get comfortably involved in the surgical environment, as well as to Undergraduate PhD Joan Altés, whose mentoring was of crucial importance for the good development of this project.

Also, this project would also not have been possible without the support of my family and friends, with special mention to my grandmother Maria del Carmen Cadenas, my partner Berta Lloveras, my fellow collegue Joana Collet, my mother Carmen Cadenas and my father Albert Palomer, who have teached me all I needed to accomplish my academical goals while enjoying life in its whole complexity.

Abstract

Loss of consciousness in general anaesthesia induction is a process of great interest not only in study and understanding of human consciousness, but particularly in the evaluation of the state of a patient in an operating room. This project creates and evaluates two simple ML models for binary consciousness classification under TIVA propofol-remifentanil TCI mediated general anaesthesia. These models are a Support Vector Machine and a Random Forest Classifier. These two models are capable of determining the state of arousal of the patient between conscious and unconscious at every second using as inputs NIBP, Heart rate and Propofol and Remifentanil effect site concentrations. SVM model has shown better performance than RFC in all assessed metrics, being those ROC AUC $_{SVM}$ = 0.97 vs. ROC AUC $_{RFC}$ = 0.93, Accuracy $_{SVM}$ = 0.9 vs. Accuracy $_{RFC}$ = 0.87 and F1-Score $_{SVM}$ = 0.91 vs. F1-Score $_{SVM}$ = 0.88. The code used to implement the chosen approach is available at a public GitHub repository.

Index of Tables

Table 1: Systemic effects of i.v. agents. SVR, Systemic Vascular Resistance; MAP, Mean Arterial Pressure; CBF, Cerebral Blood Flow (Khurram Saleem Khan, Sept. 2013, pp. 100-105) 10
Table 2: Most used Total Intra Venous Anaesthetics (Roop Kaw, 2014). 13
Table 3: Recorded variables in this project. Redundant variables have been excluded, and we the same variables recorded for propofol have been recorded for remifentanil. (Altès, 2020, pp. 26) 16
Table 4: Confusion Matrix Diagram 22
Table 5: RFC metrics 31
Table 6: SVC metrics 31
Table 7: Hardware Budget 37
Table 8: Software Budget 37
Table 9: Human Resources Budget.. Business cost of labour per hour has been used, instead of workers' labour per hour salary for more accurate results 37
Table 10: SaMD IMDRF risk categorization (FDA, 2021). 38

Index of Figures

Figure 1: Git structure (Noble desktop, 2021). 7
Figure 2: GitHub archives distribution. 8
Figure 4: Neurophysiological mechanisms of propofol’s actions in the brain. Propofol enhances γ-aminobutyric acid receptor type A (GABAA)-mediated inhibition in the cortex, thalamus, and brainstem. Shown are three major sites of action: postsynaptic connections between inhibitory interneurons and excitatory pyramidal neurons in the cortex; the GABAergic neurons in the thalamic reticular nucleus (TRN) of the thalamus; and postsynaptic connections between GABAergic and galanergic (Gal) projections from the preoptic area (POA) of the hypothalamus and the monoaminergic nuclei, which are the tuberomammillary nucleus (TMN) that releases histamine (His), the locus ceruleus (LC) that releases norepinephrine (NE), the dorsal raphe (DR) that releases serotonin (5HT); the ventral periacqueductal gray (vPAG) that releases dopamine (DA); and the cholinergic nuclei that are the basal forebrain (BF), pedunculopontine tegmental (PPT) nucleus, and the lateral dorsal tegmental (LDT) nucleus that release acetylcholine (ACh). Also shown is the lateral hypothalamus (LH) that releases orexin (Patrick L. Purdon, Clinical Electroencephalography for Anesthesiologists: Part I: Background and Basic Signatures., 2015, pp. 937–960). 13
Figure 5: Action of neuromuscular blockers on Acetylcholinergic ion channels (Marieke Kruidering-Hall, 2016) 15
Figure 6: EEG time domain visualizations in different unconsciousness levels (Patrick L. Purdon, Clinical Electroencephalography for Anesthesiologists: Part I: Background and Basic Signatures., 2015, pp. 937–960).. 17
Figure 7: Example of a black box system transforming some discrete data (blue) to continuous (red). 18
Figure 8: Visualization of algorithms vs. artificial intelligence vs. machine learning vs. deep learning (Author: Johannes Vrana, Vrana GmbH, Licenses: CC BY-ND 4.0) 19
Figure 9: Predictive modelling scheme (Parthasarathy, 2021) 19
Figure 10: SVM diagram (Pushpa Singh N. S., 2021, pp. 80-111). 20
Figure 11: Random Forest classifier diagram (Siddharth Misra, 2020, pp. 243-287) 21
Figure 12: KNN diagram (Pushpa Singh N. S., 2021, pp. 89-111) 21
Figure 13: ROC critical points (Zoronza, s.f.) 24
Figure 14: ROC-AUC 31
Figure 15: Confusion matrix for RFC 32
Figure 16: Confusion matrix for SVC 32
Figure 17: Feature importance analysis with RFC. 32
Figure 20: General overview of the project's code. 33
Figure 18: LoC Prediction Example 34
Figure 19: LoC Prediction Example under threshold = 0.85 34
Figure 21: GANTT diagram of the project 35
Figure 22: SWOT Analysis 36

Index of Equations

Equation 1: Accuracy formula 22
Equation 2: Precision formula 22
Equation 3: Recall formula 23
Equation 4: F1-Score formula 23

Table of Contents

1 Introduction 6

1.1 Objectives 6

1.2 Methodology and Structure 6

1.2.1 Version Control System 6

1.2.2 Project Structure 7

1.2.3 Project workload 8

2 Background 9

2.1 General Anaesthesia 9

2.1.1 States of anaesthesia 10

2.1.2 Anaesthetic drugs 12

2.1.3 Anaesthesia monitoring 15

2.1.3.1 Basic anaesthesia monitoring 15

2.1.3.2 Advanced monitoring systems 17

2.2 Model theory 18

2.2.1 Predictive modelling 19

2.2.2 ML algorithms 20

2.2.3 Model performance 21

2.3 State of the art 24

3 Market analysis 25

4 Solution implementation 27

4.1 Concept engineering 27

4.1.1 Different solutions 27

4.1.2 Proposed Solution 28

4.2 Detail engineering 28

4.2.1 Data pre-processing and Exploratory analysis 29

4.2.2 Model training 30

4.2.3 Model Validation 31

4.2.4 Example 33

5 Technical Viability 36

5.1 Specifications and technical characteristics 36

5.2 SWOT 36

6 Economic viability 37

7 Legal aspects 38

8 Discussion and Future Prospects 39

8.1 Implications 39

8.2 Limitations 39

8.3 Recommendations and Future Prospects 40

9 Bibliography 41

10 Annex 45

1 Introduction

This project has been conducted at the Systems Pharmacology Effect Control-Modelling (SPEC-M) Research Group of the Department of Anaesthesiology of the Hospital Clinic of Barcelona. Data collection was performed in the Operating Room (OR) number 4 of the Major Ambulatory Surgery Unit of the Hospital Clinic. The data used in the project has been collected inside this OR.

1.1 Objectives

The objectives mentioned below are conceived around the expressed interest of the anaesthesiology team to have a fast prediction of the level of consciousness in the moment of achieving unconsciousness. Even though anaesthesiologists have great knowledge in this matter, no predicted parameter is now a days used to directly approach this state of arousal patients and the anaesthesiologist should rely on indirect measures of unconsciousness such as the electroencephalogram.

Therefore, the principal aims of this project is to generate, train and validate a model in order to predict the level of Loss of Consciousness (LoC) of a patient undergoing general anaesthesia with the administration of Propofol and Remifentanil.

In order to achieve the main goals of the project, several sub-goals are defined:

Firstly, it is crucial to get deeply familiarized with the surgical environment, anaesthetic procedures and methods, and recorded physiological parameters, which will be done through practical sessions in gynaecological operating room 4 during two whole months.

Last but not least, it is necessary to properly use a control version software, generating a commit every time an improvement has been made on the code until the final application is done, and correctly handling possible errors.

1.2 Methodology and Structure

1.2.1 Version Control System

In order to achieve the goals of the project, a version control of code has been applied. Specifically, a Git Version Control System on GitHub has been implemented. These systems are used to generate parallel branches of entire working directories or folders from master (original) one so several people can work parallelly on different issues related to the same code or files, without overlapping problems. Git repositories have the option to merge the work done to the original file, the same way your colleagues do, and to therefore integrate all the work done.

This approach is also useful for a unipersonal project, as this methodology is utterly strict with the improvements to be uploaded in the Git repository. The user keeps making commits to GitHub, this is analogous to the "saving" action the user normally does. The point is that the user only commits ("saves") his work when a whole implementation is finished and correctly working. Therefore, only working code is uploaded to the Git repository using the commit command. Then, the Git VCS has a record of all the commits (or "savings") done, and with the command checkout you can return all your working directory at once to a previous commit in the past (like returning to a previously saved version). This possibility is interesting in two ways: the user can always return to a past commit where all the code is correctly working (in case something messed up) and secondly the user can easily track when and how any change was performed. When the whole work is done, the user merges the branch where he was working to the original one, also containing a working code but without the implementation just created.

The commits in Figure 1 are shown as coloured dots, and all the line with same colour dots is a branch. When, for instance, the blue line re-connects with the green line (main branch), the implementation has been merged to the main code. The whole of the image would represent a coding project.

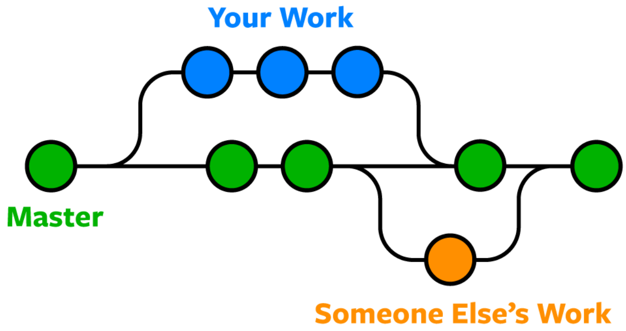


Figure 1: Git structure (Noble desktop, 2021).

As seen in the Figure to the left, two people (or more) can work simultaneously on different branch implementations and later merge them into the original master branch. On the other hand, it can be the same person working parallelly on different implementations, which is our case.

1.2.2 Project Structure

note: The code used to implement the chosen approach is available at a public GitHub repository.

The structure of the project is composed of several files divided in two main groups: Model and Report. Regarding the Model group, there are several files explained below and in the Report group there is this word document containing all the information of the project.

Inside the model group we can find several archives: The archives folder; containing the CSV files of the patients' data, the "TrainTestMatrices" folder; containing the CSV files with the data frames used to train and test the models, the ".pickle" archives; which are the already trained models for later usage, ".png" images of plots; generated inside the code for model explanation, and last but not least four python files containing the code of the project. These four python files are called libraries.py, Train_Test_adquisitions.py, Model.py and example.py.

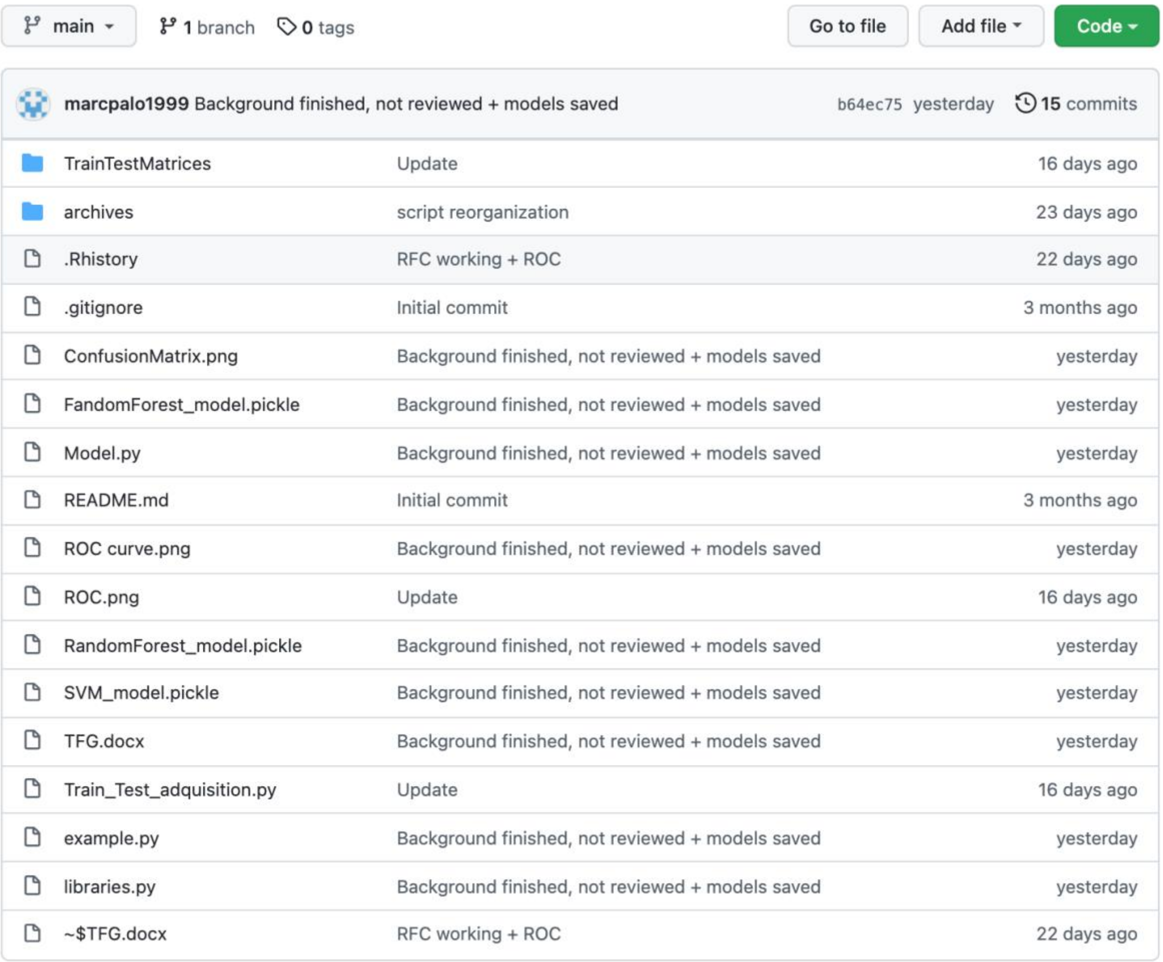
- libraries.py: includes all the libraries needed and is directly imported to the other ".py" files for their use.

- Train_Test_adquisitions.py: Takes the raw data from patients CSV and pre-processes it, selecting the patients matching the needs of the model and taking the given seconds around the target event of LoC. Later it cleans the data and separates it for model training and testing.

- Model.py: Takes the output data from previous file and trains and tests the models, and saves them for later use

- example.py: Taking the model already trained and validated, takes a new patient, and computes its moment of LoC.

The documents' structure is shown in the Figure below, as it is presented in GitHub webpage.



marcpalo1999 Background finished, not reviewed + models saved b64ec75 yesterday 15 commits

TrainTestMatrices	Update	16 days ago
archives	script reorganization	23 days ago
.Rhistory	RFC working + ROC	22 days ago
.gitignore	Initial commit	3 months ago
ConfusionMatrix.png	Background finished, not reviewed + models saved	yesterday
FandomForest_model.pickle	Background finished, not reviewed + models saved	yesterday
Model.py	Background finished, not reviewed + models saved	yesterday
README.md	Initial commit	3 months ago
ROC curve.png	Background finished, not reviewed + models saved	yesterday
ROC.png	Update	16 days ago
RandomForest_model.pickle	Background finished, not reviewed + models saved	yesterday
SVM_model.pickle	Background finished, not reviewed + models saved	yesterday
TFG.docx	Background finished, not reviewed + models saved	yesterday
Train_Test_adquisition.py	Update	16 days ago
example.py	Background finished, not reviewed + models saved	yesterday
libraries.py	Background finished, not reviewed + models saved	yesterday
~$TFG.docx	RFC working + ROC	22 days ago

Figure 2: GitHub archives distribution.

This four-file code structure has been chosen because functional programming, which is based on defining the functions (duties) of the code in a file and implementing them on another, generates clearer and more understandable codes. Furthermore, comments have been written all over the four archives to further improve their understanding.

1.2.3 Project workload

The structure of the project is divided in three differentiated temporal sections corresponding to the main three workloads of the project:

- Practical sessions in the operating room
- Working from home on the data and documents
- Document submission and personal presentation.

Firstly, practical sessions in the OR take place, where data is collected from the anaesthesia and surgical procedures in OR-4 of the Major Ambulatory Surgery Unit on Tuesdays and Thursdays from September until November. In this section, Dr. Pedro Gambús guides the recording and gathering of the intraoperative data and gives explanations around the activity undergone in the operating room and the parameters recorded.

)

Secondly, work from home takes place. In this section procedures like data analysis and document preparation are performed. Data preparation and model building, training and validation are performed using python language and Visual Studio Code user interphase. The packages used in the data analysis are a wide range, with special mention to scikit-learn. This package is a python package specifically designed for ML programming. In this section PhD undergraduate Joan Altés guides the programming process.

Finally, documents submission and presentation take place.

2 Background

In interventions under general anaesthesia some methods are used by the anaesthesiologists in order to determine the state of consciousness of the patient. Typical examples of clinically relevant functional measurements to determine LoC are presence or absence of verbal response to a homogenous command, loss of eyelash reflex, and presence or absence of corneal reflex. If the three of them are negative or absent, the patient is considered to be unconscious (Sergio Vide, 2020), thus permitting further actions such as securing the airway by means of a laryngeal mask or laryngoscopya and intubation followed by the surgical procedure itself.

This clinical assessment has been frequently endorsed by the introduction of indexes, mostly extracted from EEG signal processing, that try to indirectly indicate the depth of hypnotic effect of the patients. Bispectral Index (BIS™) is an example of this approach, which gives a value between 0-100, being 0 complete absence of electrical activity in the frontal neurons and 100 completely conscious. Recommended optimal values of BIS™ under general anaesthesia range from 40-60 (Mathur S, 2021 Jan).

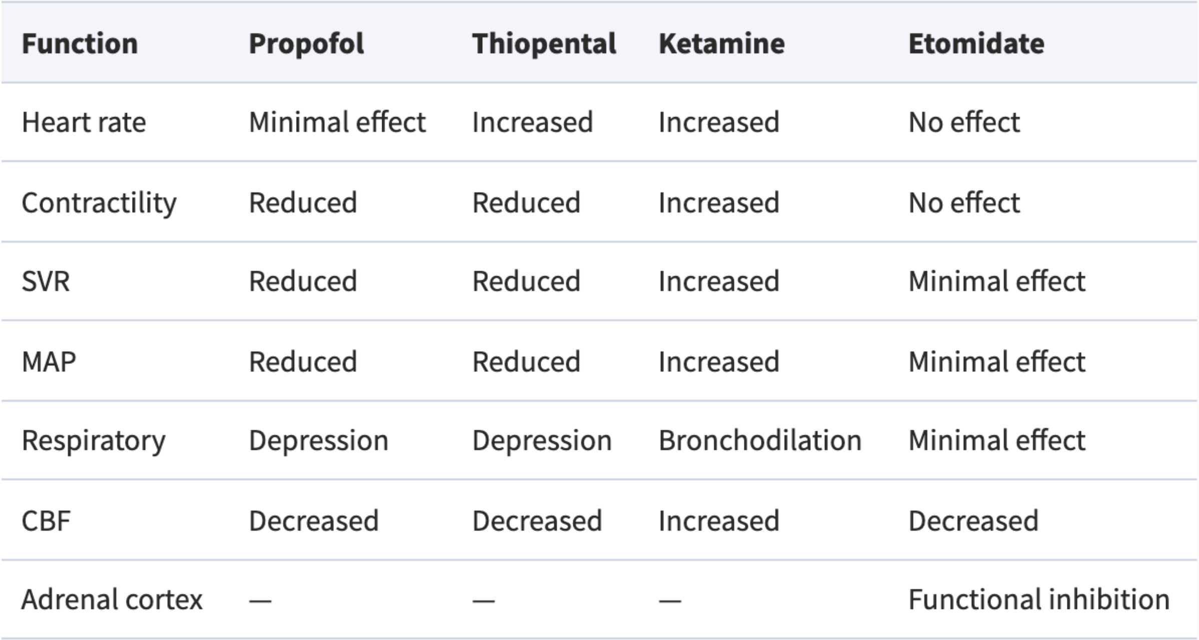
In the approach proposed, the goal was to generate an index independent from EEG but trained to identify at every second the state of consciousness of the patient through training with previous patients' recorded data of easy access, such as NIBP, HR, and propofol and remifentanil effect site concentrations.

2.1 General Anaesthesia

General anaesthesia is a drug induced reversible state of loss of consciousness, with loss of both protective reflexes as well as lack of perception of painful stimulation (Flores FJ, Aug 2017). Depending on the anaesthetics administered and their effects in the brain and muscles the effects of the anaesthesia are discerned in four different parts: unconsciousness or hypnosis, analgesia, amnesia, and skeletal muscle relaxation (Siddiqui BA, 2021).

That general anaesthesia is a reversible coma state of the patient is a widely accepted statement in the clinical environment. Although the patterns of EEG activity observed in comatose patients depend on the extent of the brain injury, they frequently resemble the high–amplitude, low-frequency activity seen in patients under general anaesthesia, therefore supporting this analogy (Emery N. Brown, 2010). This comparative rises an immediate question: How much anaesthetics are enough for an intervention under general anaesthesia? The desired effects upon CNS does not remain isolated: when anaesthetic drugs are administered a wide range of other parameters such as HR, Contractility, SVM, MAP and respiratory functions are modulated. Once the desired therapeutic effect is achieved, further increasing concentrations of anaesthetics generally enhance these undesired effects of the drugs both within the surgical procedure and less drastically throughout the recuperation process. It is therefore accepted that the optimal concentrations of anaesthetics in a surgical procedure is the one that achieves the desired therapeutic effects and not more than that (Khurram Saleem Khan, Sept. 2013, pp. 100-105).

Table 1: Systemic effects of i.v. agents. SVR, Systemic Vascular Resistance; MAP, Mean Arterial Pressure; CBF, Cerebral Blood Flow (Khurram Saleem Khan, Sept. 2013, pp. 100-105)



Function	Propofol	Thiopental	Ketamine	Etomidate
Heart rate	Minimal effect	Increased	Increased	No effect
Contractility	Reduced	Reduced	Increased	No effect
SVR	Reduced	Reduced	Increased	Minimal effect
MAP	Reduced	Reduced	Increased	Minimal effect
Respiratory	Depression	Depression	Bronchodilation	Minimal effect
CBF	Decreased	Decreased	Increased	Decreased
Adrenal cortex	—	—	—	Functional inhibition

The final desired therapeutic effect when performing general anaesthesia is to achieve the LoC state of the patient, where the effect of the anaesthetics is strong enough to start with the surgical procedure without risk of sudden awakening of the patient, but avoiding the undesired increased effect of administering too much anaesthetics as shown in Table 1.

Several pharmacological models and approaches such as Target Controlled infusion (TCI), EEG suppression analysis and BIS™ analysis have been implemented to achieve the goal of administering only the dose of anaesthetic drugs required to achieve the desired effect to individualize drug dosing (Absalom AR, Jan. 2016) and will be further discussed in the Anaesthetic drugs and Anaesthesia monitoring chapters respectively.

2.1.1 States of anaesthesia

As stated in the previous section, the states of anaesthesia are not part of a continuous process towards being under general anaesthesia, but the different effects expected from anaesthetic drugs which might be necessarily combined to achieve the therapeutic effect necessary to start the intervention safely. These states of anaesthesia are unconsciousness, which refers to patients' unawareness state; analgesia, which refers to patients' incapacity of feeling painful stimuli; amnesia, which refers to patients' inability to remember; and akinesia, which refers to patients' immobility.

Hypnosis and unconsciousness

The state of hypnosis, which is analogous to unconsciousness, is typically defined as a state of impairment of cognitive functions in the patients' brain that blocks patients' adequate response to external stimuli, regarding attention and perception (Jason A. Campagna, 2003). Hypnosis is induced by hypnotic agents such as propofol, and there are several levels of hypnosis evolving in a continuum of deeper intensity of effect for instance: drowsiness, sedation and unconsciousness.

The mechanisms of action involved in the process of hypnosis have not yet been determined accurately as a whole. Nevertheless, in the last decades it has been shown that different molecular targets in various regions of the nervous system are involved in the multiple components of anaesthetic action, and these targets can vary between specific anaesthetics. When talking about unconsciousness induction the main inhibitory neurotransmitter involved is Gamma-aminobutyric acid (GABA). This neurotransmitter acts selectively modulating $GABA_A$ receptors, which are ligand-gated ion channels persistently opening the channels and therefore inhibiting synapses in big areas of the brain (Hemmings HC Jr, (2005), pp. 503-510).

Even though several indexes mostly related to EEG have been used to quantify the level of hypnosis of patients under general anaesthesia, our intraoperative ability to evaluate levels of consciousness still remains limited. An example of this approach is Bispectral index (BIS™), but further research in this field is still necessary to construct deterministic models involving hypnosis induction and its effects upon patients.

Analgesia

In 2020 the International Association for the Study of Pain proposed a review of the definition of pain. The new accepted definition stated that pain was "An unpleasant sensory and emotional experience associated with, or resembling that associated with, actual or potential tissue damage" (Srinivasa N Raja, 2020). Therefore, analgesia could be defined as the state of mind that permits the inhibition of painful sensations in situations where it would be expected, such as in tissue damage derived from surgical action.

Amnesia

The term amnesia refers to the impossibility of a patient to remember or recall any moment or event that they perceived during the transient state. Hypnotic agents cause unconsciousness by suppressing neural mechanisms mediating arousal and awareness, but they can also cause amnesia by disrupting mechanisms of memory consolidation (Rudolph, 2004, pp. 709–720).

When a high enough dose of a hypnotic agent is administered, information from the outside world is not processed by the patient and, thus, no conscious memories are formed. In this sense, all hypnotic agents are amnesic at a certain dose. Furthermore, hypnotic agents acting on GABA receptors such as propofol might have amnesic effect even at lower sedative doses (Robert A. Veselis, 2009, pp. 1-2).

Akinesia

The term akinesia refers to the inability to perform a clinically perceivable movement. It can present as a delayed response, freezing mid-action, or finally total abolition of movement (Ramakrishnan S, 2021). In general anaesthesia akinesia is typically achieved by means of hypnotic and analgesic agent combination although depending on the type of surgery and anaesthetic drugs and doses used, it is sometimes necessary to administer a neuromuscular blocking agent, such as Rocuronium, to achieve muscle relaxation to avoid movement or even muscle contraction during the course of the surgical intervention.

2.1.2 Anaesthetic drugs

Anaesthetic drugs are the chemical compounds administered to patients in order to achieve the different states of anaesthetic desired. Depending on if the desired effect is local anaesthesia, sedation, general anaesthesia, or others the anaesthetic drugs, their quantities and the way of administration are chosen.

Anaesthesia drugs can either be inhalational or intravenous. Inhalation anaesthesia techniques means an intravenous induction usually with propofol combined with remifentanil or other opioid as analgesic drug, followed by a maintenance with an inhalation anaesthetic such as desflurane of sevoflurane combined with an intravenous opioid to avoid response to noxious stimulation. Total intravenous anaesthetics (TIVA) refers to the administration of anaesthetic agents via intravenous route during the whole anaesthetic procedure including induction and maintenance. Compared to inhalational anaesthesia, TIVA shows several benefits including better hemodynamic stability, better recovery conditions and reduced post-operative incidence of nausea and vomiting (Brian J. Anderson, 2019, pp. 177-198).

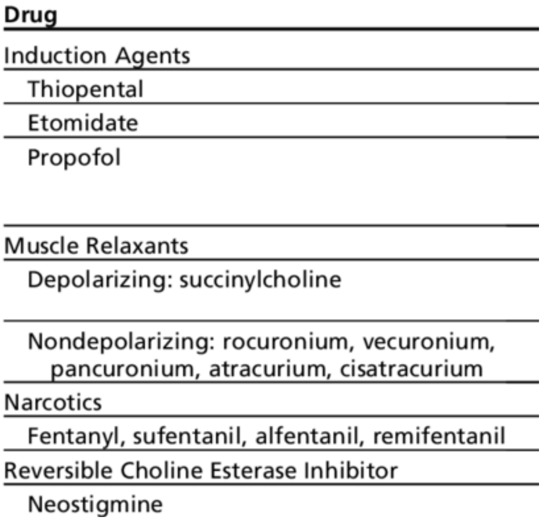
The infusion of TIVA can either be administered manually through an initial bolus injection, or through the usage of algorithm-mediated infusion pumps, such as TCI.

TCI

Progress in computing technology has allowed the development of target-controlled infusion devices, with drugs delivered to achieve specific predicted target site blood drug concentrations. Target controlled infusion (TCI) systems have been developed as a standardised infusion system for the administration of opioids, propofol and other i.v anaesthetics. Pharmacokinetic-pharmacodynamic (PKPD) models including demographic covariate factors such as age, weight, gender, and height are incorporated in the control system of the infusion pump. After entering in the infusion device the demographic data relevant as significant covariate factor for the PKPD model, the user sets the target concentration desired to be achieved either in plasma or effect site. According to the PKPD model, every ten seconds the systems inject the bolus dose of drug required to achieve the target concentration selected. Then it maintains such target concentration constant during the time required. The latter being done by adjusting the rate of a constant infusion to compensate for the amount of drug distributed from the blood to the tissues as well as the drug that is being cleared from the organism (Struys MM, Jan. 2016, pp. 56-69).

Regarding drug classification, anaesthetic drugs are divided first on their effect and later on their type. Most common drug types used are hypnotic and analgesic drugs, sometimes combined with other directly injected i.v drugs if necessary, such as Rocuronium for muscle relaxation. In the Hospital Clinic a mixture of propofol and Remifentanil is used, as this hypnotic and analgesic altogether have demonstrated to create a synergic effect, achieving the desired therapeutic effect with less concentration of both drugs (Mertens MJ, 2003). Some of the most used TIVA drugs are classified in the Table below.

Table 2: Most used Total Intra Venous Anaesthetics (Roop Kaw, 2014).



Drug
Induction Agents
Thiopental
Etomidate
Propofol
Muscle Relaxants
Depolarizing: succinylcholine
Nondepolarizing: rocuronium, vecuronium, pancuronium, atracurium, cisatracurium
Narcotics
Fentanyl, sufentanil, alfentanil, remifentanil
Reversible Choline Esterase Inhibitor
Neostigmine

Propofol

Propofol is an intravenous hypnotic agent used for procedural sedation, during monitored anaesthesia care, or as an induction agent for general anaesthesia. It may be administered as a bolus or with TCI. Propofol is prepared in a lipid emulsion which gives it the characteristic milky white appearance. Strict aseptic technique must be used when drawing up propofol as the emulsion can support microbial growth (Folino TB, 2021).

Propofol also induces amnesia, as stated in the *Section 2.1.1.1 States of anaesthesia*, as information from the outside world is not processed by the patient and, thus, no conscious memories are formed. This anaesthetic agent interacts with GABA receptor mediated ionic channels between others, as shown in figure below.

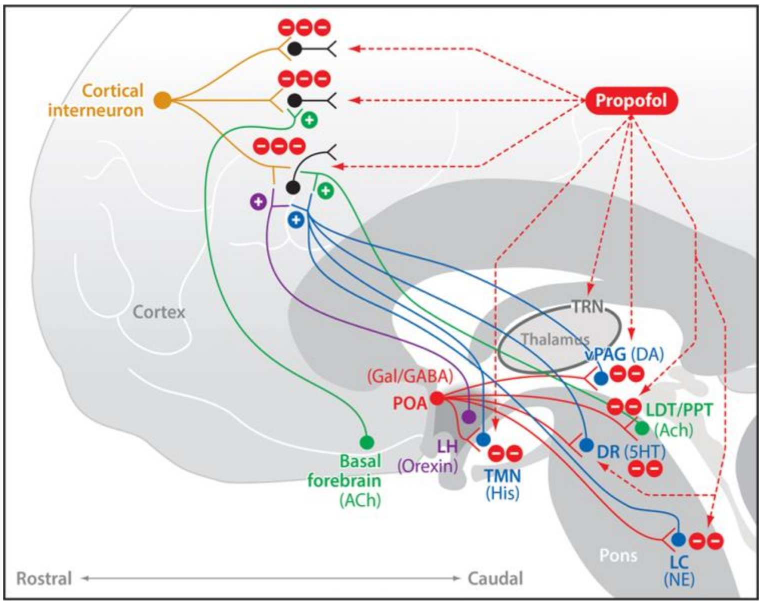


Figure 3: Neurophysiological mechanisms of propofol's actions in the brain. Propofol enhances γ-aminobutyric acid receptor type A (GABAA)-mediated inhibition in the cortex, thalamus, and brainstem. Shown are three major sites of action: postsynaptic connections between inhibitory interneurons and excitatory pyramidal neurons in the cortex; the GABAergic neurons in the thalamic reticular nucleus (TRN) of the thalamus; and postsynaptic connections between GABAergic and galanergic (Gal) projections from the preoptic area (POA) of the hypothalamus and the monoaminergic nuclei, which are the tuberomammillary nucleus (TMN) that releases histamine (His), the locus ceruleus (LC) that releases norepinephrine (NE), the dorsal raphe (DR) that releases serotonin (5HT); the ventral periacqueductal gray (vPAG) that releases dopamine (DA); and the cholinergic nuclei that are the basal forebrain (BF), pedunculopontine tegmental (PPT) nucleus, and the lateral dorsal tegmental (LDT) nucleus that release acetylcholine (ACh). Also shown is the lateral hypothalamus (LH) that releases orexin (Patrick L. Purdon, Clinical Electroencephalography for Anesthesiologists: Part I: Background and Basic Signatures., 2015, pp. 937–960).

The figure above shows the inhibitory action of propofol in various sites of the brain, including cortical and subcortical structures. The disconnection between these two cerebral structures has been widely discussed to be the main component in loss of consciousness in patients anaesthetised with this drug (P. Guldenmund, 2015).

Several PK/PD models exist for propofol. Some have been integrated in commercially available target-controlled infusion (TCI) systems, which in and by itself attests to the usefulness of these PK/PD models for propofol: TCI systems facilitate clinical practice and maintenance of intravenous anaesthesia (Engbers FH, Feb. 2010). Commercial TCI devices incorporate the Marsh and the Schnider propofol PK/PD models. For effect-site targeting, the Schnider model should be used, for plasma targeting, the Marsh model. The use of the lean body mass (LBM) equation in the Schnider model limits its use in obese patients (Sergio Vide, 2020, p. 105).

Remifentanil

Remifentanil, a fentanyl derivative, is a short acting, nonspecific esterase metabolised, selective mu-opioid receptor agonist, with a pharmacodynamic profile typical of opioid analgesic agents. Notably, the esterase linkage in remifentanil results in a unique and favourable pharmacokinetic profile for this class of agent. Adjunctive intravenous remifentanil during general anaesthesia is an effective and generally well tolerated opioid analgesic in a broad spectrum of patients, including adults and paediatric patients, undergoing several types of surgical procedures.

As mentioned in the *section 2.1.1.1 States of Anaesthesia*, remifentanil is used as an intravenous analgesic agent and it is usually infused together with propofol, as they have presented a synergic combination achieving the same therapeutic effect with less concentration of both agents (Mertens MJ, 2003).

Rocuronium

Rocuronium bromide is an amino steroid non-depolarizing neuromuscular blocker or muscle relaxant used in modern anaesthesia to facilitate tracheal intubation by providing skeletal muscle relaxation, commonly required for surgery or mechanical ventilation. Rocuronium is also commonly used to avoid unvoluntary hiccups while operating.

In general, neuromuscular blocking agents are divided between depolarizing and non-depolarizing. Depolarizing muscle relaxants act as acetylcholine (ACh) receptor agonists, whereas nondepolarizing muscle relaxants function as competitive antagonists as seen in figure below (Butterworth, 2013, p. chapter 13).

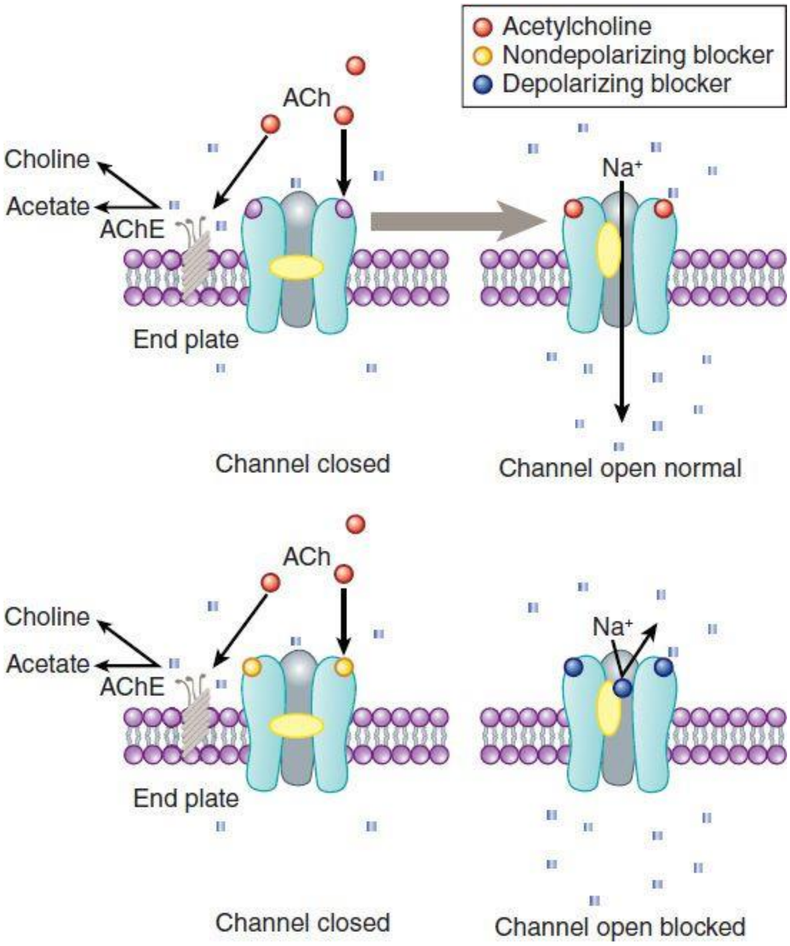


Figure 4: Action of neuromuscular blockers on Acetylcholinergic ion channels (Marieke Kruidering-Hall, 2016)

When comparing, non-depolarizing blockers are reversed by acetylcholinesterase inhibitor drugs since non-depolarizing blockers are competitive antagonists at the ACh receptor so can be reversed by increases in ACh. On the other hand, the depolarizing blockers already have ACh-like actions, so these agents have prolonged effect under the influence of acetylcholinesterase inhibitors. Administration of depolarizing blockers initially produces fasciculations (a sudden twitch just before paralysis occurs). This is due to depolarization of the muscle. Also, post-operative pain is associated with depolarizing blockers (Neuromuscular blocking agents, 2021).

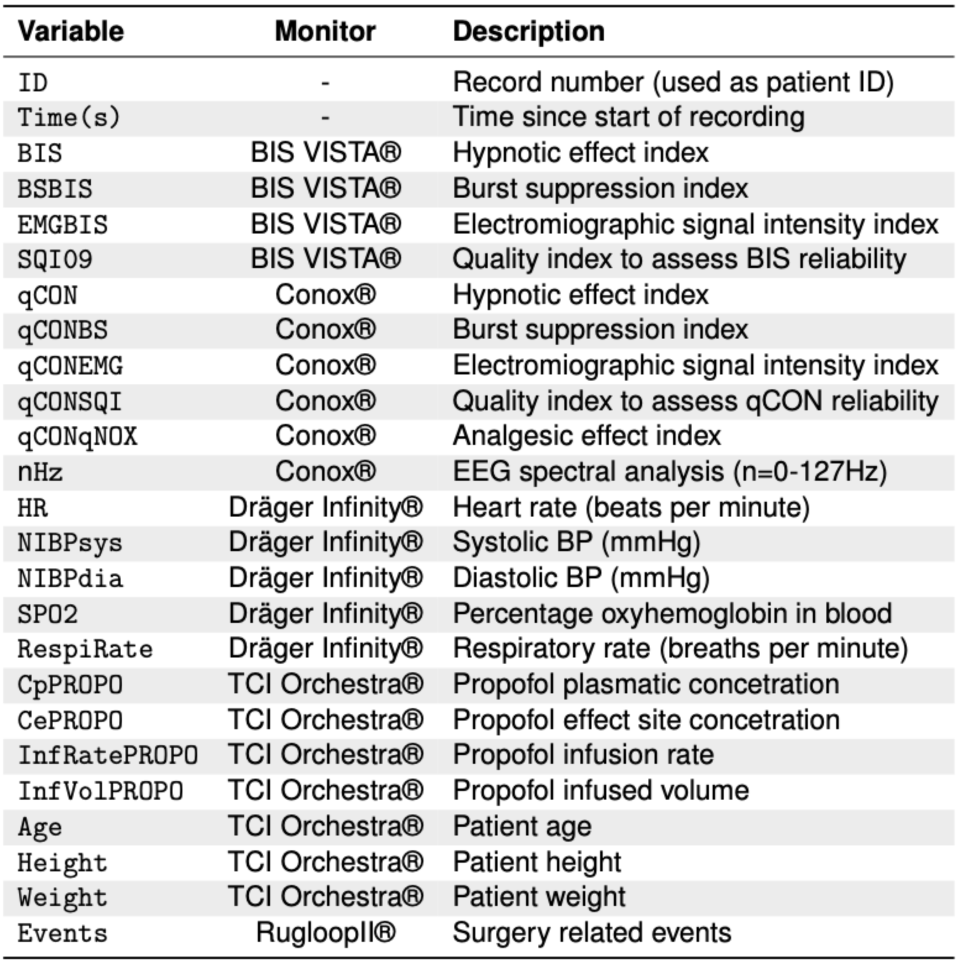
2.1.3 Anaesthesia monitoring

2.1.3.1 Basic anaesthesia monitoring

Anaesthesia monitoring is a crucial control system to guarantee the correct physiological state of the patient under surgery. Several sensors are used to record data based on biomedical signals and biomedical parameters in order to analyse in situ the correct state of the patient and also to later process the data to obtain information and generate models, as it is the goal of this project.

The list of variables recorded in the CMA OR number 4 through anaesthesia monitoring are the following:

Table 3: Recorded variables in this project. Redundant variables have been excluded, and we the same variables recorded for propofol have been recorded for remifentanil. (Altès, 2020, pp. 26)



Variable	Monitor	Description
ID	-	Record number (used as patient ID)
Time(s)	-	Time since start of recording
BIS	BIS VISTA®	Hypnotic effect index
BSBIS	BIS VISTA®	Burst suppression index
EMGBIS	BIS VISTA®	Electromiographic signal intensity index
SQI09	BIS VISTA®	Quality index to assess BIS reliability
qCON	Conox®	Hypnotic effect index
qCONBS	Conox®	Burst suppression index
qCONEMG	Conox®	Electromiographic signal intensity index
qCONSQI	Conox®	Quality index to assess qCON reliability
qCONqNOX	Conox®	Analgesic effect index
nHz	Conox®	EEG spectral analysis (n=0-127Hz)
HR	Dräger Infinity®	Heart rate (beats per minute)
NIBPsys	Dräger Infinity®	Systolic BP (mmHg)
NIBPdia	Dräger Infinity®	Diastolic BP (mmHg)
SPO2	Dräger Infinity®	Percentage oxyhemoglobin in blood
RespiRate	Dräger Infinity®	Respiratory rate (breaths per minute)
CpPROPO	TCI Orchestra®	Propofol plasmatic concetration
CePROPO	TCI Orchestra®	Propofol effect site concetration
InfRatePROPO	TCI Orchestra®	Propofol infusion rate
InfVolPROPO	TCI Orchestra®	Propofol infused volume
Age	TCI Orchestra®	Patient age
Height	TCI Orchestra®	Patient height
Weight	TCI Orchestra®	Patient weight
Events	RugloopII®	Surgery related events

Electrocardiogram

Electrocardiogram (abbreviated as EKG or ECG) is a relationship between voltage and time recorded from cutaneous electrodes positioned in different derivations. Willem Einthoven first invented it in 1902. An EKG is an integral part of the initial evaluation of a patient suspected of having a cardiac-related problem or, in our case, a biomedical signal recorded in the OR to detect cardiac abnormalities throughout the operation, either way produced by the general anaesthesia or the operation itself (Yasar Sattar, 2021). Several parameters can be derived from ECG as heart rate (HR) and ST segment. HR changes might have relationship with hypnotic and analgesic effect (Jeanne, 2009, pp. 91-96). It is generally desirable to maintain HR between 40 and 100 beats per minute during an intervention.

Arterial blood pressure

Arterial blood pressure can be measured invasively for accurate results, throughout the use of an arterial catheter with a pressure transducer attached to the end. This invasive approach is usually introduced in the radial arteria. On the other hand, non-invasive arterial blood pressure (NIBP), which is the usual approach unless specific indications are met, is measured with an automated oscillatory cuff every 120 seconds (Gbenga Ogedegbe, 2010, pp. 571-586).

It is desirable to keep mean NIBP above 60mmHg and systolic NIBP and diastolic NIBP below 140 and 90mmHg, and there are drugs available to modulate the NIBP of the patient.

Pulse oximetry

Pulse oximetry is a non-invasive method to measure the oxygen saturation in the blood by emitting light at specific wavelengths through tissue (most commonly the fingernail bed). Deoxygenated and oxygenated haemoglobin absorb light at different wavelengths (660 nm and 940 nm respectively), the absorbance at the characteristic wavelengths is then assessed to give a saturation value from 0 to 100% (Torp, Modi, & Simon., 2021).

Capnography

Capnography is the monitoring of the concentration or partial pressure of carbon dioxide (CO2) in the respiratory gases. During general anaesthesia the patient is intubated, and capnography shows the effectiveness of CO2 elimination through the lungs into the anaesthesia equipment. It is therefore useful to assess the correct intubation of the patient. Indirectly, it monitors the production of CO2 by tissues and the circulatory transport to the lungs (Pandya & Sharma., 2021).

2.1.3.2 Advanced monitoring systems

2.1.3.2.1 Electroencephalogram

EEG is the recording of electric activity of the pyramidal neurons of the brain cortex. It can be shown as a continuous graph of voltage against time. The electric activity recorded corresponds to the anatomical cortical area where electrodes have been located. It is a continuous non-invasive measure of brain activity. It consistently reflects the changes induced by anaesthetic drugs. It might indirectly reflect activity from subcortical regions due to their straight electrical interaction (Patrick L. Purdon, 2015, pp. 937-960). Several mathematical approaches have been proposed and used to analyse the complexity of EEG. Fourier analysis performed through the so-called Fast Fourier Transform algorithm to the EEG the signal allows to quantify activity in the frequency domain (Shaker, 2003).

Many of the changes that occur in the brain with changes in consciousness states of the patients can be readily observed in unprocessed EEG recordings. Different behavioural and neurophysiological states induced by anaesthetics are associated with different EEG waveforms. For example, the figure below shows the EEG of the same patient in different states of propofol-induced sedation and unconsciousness (Patrick L. Purdon, 2015, pp. 937–960). EEG patterns in general anaesthesia are altered depending on the drugs used, the dose and duration of infusion, with significant variance between subjects. In general anaesthesia low frequency, high amplitude EEG profiles are enhanced as unconsciousness deepens (Emery N. Brown, 2010).

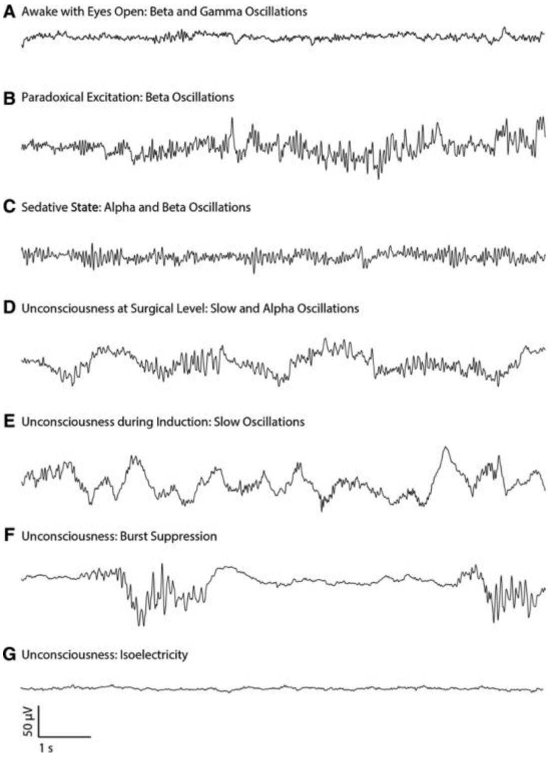


Figure 5: EEG time domain visualizations in different unconsciousness levels (Patrick L. Purdon, Clinical Electroencephalography for Anesthesiologists: Part I: Background and Basic Signatures., 2015, pp. 937–960)..

Furthermore, in section F of the figure we can observe a burst suppression event, where periods of high amplitude and frequency (called burst) are followed by low frequency and amplitude, even isoelectric zones (the suppression). It is a sign of unnecessarily high level of unconsciousness, and the facultative normally uses this information to adjust the anaesthetic dose or infusion rate.

As the raw EEG is complex to analyse by the non-expert user, several approaches of EEG-based indexes have been developed to information of the state of the patient. This indexes generally extract parameters from the EEG through signal processing techniques, for later introducing this information into a confidential algorithm to obtain a 0-100 index representative of the state of unconsciousness of the patient (Mathur S, 2021 Jan).

Bispectral Index (BIS™) is one of the most popular EEG-based indexes, and it has been designed to assess the hypnotic effect of anaesthesia, being 0 complete absence of electric activity of the frontal neurons and around 100 completely conscious. BIS™ monitoring has shown to impact positively in patients' time of extubating, postoperative recovery, memory loss and cognitive impairment in general (Carlos Rogério Degrandi Oliveira, 2017 Jan-Feb, págs. 72-84).

The main drawback of EEG as a monitoring physiological signal is that it is an advanced monitoring approach, being slightly difficult to place, interpretate and with fungible sensors of relatively high price. Unlike ECG, EEG is not used in all the surgical interventions.

2.2 Model theory

In general terms a system can be defined as a box (or transfer function), with a process happening inside of it, which takes some input arguments and delivers some output results. Depending on the processes inside the box and the type of inputs and outputs the system can be classified in several ways. If the process inside the box is not known, it is called a black box.

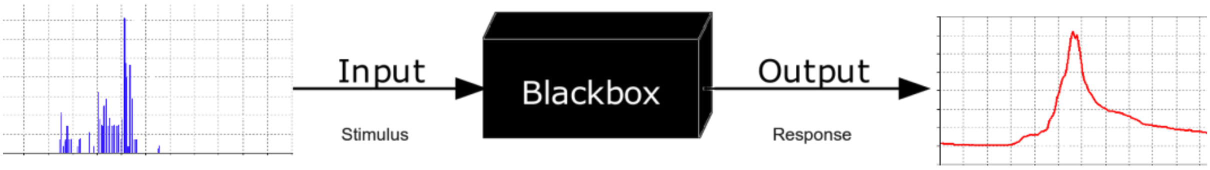


Figure 6: Example of a black box system transforming some discrete data (blue) to continuous (red).

As a matter of fact, any type of modelling is intrinsically a representation, numerical or not, of the part of the world surrounding us we wish to study. When models are developed, some input variables or characteristics are used, and some others are excluded; it is therefore straightforward that models can be good enough to be useful or not but will never pretend to explain a real process in its whole complexity. Otherwise, they will try to simplify the studied system in a way the results have significance.

If a given system is particularly a mathematical model, which is a representation of reality based on mathematical relationships, it can be classified in several ways explained below (Pospeev, 2019):

- Deterministic vs probabilistic: The model's output is the same always for a given set of input values, or otherwise it has an intrinsic randomness which makes the output values vary even for the same set of input values. In this last case we usually see that state variables of the system are not described by unique values, but for probabilistic equations.

- Static vs. dynamic: State variables in static models are not time dependant but in steady state, while in dynamic models' state variables do depend on time.

- Discrete vs. continuous: The equations inside the model follow a continuous paradigm or a discrete one. For instance, the parameters we measured in the OR room were recorded every 1 second, so our model will be discrete.

- Linear vs. nonlinear: Referring the distribution of the mathematical approximations to represent reality, if they are based on linear relationships or of higher order relationships.

- Etc.

2.2.1 Predictive modelling

When talking about predictive modelling, the adjective predictive stands for its finality to obtain information about what is going to happen in the future based on information of the past.

For instance, sophisticated predictive models are used to predict health events in patients and to screen high risk individuals, such as for predicting cardiovascular disease, breast cancer or anaesthetic complications and events. Predictive modelling does not inherently belong to any of the classifications stated above but takes different forms depending on its characteristics and those of the algorithms inside it.

Therefore, a model is predictive if it uses information from the past, e.g., past examples, in order to achieve a future prediction or output result. Even though the term Machine Learning model is frequently used, what we are handling is a predictive model, which inside it we are using a ML algorithm. Also, methods and algorithms used inside the model can be of several types and origins: basic statistics, time-dependant causal equations, AI algorithms such as ML and DL, and a long etcetera.

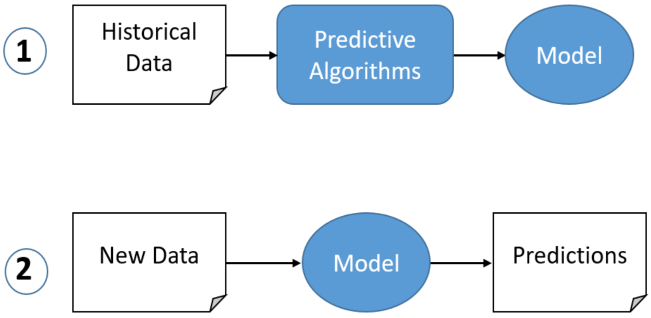


Figure 8: Predictive modelling scheme (Parthasarathy, 2021)

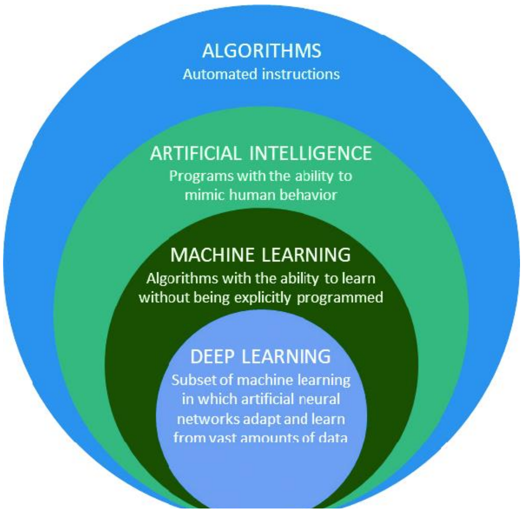

ALGORITHMS
Automated instructions
ARTIFICIAL INTELLIGENCE
Programs with the ability to mimic human behavior
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Figure 7: Visualization of algorithms vs. artificial intelligence vs. machine learning vs. deep learning (Author: Johannes Vrana, Vrana GmbH, Licenses: CC BY-ND 4.0)

But what is exactly ML and what has in common with AI? AI is a set of algorithms and techniques which pretend to mimic or assemble human like mental processes to accomplish tasks which usually require of human intelligence. ML is a subset of the AI paradigm, which is characterised by being capable of learning without being explicitly programmed. Finally, DL is a subset of ML characterised by a brain-like net of artificial neural networks (NN) which the DL algorithm uses to learn, also without being explicitly programmed but using vast amounts of data.

2.2.2 ML algorithms

In this section we will review three of the most spread ML algorithms, Support Vector Components, Random Forest Classifier and K-Nearest neighbour.

Support Vector Machine

SVM is a supervised machine-learning algorithm that provides classification, regression, and outlier detection. SVM learns by creating a separating "line" between the two class points of the training data (e.g., drawn in a x-y plane). This "line", which actually is a hyperplane, must perfectly separate the two classes, and its distance from both classes is to be maximized (Pushpa Singh N. S., 2021, pp. 80-111).

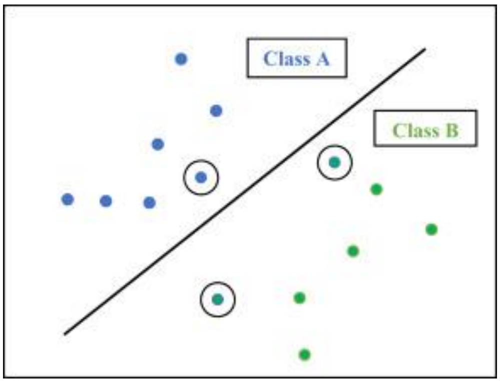


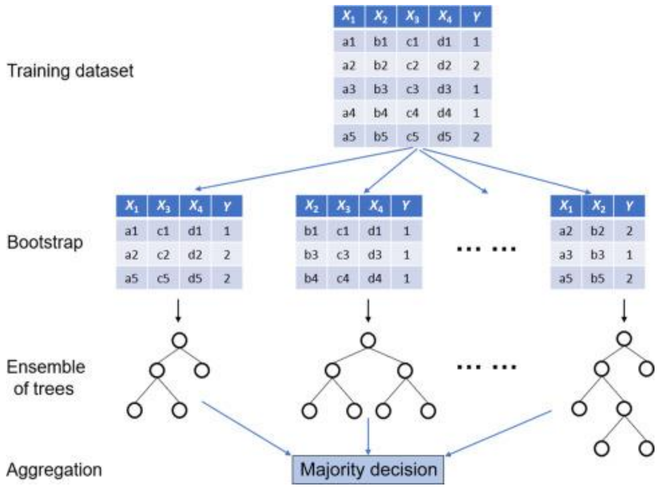
Figure 9: SVM diagram (Pushpa Singh N. S., 2021, pp. 80-111).

Still regarding learning, the closest points to the margin (the hyperplane, also call decision boundary) are called "Support Vectors" and are the ones defining the behaviour of the SVM. The distance from the points to the decision boundary is computed, and the minimum value of this whole is considered the functional margin of the model respect to the dataset. Obviously, the larger the functional margin, the better predictive characteristics will be present in the model. Finally, if it is not possible to separate the two classes through a linear approach, the "Kernel trick" is used, meaning that the model projects the dimensionality information provided to higher dimensional spaces where separation is easier (Pushpa Singh N. S., 2021, pp. 80-111).

Finally, after the training is done, the classifier just draws the new data, the one to be predicted, into the same plane, and depending on the side of the hyperplane where it falls it will be classified as one class or the other.

Random forest classifier

RF classifier is an ensemble method that trains several decision trees in parallel with bootstrapping followed by aggregation, jointly referred as bagging. Bootstrapping indicates that several individual decision trees are trained in parallel on various subsets of the training dataset using different subsets of available features, as seen in Fig.9. Bootstrapping ensures that each individual decision tree in the random forest is unique, which reduces the overall variance of the RF classifier (Siddharth Misra, 2020, pp. 243-287).



X_1	X_2	X_3	X_4	Y
a1	b1	c1	d1	1
a2	b2	c2	d2	2
a3	b3	c3	d3	1
a4	b4	c4	d4	1
a5	b5	c5	d5	2

X_1	X_3	X_4	Y
a1	c1	d1	1
a2	c2	d2	2
a5	c5	d5	2

X_2	X_3	X_4	Y
b1	c1	d1	1
b3	c3	d3	1
b4	c4	d4	1

X_1	X_2	Y
a2	b2	2
a3	b3	1
a5	b5	2

Figure 10: Random Forest classifier diagram (Siddharth Misra, 2020, pp. 243-287)

Regarding the ensemble of trees, each internal node (seen as a circle in Figure 9) represents a 'test' on an attribute (e.g., whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). A node that has no children is a leaf (Donges, 2021).

For the final decision, RF classifier aggregates the decisions of individual trees; consequently, RF classifier exhibits good generalization. RF classifier tends to outperform most other classification methods in terms of accuracy without issues of overfitting and does not need feature scaling (Donges, 2021).

k-Nearest neighbour classifier

KNN is a nonparametric, supervised machine-learning method used for classification and regression. Predictions are made for a new input (X) by examining the complete training set and summarizing the value or classification of the K nearest instances. Distance-based methods like Euclidean or Manhattan or Minkowski are used to define the K instances in the training dataset that are most similar or “near” in order to predict the output class label of the new input (X) (Pushpa Singh N. S., 2021, pp. 89-111).

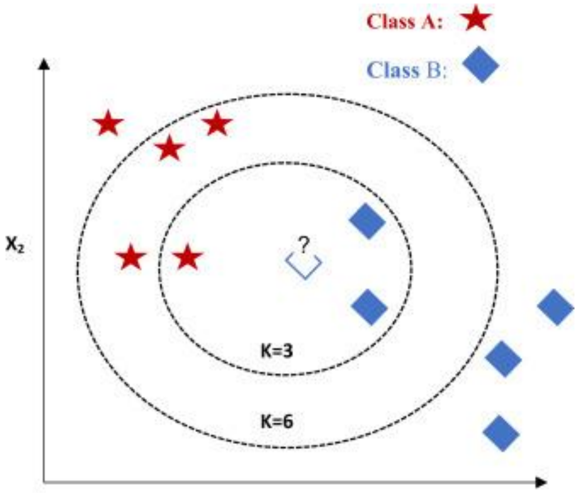


Figure 11: KNN diagram (Pushpa Singh N. S., 2021, pp. 89-111)

If for instance we take the Euclidean example in figure above, for k=3 we get that out new input X with classification unknown has two blue neighbours and one red, it will therefore be classified as blue. On the other hand, if K=5, we get 4 red neighbours and two blue, and the output would be therefore red.

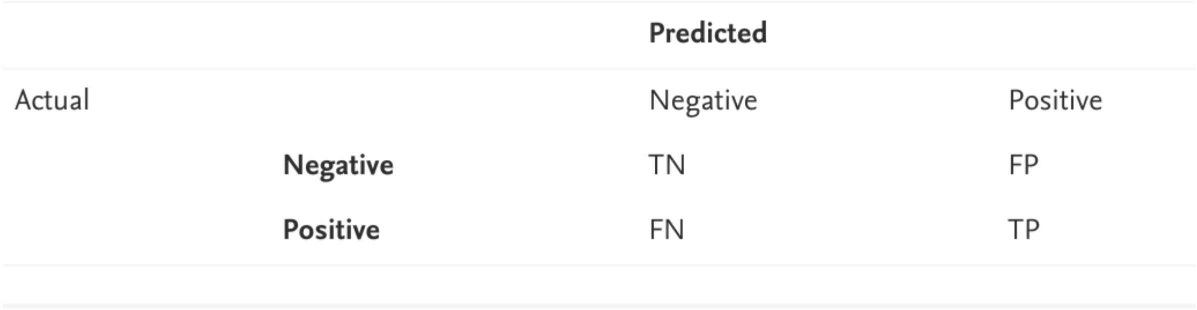
.2.3 Model performance

Evaluating a model performance is a crucial part of model construction, as the metrics obtained from this analysis and the specific needs of the problem to be solved or predicted will determine if the model is valid or not for the purpose it was created. In general, several metrics are used to evaluate model performance depending on the type of problem to solve and the model used. In our specific case, the most common methods used to evaluate ML binary classification model performance are confusion matrix and ROC curve.

Confusion Matrix

Confusion matrices represent counts from predicted and actual values. The output “TN” shown in the figure below stands for True Negative which shows the number of negative examples classified correctly as negative. Similarly, “TP” stands for True Positive which indicates the number of positive examples classified as positive. The term “FP” shows False Positive value, i.e., the number of actual negative examples classified as positive; and “FN” means a False Negative value which is the number of actual positive examples classified as negative.

Table 4: Confusion Matrix Diagram



		Predicted	
Actual		Negative	Positive
	Negative	TN	FP
	Positive	FN	TP

From confusion matrices we obtain these four values, which are later used to compute the metrics which evaluate the model, but they already offer a glimpse of the model performance at first sight. This mentioned metrics are Accuracy, Precision, Recall and F1-Score, which are explained below (Pushpa Singh N. S., 2021, pp. 88-111).

The accuracy of a model is calculated using the given formula below.

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

Equation 1: Accuracy formula

As shown above, accuracy of an algorithm is represented as the ratio of correctly classified patients (TP+TN) to the total number of patients

Regarding precision or specificity of an algorithm, it represents the ratio of correctly classified patients as positive (TP) to the total patients predicted to be positive (TP+FP).

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

Equation 2: Precision formula

This metric gives us knowledge about how many patients are actually positive from the whole that has been classified as positive

Recall or sensitivity metric is defined as the ratio of correctly classified positive patients (TP) divided by total number of patients that are actually positive.

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

Equation 3: Recall formula

The perception behind recall is how many patients have been classified as positive from the total amount of patients that actually are positive. Recall is also called as sensitivity, meaning the ability of the model to predict a positive when analysing one.

F1 score is also known as the F Measure. The F1 score states the equilibrium between the precision and the recall with the formula below:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + recall}$$

Equation 4: F1-Score formula

As seen in Equation 4, F1-score is computed as the harmonised mean between precision and recall. This metric is useful because dumb models predicting unbalanced data can generate an output classifying every sample as the major category, therefore being useless as a classification but presenting good metrics in, for instance, accuracy and recall. Harmonised mean (HM) is an operator that strongly penalises extreme values. This means that if precision and recall are different from each other, as in the situation explained above, the HM will penalise the metric more than the arithmetic mean. Therefore F1-score is more sensible to extreme precision and recall values and gives us more meaningful information of the performance of the model (Tavish Srivastava, 2019).

Receiver Operating Curve (ROC)

ML predictive models usually provide a probability for the classification, instead of an actual classification, and later a threshold is defined to determine at which probability we consider the value to one output or the other. For instance, if we are classifying binary between 0 and 1, the model gives us a probability of 0.82 to be 1, if we put the threshold at 0.8, the output would be classified as 1, but if we put it at 0.9 it would be classified as 0. It is then desirable to determine at which threshold the model classifies better the observations, and in this scenario is where the ROC acquires importance.

A receiver operating characteristics (ROC) graph is a technique for visualizing, organising, and selecting classifiers based on their performance (Fawcett, 2006, pp. 861-874). But an actual binary classifier represented in a ROC would be drawn as a point. The characteristic curve, which usually resembles something like in the figure below, is the different values of sensitivity and 1-Specificity that present the binary classifier for each of the possible thresholds defined. This way we can assess the point, depending on our interests, where we will define the final threshold of the model.

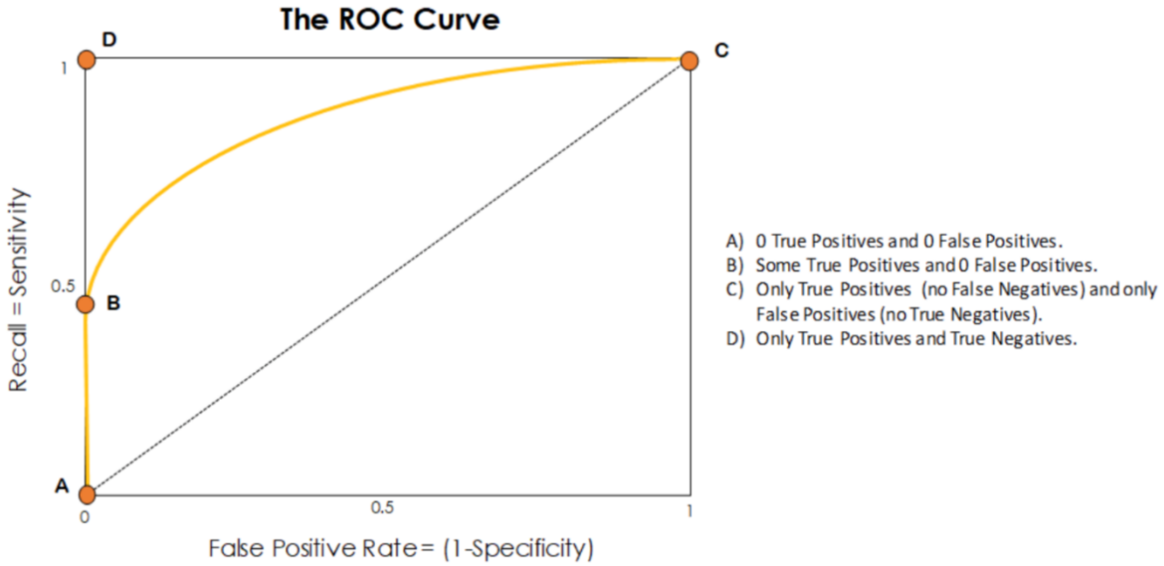


Figure 12: ROC critical points (Zoronza, s.f.)

In the figure above we can observe some critical points and depending on the problems' necessities we will choose a point or another as the final threshold. For instance, if it is not so important to detect all real positives but it is to avoid false positives, point B would be chosen as it never classifies a negative person as positive, while it still correctly classifies some of the real positives.

The most common way to assess the whole performance of the model at all the thresholds is to compute the area under the curve (AUC), which is a parameter that goes from 0 to 1, with 1 being a perfectly accurate model, that is, sensitivity equals 1 even when specificity equals 1, and 0.5 representing half of the predictions right, that is, random probability which is illustrated by a straight line from origin to top right corner in ROC curve shown in figure above.

2.3 State of the art

There has been a surge of research into various neurobiological underpinnings of consciousness in the past decade. The theories regarding this complex phenomenon have not yet come to converge into a standardised deterministic theory that can be used to correctly direct consciousness models.

Most of the approaches regarding the study of consciousness have been developed around the raw EEG signal or derived parameters, such as (Patlatzoglou K, 2020 Jul), who uses 3D Convolutional Neural Networks to assess the state of consciousness of the patient using raw EEG.

Another interesting State of The Art approach comes from (Jesus Pujol, 2021) who proposes fMRI BOLD imaging combined with a repetitive task to assess the effect of some parameters like age, sex and propofol in a LoC survival analysis.

Finally, other approaches that must also be taken into consideration are consciousness related parameters, such as BIS™ (also EEG derived). This parameter does not assess LoC in a particular way but rather consciousness state of the patient in general terms. BIS™ and similar approaches will be also reviewed in next *Section 3. Market Analysis.*

3 Market analysis

Many monitors and indices which allow personalized monitoring of anaesthesia effect are available. Now a days, all the commercialised indices are based on EEG analysis, but with some differences in their approaches of the problem.

The most used from all the indices is the BIS™ index (Aspect Medical, USA), which all we know about its background is that it is calculated by a statistical multivariate model using nonlinear functions of electroencephalography-based subparameters. The actual algorithm of this index is proprietary.

Some other examples of monitoring indices depth of anaesthesia (DoA) are:

- Narcotrend (Narcotrend, Hannover)
- Entropy Module (GEHealthcare, Finland)
- SNAP II (Everest Biomedical, USA)
- Sedline (Hospira, USA)
- Conox (Quantium Medical, Spain)

All the stated above DoA indices use different, usually proprietary algorithms that use EEG derived parameters or signal analysis itself. None of them has intended to assess DoA from a different approach from the first and most hegemonic index, BIS™. Due to this reality, most papers related to these indices try to compare themselves to BIS™ in order to gain a part of its market niche, always with little success, instead of differentiating themselves from BIS™. The use of BIS™ to assess DoA is extended to a point that most monitoring companies offer BIS™ OEM embedded in their anaesthesia workstations, even companies that have their own index to assess DoA.

Comparative analysis of the different commercialised products:

	BIS™
Narcotrend (NT)	Technology: Narcotrend is based on an EEG classification algorithm based on A to F phases and subdivided into 14 subphases as proposed by (Kugler, 1981). The algorithm is then trained with manually classified EEG signals to identify this 14 subphases (B. Schultz, 2002) Performance: Both NT and BIS™ are reliable monitor systems to distinguish different states during maintenance and emergence of propofol/remifentanil anaesthesia accurately, but NT and BIS™ seemed not to be reliable depth-of-anaesthesia monitors, because the analgesic power of anaesthesia is not clearly reflected. (Schmidt, et al., May 2004)
Entropy Module	

	Technology: Analysing entropy of the EEG signal (measurement of disorder) two parameters are obtained, State Entropy (SE) and Response Entropy (RE). The parameters assessed are used to provide useful information regarding the cortical state of the patient as well as an indirect measure of adequacy of analgesia. Performance: Entropy has been found comparable to the current clinical gold standard Bispectral Index (BIS) for various GABA-ergic i.v. induction agents and volatile anaesthetics like propofol and sevoflurane. Entropy guidance may not be used during ketamine or nitrous oxide administration, since there is no reliable correlation to the patient's state of consciousness. (Bein, March 2006)
SNAP II	Performance: The SNAP index can assess a patient's level of consciousness and is correlated to BIS. On the other hand, there is significant bias between the SNAP and BIS indices and therefore, the indices are not interchangeable. The SNAP index returns to baseline before awakening, whereas the BIS index remains below baseline at awakening, suggesting that the SNAP index may be more sensitive to unintentional awareness. (C.A. Wong, 2006)
SedLine	Technology: The PSI index monitor (SedLine) is based on a quantitative analysis of the classical frequency bands (α, β, δ, γ, and θ) after being submitted to a fast Fourier transformation. Both temporal and spatial gradients occur within the frequency bands during different phases of anaesthesia. Four channels are recorded by the PSI monitor to assess such spatial gradients (G.A. Pérez, 2021). Performance: It has been reported that the PSI can predict the arousal level at various stages of the anaesthetic delivery (L.S. Prichep, 2004). One of the advantages of the monitor is that it features four EEG channels which collect information from both sides of the brain. This allows to detect, for example, asymmetry in the electrical activity of the patient.
Conox	Technology: qCON and qNOX are parameters extracted from the raw EEG using an adaptive neuro-fuzzy inference system (ANFIS). These parameters can predict the DoA and the likelihood of movement response to different anaesthetic and surgical stimuli (Gambús PL, Feb 2011). ANFIS is a data driven modelling approach based on the principles of Fuzzy Logic. Performance: As the main difference with BIS and other indices, Conox monitoring offers a double index monitoring: qCON and QNOX. These indices assess the DoA and analgesic depth respectively. The indices qCON and qNOX are able to detect differences between the times of actions of hypnotic and analgesic agents. Also, qCON showed faster decrease during induction while qNOX showed a faster increase during recovery, therefore being useful to assess pain during LoC (Umberto Melia, 2016).

In light of the above, it is visible that most commercialised indices assess DoA, and only partially assess analgesia, and qNOX presents a direct difference in this crucial aspect. Compared to our algorithm, none of these indices directly assesses LoC event.

Furthermore, all the commercialised indices are based on EEG analysis and therefore need EEG monitoring, which is an advanced technique. None of the above indices can predict consciousness related events using basic monitoring signals and parameters, as done in this project. It is therefore straightforward that there is a market niche to the assessment of unconsciousness related measurements without the use of advanced monitoring techniques, specifically EEG.

4 Solution implementation

4.1 Concept engineering

The project is conceived around the interest in knowing the state of consciousness of a patient under general anaesthesia. Knowing this information is of critical value for an intervention under general anaesthesia because knowing the patient is unconscious is determining to start the intervention, as well as to acknowledge that the patient has not experienced anaesthesia awareness. The desired concept is to use widely recorded parameters in surgeries as input variables to generate a new parameter indicating consciousness. A key point to take into consideration is that the desired new parameter regarding patients' consciousness should not represent an extra workload for the OR professionals, nor imply the modification of patients' monitoring habits. This way it is possible to guarantee lower cost and easy implementation of the solution, even its viability.

4.1.1 Different solutions

Solution 1:

The first approach thought was to assess the problem as a classification problem, based on statistical correlational threshold, where some of the recorded parameters with proven correlation to patients' arousal state were studied to know at what values the patient presented arousal. This way a weighted new parameter could be created and validated.

However, this approach presented some issues with the capacity of finding directly extractable features that were significant for the desired output. Also, this approach would be too sensitive to interpersonal variability between patients and could be easily biased when introducing different scenarios as intraoperative drug administration for managing patients' biological constants.

Solution 2:

In the second proposed solution the use of EEG signals recorded through the intervention was assessed. EEG are known to present tight relation with the state of arousal of the patient, and it has been the focus of several approaches to solve the desired problem, as stated in *Section 3*. The use of EEG signal or EEG derived parameters was also taken apart, as EEG monitoring is an advanced monitoring technique and it is not implemented in all the ORs, and it needs a special sensor and a professional to place it. Also, EEG signal processing requires of advanced signal analysis skills which are beyond the time limitations of this project.

Solution 3:

The third solution proposed is to implement a type of human-like learning algorithm, such as ML and DL algorithms, in order to solve the problem. This kind of algorithms not only are capable of learning from historical data, but sometimes they are also capable of extracting the importance of the variables used to determine the outcome. This approach would not only help in the selection of parameters needed to predict the state of arousal, it also is capable of learning from the recorded data from previous operations and usually presents greater performance than simpler statistical models as those proposed in solution 1.

On the other hand, ML and DL algorithms need a lot of data and vast amounts of data respectively, if such requirements are met their performance usually surpasses the ones obtained with other type of models.

4.1.2 Proposed Solution

A simple solution is the key. It must be easily implementable in an OR where professionals already have a lot of work to do, and ultimately hand an easily understandable, significative, and reliable information without imposing changes in the functioning of the OR, the surgery, or the monitoring techniques.

In light of the above, the problem we aim to solve is a conscious/unconscious binary classification problem that must be solved using monitoring parameters generally available in ORs, that do not increment workload for the sanitary professionals and that do not require of any investment neither in machinery nor in extra sensors. Therefore, the proposed solution to solve this problem is to implement a simple python code, which uses as input some generally recorded parameters in ORs, and that is capable of processing this information to introduce it to a ML model and obtain a binary classification value (0 or 1) to assess if the patient is conscious or unconscious. This solution has been chosen over the other two for its simplicity as it does not require investment or intervention of the practitioners. It is also capable of extracting the most valuable parameters to determine the output in a measurable way, and it is not dependent on EEG.

An agreement has not been reached yet about which is the event that precedes actual LoC, as discussed in *Section 2. Background*. In this direction, the recorded event chosen to define unconsciousness in this project is when the patient loses its verbal response. Therefore, what we are actually predicting is the loss of verbal response of the patients, which is a conservative criteria.

Finally, ML algorithms have been chosen over DL algorithms as they are faster to train, simpler to implement and last but not least, they do not require the vast amount of data needed in DL, whose requirements exceeds our records.

4.2 Detail engineering

The first step to implement our solution is to accurately define the problem and our desired results, as shown in *Section 4.1.2*. As the solution proposed is based on a ML predictive algorithm, the second step is to obtain the needed data. In our case the data used has not been specifically recorded for this project, but it is systematically gathered by the anaesthesiology team from OR number 4 as mentioned in *Section 1*. It could be said that we are taking advantage from a dataset which is continuously being recorded by the practitioners.

Once we are clear about what we want to do and by what means, it is crucial for the project to define an objective way to evaluate its results. The basis of some statistical methods for model performance have been mentioned in *Section 2.1.3.2*, so metrics such as ROC AUC, accuracy and F1-Score will assess the performance of the model and therefore the results of the project.

At this point we have a clear intention, the method to obtain the desired results and the parameters used to define project's success. Now we can start to define the actual parts of the project. As seen in figure below, a ML project starts by pre-processing the data to obtain a clearer and easier to handle dataset. Also, this process is iterative with exploratory analysis, as it is important to avoid artifacts in the dataset and keep visualising the final shape the data is acquiring. Once artifacts, such as empty values, have been removed, the data must be processed according to models' necessities and ours, as some models require for instance normalised data. Data is later separated into train and test data. Further explanation on the process is detailed in *Section 4.2.1* below.

When the preliminary steps, pre-processing and exploratory analysis are done, the development of the model starts. First, a feature importance analysis is performed, in order to later select the most significative variables to introduce to our model. Later, the given ML model or models are trained with this data.

Finally, the last step of the project is model testing and documentation writing. As a part of any model generation, after training comes the testing. The testing dataset generated in the data pre-processing step is a part of our data that the model has never seen. This data is used to validate the model: input values are given to the model and then weather the model prediction are correct or not is checked. Here the performance metrics specified in *Section 2.1.3.2* are used to evaluate if the model is performing well. Later, the functioning of the model must be explained in the documentation, as professionals usually do not rely on methods that are not scientifically supported or that have not been validated. In the *sections below, from 4.2.1 to 4.2.4* the steps of the project will be explained in detail.

note: In order to correctly implement functional programming and VCS along all the steps of the project, four python codes have been created as well as different directories to contain patients' CSVs and code outputs, as explained in *Section 1.2.2 Project Structure*.

4.2.1 Data pre-processing and Exploratory analysis

Data acquisition

Regarding data acquisition, the variables recorded are defined in chapter 2.1.2 Anaesthesia monitoring, and "events". A special customizable keyboard was used where events were programmed so that it could be entered with only one keystroke. It allowed a fast entry of events with minimal error. Nevertheless, the data used in the model has not been specially compiled for this project, as mentioned in 1st paragraph of *Section 4.2*. The data used is from a previous study (Marco-Ariño N., 2021 Oct) where the gender covariate factor of the patients in our set of data is female. The data management is controlled by Dr. Pedro Gambús and PhD Student Joan Altés from a safe online cloud server in a Coma Separated Value (CSV) format. Each CSV file is a different patient intervention file, compiling high resolution data at a rate of 1 second per observation (or line). Legal aspects regarding patients' data will be discusses in *Section 7. Legal Aspects.*

Data loading and processing

Firstly, the "libraries.py" archive is created, and all the needed libraries are hosted in there. This archive is later called in the other three in order to import all the libraries as a block avoiding repetition in the code.

The second python archive is now created and named "Train_Test_adquisition.py". The function read_csv()is used to read the file that comprehends all the variables, attributes and characteristics of each patients' file. It is then used to collect all the patients with recorded 'EVENTS' of type 'verb' and 'verbal', which are the names for loss of verbal response in the surgery. Visual Studio Code inline visualization is used for exploratory analysis, as well as plots and histograms to detect distributions over the data.

Later, a for loop is used to iterate on each of these patients' CSV file and process its extracted dataframe. This processing is made mainly through two functions: finder( ) and data_preprocessing(). Trough finder( ) function we obtain each patients' moment of loss of verbal response. Later, this moment is taken by data_preprocessing( ) and used to delimit 60 seconds above and below loss of verbal response. Then, and also inside data_preprocessing( ), the interpolate() function is used to fill gaps of 2 recurrent values or less using a linear interpolation of the values below and above.

Handling empty values (NAN values in python) is done because almost no ML model supports empty values, and if we just eliminate all the observations (lines of the data frame) containing any empty value the data is drastically reduced. Finally, some more empty values are handled, deleting the whole observation in some cases, and filling the gap with the mean value of the variable or 0, depending on each case. For instance, all observations previous to Remifentanil start of infusion are empty, but its information is useful as in most cases Propofol has already been infusing for several seconds, then we just fill the gaps prior to infusion with 0s instead of deleting them. Once NaN handling is done, the data_preprocessing( ) function returns two dataframes: one with all the variables and 120 observations processed of the patient and one with the state of consciousness of the patient for each of the 120 seconds.

Data saving

Finally, each patients' processed data frame is randomly stored in any of the learning or testing data sets and concatenated to the previous patients' information. This dataframes, X_Train, y_train, X_test and y_test, which are data (X) and consciousness state (y) are saved in new CSV files to be read later by the Model.py python script. The archive "Train_Test_adquisition.py" finishes here.

4.2.2 Model training

At this point of the process, we already have the most important checkpoint: the data processed and ready to introduce it to the model for its training. Now the third python file "Model.py" is created. In this file the models will be trained and tested, and later saved for further use.

Dataframe import and model training

Firstly, the four dataframes for train and test of the models are loaded with the function read_csv() from the saved archives in the *section above*. The first model, Support Vector Components is initiated and trained with the fit() function, introducing X_train (processed data from the selected variables used for the model) and y_train (conscious or unconscious state of the patient for each observation of the X_train dataframe). The same is done with the second model, a Random Forest Classifier. The parameters used to tune the models are either selected automatically if the model permits, or iteratively based on which ones show better results. Model outcome is seen to perform similarly independently on the values of the parameters chosen.

Once the model is trained, the predicted values for both of them are computed with the predict( ) function and the predict_proba( ) function. The predict( ) function directly provides the binary outcome prediction (0 or 1), while the predict_proba( ) provides the probability of falling into the class 1. Predicting the probability is interesting in order to perform the ROC curve, as it is based on computing iteratively the True Positive Rate (TPR) and False Positive Rate (FPR) for different thresholds used to consider the output as 0 or 1 (e.g. if the probability of being unconscious or 1 is 0.7 and the threshold is at 0.8, this case would be considered conscious or 0).

4.2.3 Model Validation

Model validation

Model validation is also studied in “Model.py” archive. In order to evaluate the model metrics such as AUC, accuracy and F1-Score have been used. First of all, the ROC curve and its AUC for both models are performed. We can observe in the figure on the right that both models perform generally well, with high AUC values and with high values of TPR for still low values of FPR, meaning that we can identify most unconscious moments without increasing uncontrollably the False Positives. It can be observed that SVC performs a bit better that RFC in terms of AUC.

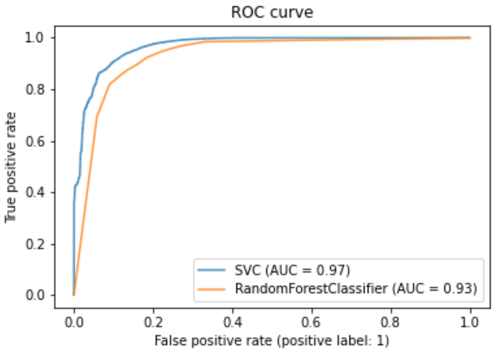
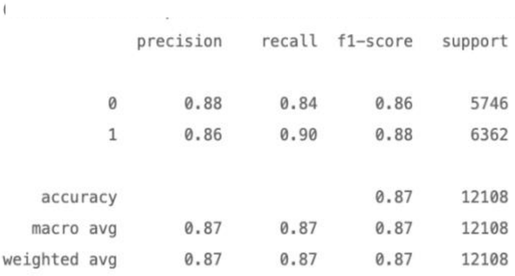


Figure 13: ROC-AUC

Regarding accuracy and F1-Score, both models also perform with high values, all of them above 0.84.

Table 5: RFC metrics

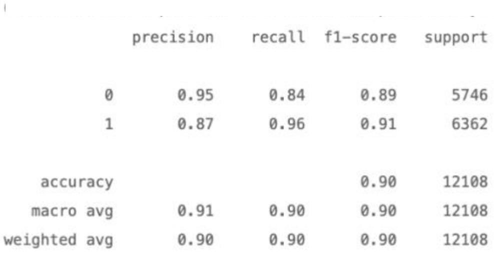


```
              precision    recall  f1-score   support

           0       0.88      0.84      0.86      5746
           1       0.86      0.90      0.88      6362

    accuracy                           0.87     12108
   macro avg       0.87      0.87      0.87     12108
weighted avg       0.87      0.87      0.87     12108
```

Table 6: SVC metrics



```
              precision    recall  f1-score   support

           0       0.95      0.84      0.89      5746
           1       0.87      0.96      0.91      6362

    accuracy                           0.90     12108
   macro avg       0.91      0.90      0.90     12108
weighted avg       0.90      0.90      0.90     12108
```

Concretely, SVC overperforms RFC in precision, recall, F1-Score on both predicting consciousness and unconsciousness, as seen in the tables below.

Also, confusion matrices are useful to visually comprehend how a model is performing. The confusion matrices for both models are shown below.

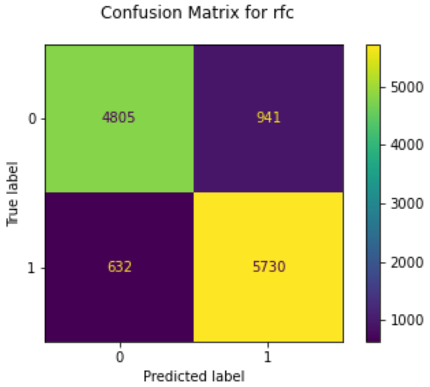


Figure 14: Confusion matrix for RFC

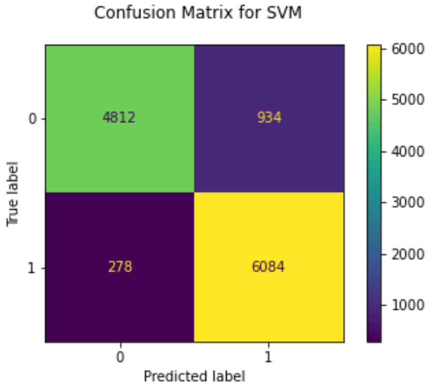


Figure 15: Confusion matrix for SVC

Feature importance

Another feature importance is applied using RFC feature importance characteristic in order to glimpse the weight of each of the variables in the outcome of the model. The results are shown in figure on the right. It is visible that Propofol effect site concentration is the variable that contributes the most to predict the event of LoC, which is consistent with previously expected results.

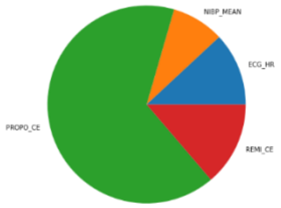
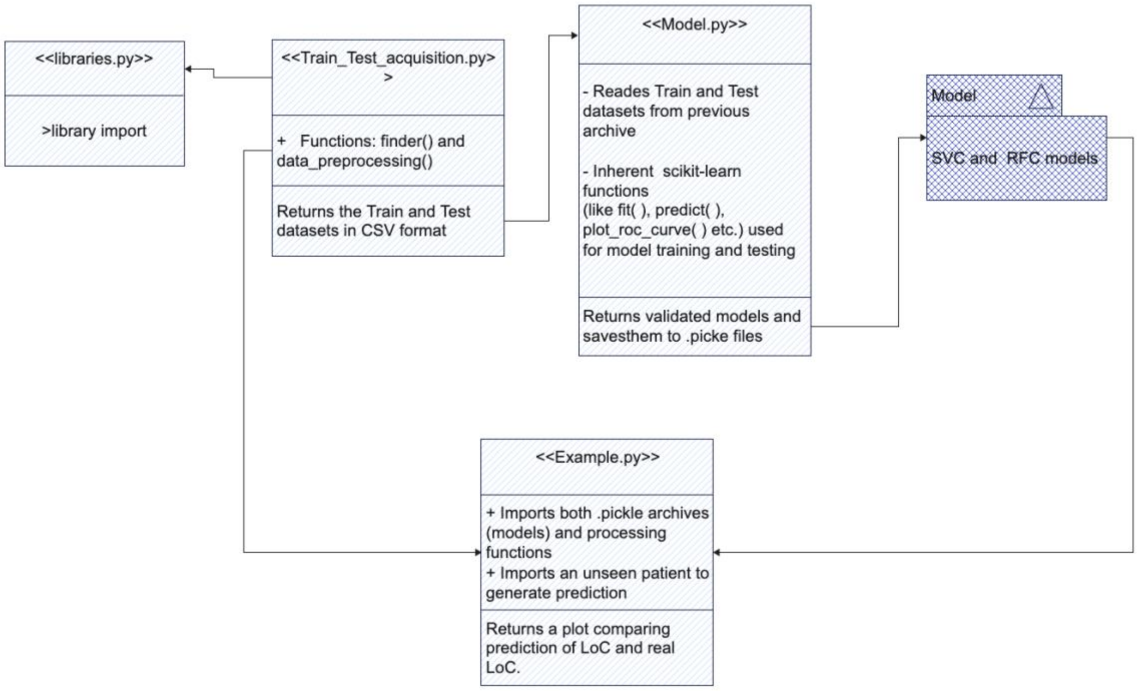


Figure 16: Feature importance analysis with RFC.

Finally, both models are saved in "SVM_model.pickle" and “RandomForest_model.pickle" respectively, using python’s with open file handling methodology, in order to save them for later use. Here ends the “Model.py” archive.

To sum up, in the figure below a diagram of the whole structure of the code is shown.


<<libraries.py>>
>library import
<<Train_Test_acquisition.py>
>
+ Functions: finder() and data_preprocessing()
Returns the Train and Test datasets in CSV format
<<Model.py>>
- Reades Train and Test datasets from previous archive
- Inherent scikit-learn functions (like fit( ), predict( ), plot_roc_curve( ) etc.) used for model training and testing
Returns validated models and savesthem to .picke files
Model
SVC and RFC models
<<Example.py>>
+ Imports both .pickle archives (models) and processing functions
+ Imports an unseen patient to generate prediction
Returns a plot comparing prediction of LoC and real LoC.

.

Figure 17: General overview of the project's code.

4.2.4 Example

The final aim of the algorithm is to predict the moment of LoC of a patient. In order to assess the methodology and performance used to create and validate the model predicting this event the explanation in *Section 4.2 Detailed Engineering* has been developed. On the other hand, it is useful to visually observe what the model outputs around LoC in order to understand what has been done.

In order to do this observation, a patient previously separated from train and test datasets has been used to exemplify what the model is doing, and the archive “Example.py” has been created for this purpose as seen in the figure above.

As explained in *Section 4.2.2 Model Training*, the model returns a value between 0 and 1 for every set of parameters coming from the patient with a ratio of 1 entrance of parameters per second. In the figure below the clinical evaluation of LoC (orange) being 0 conscious and 1 unconscious is compared to the predicted values (blue) of the SVM previously trained model.

As can be observed, the models’ prediction of unconsciousness increases as time (from start of propofol infusion) goes through. And at some point, the model considers the patient is steadily unconscious

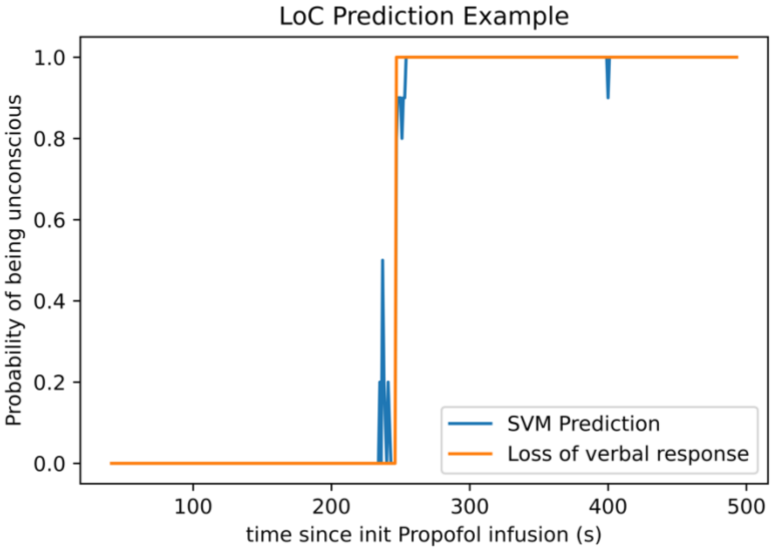


Figure 18: LoC Prediction Example

At this point the model is just predicting a probability for our desired event. An important parameter to assess is the probability at which we consider the event is happening. This is called a threshold. The ROC curves shown in *section 4.2.3 Model Validation* are a graphical representation of the different outcomes the model gives depending on the threshold defined.

The threshold has been therefore chosen at the moment the author sees the false positive rate (1-specificity) starts to increase. This is so because when assessing an unconsciousness prediction, it is not acceptable to predict the patient as unconscious when this is not true, due to its clinical implications. Luckily, the true positive rate (sensitivity) is still high at the needed threshold (~0.85).

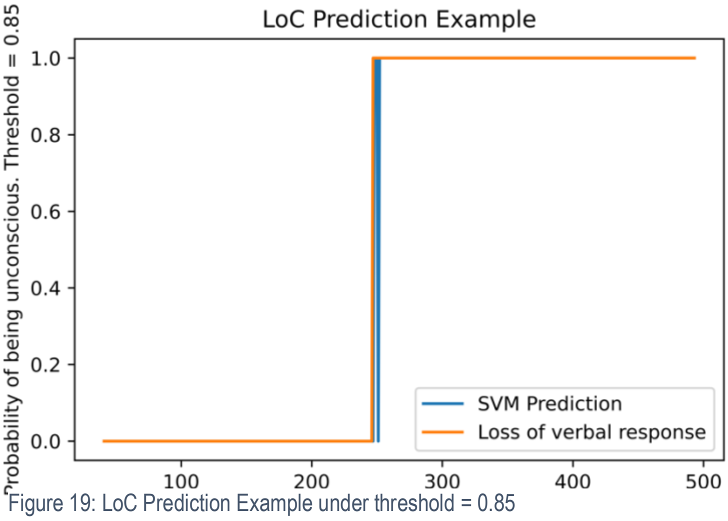


Figure 19: LoC Prediction Example under threshold = 0.85

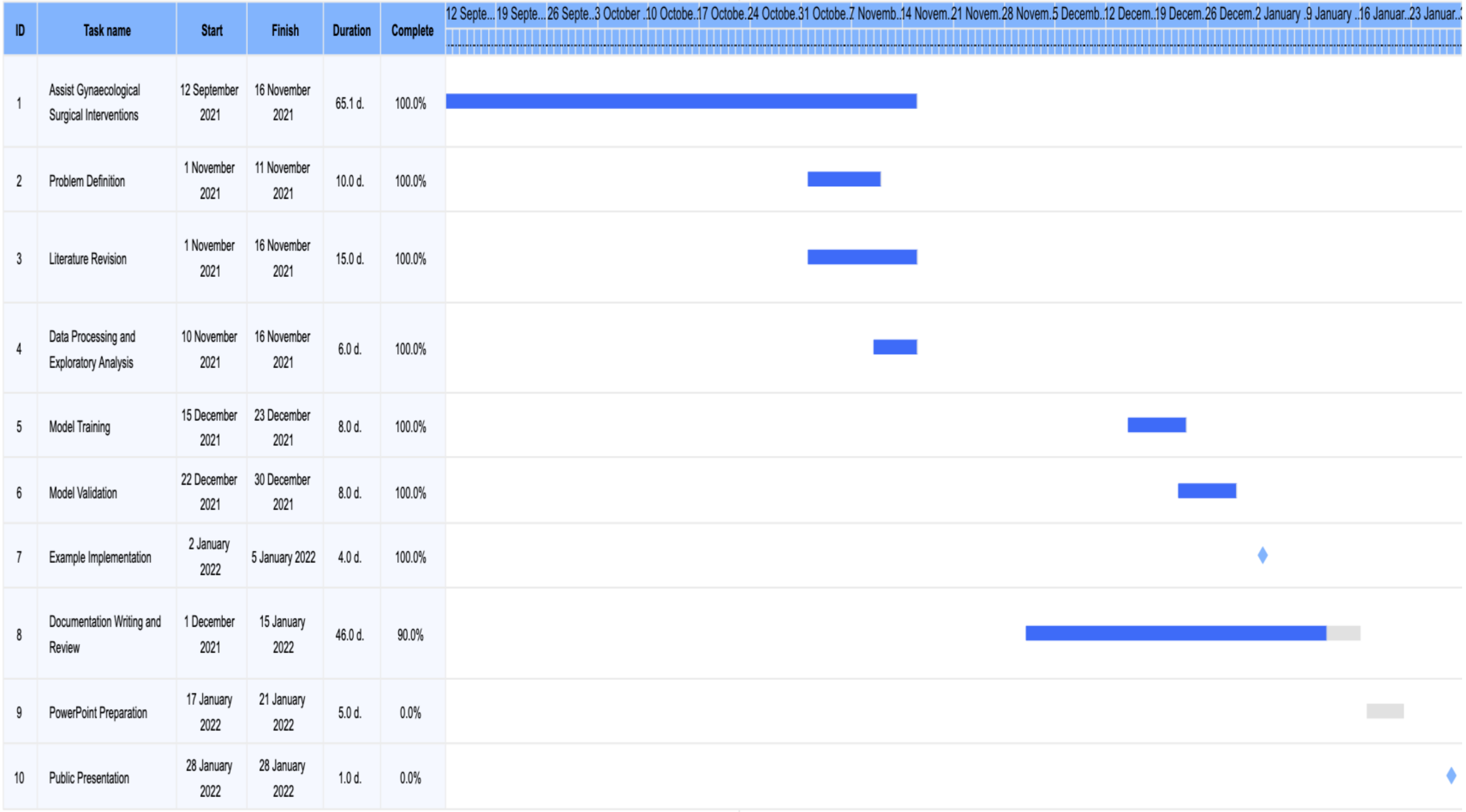
As can be seen in the figure to the left, after applying a threshold equal to 0.85, the prediction of the model is really similar to the actual assessment of the anaesthesiologist.

This is only one example, but visually useful to explain the potential of this kind of algorithms in predicting clinical events.

Time management is one of the key points of a project success. A first glimpse of this project task segmentation is reviewed in *Section1.4.2 Project workload*, and in this section, it is presented in detail.

The Figure below is a GANTT diagram, representing the different tasks forming the project, their time domain, their completion % and a visual time-dependant representation of the mentioned characteristics.

A GANTT diagram is a project managing tool used to generate an overall look on project follow-up, and that pretends to help in managing time. This figure is just an example of the current situation at the moment of writing this documentation.



ID	Task name	Start	Finish	Duration	Complete
1	Assist Gynaecological Surgical Interventions	12 September 2021	16 November 2021	65.1 d.	100.0%
2	Problem Definition	1 November 2021	11 November 2021	10.0 d.	100.0%
3	Literature Revision	1 November 2021	16 November 2021	15.0 d.	100.0%
4	Data Processing and Exploratory Analysis	10 November 2021	16 November 2021	6.0 d.	100.0%
5	Model Training	15 December 2021	23 December 2021	8.0 d.	100.0%
6	Model Validation	22 December 2021	30 December 2021	8.0 d.	100.0%
7	Example Implementation	2 January 2022	5 January 2022	4.0 d.	100.0%
8	Documentation Writing and Review	1 December 2021	15 January 2022	46.0 d.	90.0%
9	PowerPoint Preparation	17 January 2022	21 January 2022	5.0 d.	0.0%
10	Public Presentation	28 January 2022	28 January 2022	1.0 d.	0.0%

Figure 20: GANTT diagram of the project

5 Technical Viability

5.1 Specifications and technical characteristics

The aim of the project is to develop a ML predictive model to assess LoC event under general anaesthesia. At this point of the project, it is interesting to acknowledge the possibilities of the proposed solution to come to a good end.

One of the most famous project management tools used to evaluate the situation of the solution is the SWOT diagram. This analysis tool defines the solution situation regarding solutions' internal attributes (Strengths and Weaknesses) and external world (Opportunities and Threads). As shown in the section below, this SWOT analysis generates a general overview on the solution regarding its real situation, and facing it to not only its capacities, but also to regarding external factors.

5.2 SWOT

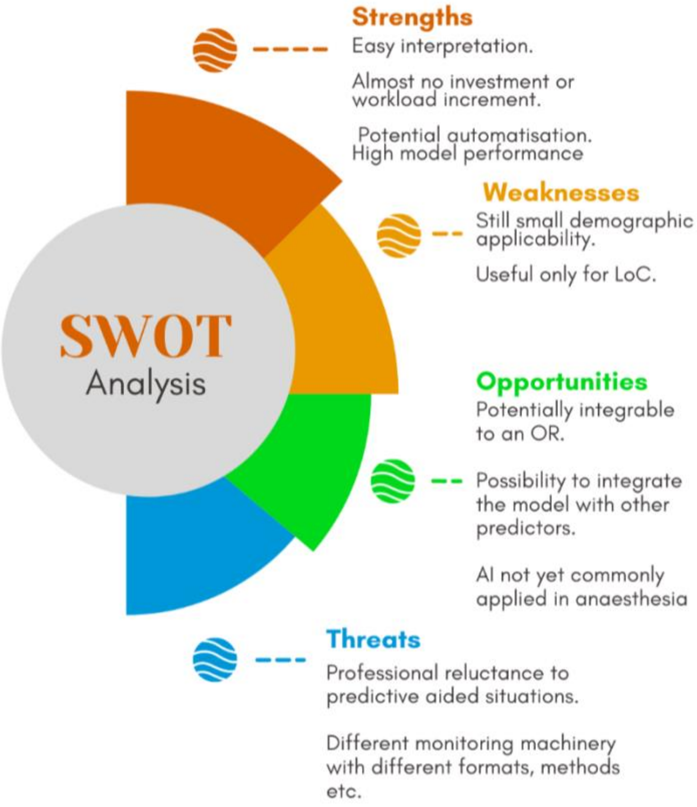

SWOT
Analysis
Strengths
Easy interpretation.
Almost no investment or workload increment.
Potential automatisation.
High model performance
Weaknesses
Still small demographic applicability.
Useful only for LoC.
Opportunities
Potentially integrable to an OR.
Possibility to integrate the model with other predictors.
AI not yet commonly applied in anaesthesia
Threats
Professional reluctance to predictive aided situations.
Different monitoring machinery with different formats, methods etc.

Figure 21: SWOT Analysis

As can be seen on the figure, the main strengths of the project go around its simplicity for interpretation and implementation, as it uses typically recorded variables in ORs. It also discusses opportunities in the sense that, even though it is still restricted to a small demographic group and only useful to predict LoC as discusses in *Section 8.2 Limitations*, the model is potentially integrable with other predictors inside an OR.

Finally, external threads such as professionals' reluctance to incorporate predictive algorithms to aid in clinical decisions are a barrier for the model incorporation to the clinical environment. Another huge thread is the fact that even though needed variables are always recorded in almost every intervention, data format and handling is usually different for most monitoring commercial systems, and integration is not that easy

6 Economic viability

In order to correctly manage a project, economic studies of the costs must be performed, as usually a pre-defined budget and an idea is all we have, and it is crucial to delimitate the project to an extent we can economically assume to avoiding the undesired scenario of having to stop the project before it ends.

Regarding the Economical point of view, the project has three different types of expenses. First of all, the Hardware expenses, which include all the material needed to analyse the data and do the project. In here, none of the expenses derived from data acquisition like the OR, human resources from healthcare professionals, monitoring systems and sensors etc. have been taken into consideration because these are expenses the hospital would have independently of the realisation of this project.

Table 7: Hardware Budget

	Hardware
	Cost (€)
MacBook Air Computer	1130
Data acquisition	0
	1130

On the other hand, all the software machinery used to do the project is either of open source, or free to use, with the great exception of a diagram web service called Canva, which has a price of 110 euros per year as shown in the Table below.

Table 8: Software Budget

	Software
	Cost
Python Programming	Open source
ML packages	Open source
Canva diagram generator	110 €/year
Visual Studio Code	Free
	110

Finally, the table below is an estimation of the cost of the human resources used in the project, which is 5500 euros.

Table 9: Human Resources Budget.. Business cost of labour per hour has been used, instead of workers' labour per hour salary for more accurate results

	Human Resources		
	Cost (€ /hour	Hours	Cost
Biomedical Engineering Student	10	350	3500
Anaesthesiologist MD	35	40	1400
Undergraduate PhD Student	20	30	600
			5500

It is therefore straightforward that the total cost of implementation of this project is of 6740 euros.

7 Legal aspects

Regarding regulatory issues, the product derived from this project belongs to a class of medical products called Software as Medical Device (SaMD) which is defined by the International Medical Device Regulators Forum (IMDRF) as "software intended to be used for one or more medical purposes that perform these purposes without being part of a hardware medical device."

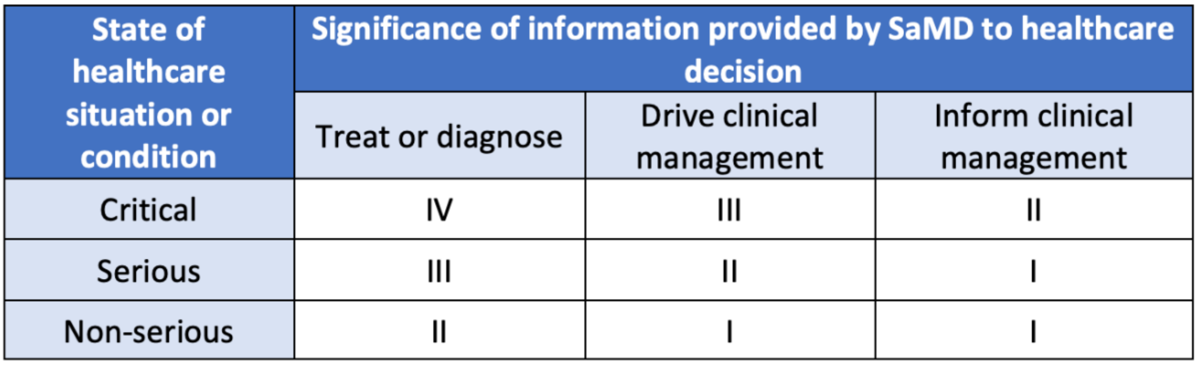
The Food and Drug Administration of the United States (FDA) considers AI based SaMD as a special paradigm inside SaMD. The IMDRF SaMD risk categorization framework takes a risk-based approach to categorize SaMD based on intended use, similarly to traditional risk-based approaches used by the FDA. In fact, the risk categorization is based on these two factors (FDA, 2021):

1) Significance of information provided by the SaMD to the healthcare decision, which identifies the intended use of the information provided by the SaMD – i.e., to treat or diagnose; to drive clinical management; or to inform clinical management.

2) State of healthcare situation or condition, which identifies the intended user, disease or condition, and the population for the SaMD – i.e., critical; serious; or non-serious healthcare situations or conditions.

The health care condition in an intervention under general anaesthesia would be considered between non-serious and serious. On the other hand, the significance of information provided by our algorithm is to inform clinical management but could also be to drive the clinical management at some point. It is therefore straightforward that the risk categorization for our product is Class I and Class II, as seen in the Table below.

Table 10: SaMD IMDRF risk categorization (FDA, 2021).



State of healthcare situation or condition	Significance of information provided by SaMD to healthcare decision		
	Treat or diagnose	Drive clinical management	Inform clinical management
Critical	IV	III	II
Serious	III	II	I
Non-serious	II	I	I

Following the recommendations for Class I and II products approval, our SaMD requires an evaluation of: performance, used inputs and intended use. All these elements that have been covered in this project. FDA also enhances good machine learning practices (GMLP), that include assessing the relevance of the data used to train the model for the specific application and an appropriate level of transparency and understandability of generated outputs towards the users.

Regarding European Union regulations, SaMD is not specifically treated inside European Medicines Agency (EMA), but our project must match a more general regulation for Medical Device Applications called Medical Device Regulation (MDR). This MDR standards also perform a risk categorization based on the intended purpose of the devices and their inherent risks and, in general terms, the MDR requirements match those of the FDA.

Once the requirements are met the product should not have problems to pass the regulations, even though further studies in biased training should first be successfully accomplished in order to obtain the approvals.

8 Discussion and Future Prospects

8.1 Implications

The point that SVC and RFC models show good overall performance in predicting the LoC event of a patient during anaesthesia induction means several benefits from a clinical point of view.

First of all, the clinical support to practitioners with a model being trained with huge amounts of historical data goes in the direction of personalised medicine, if the model is correctly trained with balanced and unbiased data. This way practitioners can obtain a fast and unbiased information regarding LOC coming from already collected data that has been analysed to provide predictive ability.

Finally, the same digitalisation of this duty is interesting from the process automation point of view. Automation and prediction do not mean taking away decision abilities from the attending anaesthesiologist but rather providing another source of information to support the process of decision making. Even though prediction of intraoperative events can be of clinical usefulness it does not pretend to replace practitioners' work, but rather improve surgical performances and practitioners' comfort. Also, these predictive capacities constitute an initial step towards a partially automated anaesthesia process, that even though it would never be desirable, could be of crucial interest in extreme situations where no medical professionals are available, such as in war zones, remote locations, spatial stations etc.

8.2 Limitations

The limitations of a project are defined as the barriers and difficulties acknowledged during or even prior to execution that were not possible to handle or overcome. Limitations are boundaries or problems that have made some goals of the project totally or partially unachievable. In our specific case and given that our project is based in data analysis and model generation, it would be expected that most of the limitations were found around the quality, structure, composition, or origin of the data.

As mentioned above, the first limitation that came up when achieving our goals was the origin of the data. Due to the nature of the OR where our patients' data is collected, the samples used to both train and test the model are composed mainly by individuals of female sex, from Mediterranean origin and of advanced age. This clear under representation of male sex, different ethnics and ages limits the possibility of model generalisation. In order to evaluate the models' applicability to individuals of different characteristics than the ones used to train and test the model further validation tasks should be performed.

On the other hand, the model is not simply a LoC model, but a ML model trained to predict loss of verbal response, as a surrogate measure of LOC event during induction of general anaesthesia. This means that as the samples used to train the model where from 60s before and after the patients fell unconscious, the applicability of the model to assess consciousness state of the patient at any other moment of the intervention rather than anaesthesia induction would not correspond without previous validation. Also, the model does not predict LoC, but loss of verbal response as a surrogate measure of LOC. This is due to the limitation of LoC concept, which has no scientifically defined measurable expression, and therefore loss of verbal response is the used to assess LoC.

Furthermore, the inputs of the model, which go from NIBP to HR and Remifentanil and Propofol effect site concentrations limit the types of general anaesthesia where the model can be used. It is therefore straightforward that the model is limited to TIVA-TCI Propofol and Remifentanil mediated general anaesthesia. In the same direction, another limitation of the project is the necessity of software integration, as the four input variables mentioned above come from Anaesthesia monitor (NIBP and HR) and from the TCI automatic pumps (Ce_{Propo} and Ce_{Remi}). Without the integration of this information, it is not possible for the model to compute the predictions.

Finally, the last limitation is regarding the selected variables used to train the model. The first selection was done under the supervision of Dr Pedro Gambús, who has a wide knowledge on the topic and mentored the most logical selection. This means that the initial feature selection was not properly handled by feature importance techniques or by statistical evidence but rather informally.

8.3 Recommendations and Future Prospects

Future studies around this project should start by performing a prospective validation of the model with balanced data, as mentioned in *Section 8.2. Limitations* in paragraph 2, in order to perform a reliable generalisation on a wider population. Performance of the model should be given in ROC AUC and results showed with confidence intervals.

Another recommendation for future implementation is the study of DL algorithms to perform the same predictions achieved with ML in this project and compare performances. As data gathering is a continuous process done in the OR number 4, it is possible that in some time enough data will be gathered to adequately train and validate DL models with high complexity structures.

Some more future implementations related to patients' consciousness such as anaesthesia awareness could also be studied with little modifications of the proposed models, and it would be of clinical interest to prophylactically predict this undesirable scenario in surgical interventions. Other intraoperative events such as patient movement, BAR suppression and hypotension between others could be or are already being modelled with similar approaches to the one proposed, and to prototype an integrated system of this predictions into the OR infrastructure is a future prospect to take into consideration.

Finally, introduction of new data such as health habits, comorbidities and data from same patients' past interventions would be an improvement in current personalised medicine approaches. Also, feature engineering to automatically select important variables to perform predictions is a strong recommendation that, even though it has not been implemented in this project, is part of good AI programming practices to improve models' performance.

9 Bibliography

1. Siddiqui BA, K. P. (2021, January). Anesthesia Stages. Retrieved from NCBI: https://www.ncbi.nlm.nih.gov/books/NBK557596/
2. Emery N. Brown, M. P. (2010). General Anesthesia, Sleep, and Coma. NIH. Retrieved from NCBI: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3162622/#R10
3. Guarracino F, L. F. (2005). Target controlled infusion: TCI. EDIZIONI MINERVA MEDICA.
4. Sebel PS, B. T. (Anesth Analg. 2004). The incidence of awareness during anesthesia: a multicenter United States study. Retrieved from PubMed: https://pubmed.ncbi.nlm.nih.gov/15333419/
5. Smith G, D. J. (2021). General Anesthesia for Surgeons. StatPearls. Treasure Island (FL): StatPearls Publishing;.
6. Guarracino F, L. F. (2005). Target controlled infusion: TCI. Minerva Anestesiol. .
7. Mertens MJ, O. E. (2003). Propofol reduces perioperative remifentanil requirements in a synergistic manner: response surface modeling of perioperative remifentanil-propofol interactions. Anesthesiology.
8. Jason A. Campagna, M. P. (2003). Mechanisms of Actions of Inhaled Anesthetics . N Engl J Med .
9. Hemmings HC Jr, A. M. (2005). Emerging molecular mechanisms of general anesthetic action. Trends Pharmacol Sci.
10. Srinivasa N Raja, D. B.-J. (2020). The revised International Association for the Study of Pain definition of pain: concepts, challenges, and compromises. nternational Association for the Study of Pain.
11. Ramakrishnan S, D. J. (2021). Akinesia. StatPearls Publishing, Treasure Island (FL).
12. Roop Kaw, M. M. (2014). Obesity Hypoventilation Syndrome and Anesthesia Considerations.
13. J.AndersonJamesHoughton, B. (2019). Total Intravenous Anesthesia and Target-Controlled Infusion.
14. Brian J.Anderson, J. H. (2019). Total Intravenous Anesthesia and Target-Controlled Infusion.
15. Folino TB, M. E. (2021). Propophol. Treasure Island (FL): StatPearls Publishing;.
16. P. Guldenmund, I. G. (2015). Propofol-Induced Frontal Cortex Disconnection: A Study of Resting-State Networks, Total Brain Connectivity, and Mean BOLD Signal Oscillation Frequencies.
17. Marieke Kruidering-Hall, P. L. (2016). Skeletal Muscle Relaxants. Retrieved from https://basicmedicalkey.com/skeletal-muscle-relaxants/
18. Yasar Sattar, L. C. (2021). Electrocardiogram . Treasure Island (FL): StatPearls Publishing; .
19. Neuromuscular blocking agents. (2021). Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Neuromuscular-blocking_drug#Non-depolarizing_blocking_agents
20. Altès, J. (2020). Machine learning prediction of adverse events in anaesthesia after laryngeal mask airway insertion.
21. Torp, K. D., Modi, P., & Simon., L. V. (2021). Pulse oximetry. StatPearls Publishing LLC.
22. Pandya, N. K., & Sharma., S. (2021). Capnography And Pulse Oximetry. StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing;.
23. Shaker, M. M. (2003). EEG Waves Classifier using Wavelet Transform and Fourier Transform . World Academy of Science, Engineering and Technology 3.
24. Carlos Rogério Degrandi Oliveira, W. M. (2017 Jan-Feb). Benefit of general anesthesia monitored by bispectral index compared with monitoring guided only by clinical parameters. Systematic review and meta-analysis. Braz J Anesthesiol. .
25. Pospeev, L. (2019, October 15). pospeev.com. Retrieved from https://www.pospeev.com/post/types-of-mathematical-models

26. Parthasarathy, S. (2021, September 16). logianalytics.com. Retrieved from https://www.logianalytics.com/predictive-analytics/what-is-predictive-analytics/
27. Tavish Srivastava. (2019, August 6). 11 Important Model Evaluation Metrics for Machine Learning Everyone should know. Retrieved from www.analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/#h2_17
28. Zoronza, J. (n.d.). The ROC Curve: Unveiled. Retrieved from towardsdatascience.com: https://towardsdatascience.com/the-roc-curve-unveiled-81296e1577b
29. Donges, N. (2021, September 16). A Complete Guide to the Random Forest Algorithm. Retrieved from Built in Beta: https://builtin.com/data-science/random-forest-algorithm
30. Wilson, D. G. (2017). Pharmacokinetics of TIVA/TCI. UCT Department of Anaesthesia & Perioperative Medicine.
31. Noble desktop. (2021, October 6). Retrieved from https://www.nobledesktop.com/learn/git/git-branches
32. IMDRF. (n.d.). Retrieved from https://www.imdrf.org/
33. FDA. (2021). fda.gov. Retrieved from US FDA Artificial Intelligence and Machine Learning Discussion Paper: https://www.fda.gov/files/medical%20devices/published/US-FDA-Artificial-Intelligence-and-Machine-Learning-Discussion-Paper.pdf
34. Patlatzoglou K, C. S. (2020 Jul). Generalized Prediction of Unconsciousness during Propofol Anesthesia using 3D Convolutional Neural Networks. . Biol Soc. .
35. Jesus Pujol, L. B.-H.-V.-S.-C. (2021). Largest scale dissociation of brain activity at propofol-induced loss of consciousness. Sleep Research Society.
36. Schmidt, G. N., Bischoff, P. M., Standl, T. M., Lankenau, G. M., Hilbert, M., & am Esch, J. S. (May 2004). Comparative Evaluation of Narcotrend™, Bispectral Index™, and Classical Electroencephalographic Variables During Induction, Maintenance, and Emergence of a Propofol/Remifentanil Anesthesia. Anesthesia & Analgesia.
37. Bein, B. (March 2006). Monitoring Consciousness. In B. B. MD, Best Practice & Research Clinical Anaesthesiology (pp. 101-109).
38. C.A. Wong, R. F. (2006). A comparison of the SNAP II™ and BIS XP™ indices during sevoflurane and nitrous oxide anaesthesia at 1 and 1.5 MAC and at awakening, . British Journal of Anaesthesia, 181-186.
39. Kugler, J. (1981). Elektroenzeophalographie in Klinik und Praxis : eine Einführung. Stuttgart ; New York : Thieme.
40. B. Schultz, U. G. (2002). Automatic Classification Algorithms of the EEG Monitor Narcotrend for Routinely Recorded EEG Data from General Anaesthesia: a Validation Study.
41. G.A. Pérez, J. P. (2021). Modelling the PSI response in general anesthesia. Journal of Clinical Monitoring and Computing, vol. 35, pages1015–1025.
42. L.S. Prichep, L. G. (2004). The Patient State Index as an indicator of the level of hypnosis under general anaesthesia, . British Journal of Anaesthesia, Volume 92, Issue 3, Pages 393-399,.
43. Umberto Melia, E. G. (2016). Comparison of the qCON and qNOX indexes for the assessment of unconsciousness level and noxious stimulation response during surgery. Barcelona.
44. Gambús PL, J. E.-C. (Feb 2011). Modeling the effect of propofol and remifentanil combinations for sedation-analgesia in endoscopic procedures using an Adaptive Neuro Fuzzy Inference System (ANFIS). Barcelona: Anesthesia Analg.
45. Pushpa Singh, N. S. (2021). Chapter 5 - Diagnosing of disease using machine learning. Subchapter 5.6.1. Support vector machine (SVM). In N. S. Pushpa Singh, Machine Learning and the Internet of Medical Things in Healthcare.
46. Khurram Saleem Khan, F. F. (Sept. 2013). Pharmacology of anaesthetic agents I: intravenous anaesthetic agents. In Continuing Education in Anaesthesia Critical Care & Pain (pp. 100-105).

47. Sergio Vide, C. N. (2020). Chapter 7: Hypnotic Effect: Inducing Unconsciousness and Emergence. In J. F. Pedro L. Gambús, Personalized Anaesthesia: Targeting Physiological Systems for Optimal Effect (p. 113). Cambridge Universoty Press.
48. Mathur S, P. J. (2021 Jan). Bispectral Index. StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing;.
49. Flores FJ, H. K. (Aug 2017). Thalamocortical synchronization during induction and emergence from propofol-induced unconsciousness. Proc Natl Acad Sci U S A.
50. Absalom AR, G. J. (Jan. 2016). Target-Controlled Infusion: A Mature Technology. Anesth Analg., 122(1):70-8.
51. Hemmings HC Jr, A. M. ((2005)). Emerging molecular mechanisms of general anesthetic action. Trends in pharmacological sciences.
52. Robert A. Veselis, M. [. (2009). Propofol and midazolam inhibit conscious memory processes very soon after encoding: An event related potential study of familiarity and recollection in volunteers.
53. Robert A. Veselis, M. [. (2009). Propofol and midazolam inhibit conscious memory processes very soon after encoding: An event related potential study of familiarity and recollection in volunteers. Anaesthesiology.
54. Rudolph, U. A. (2004). Molecular and neuronal substrates for general anaesthetics. Nat Rev Neurosci, 709–720.
55. Brian J. Anderson, J. H. (2019). Total Intravenous Anesthesia (Tiva)- A Brief Review. In A Practice of Anesthesia for Infants and Children (Sixth Edition) (pp. 177-198). Elsevier.
56. Struys MM, D. S. (Jan. 2016). The History of Target-Controlled Infusion. Anesth Analg., 122(1):56-69.
57. Engbers FH, S. N. (Feb. 2010). Pharmacokinetic models for propofol: defining and illuminating the devil in the detail. British Journal of Anaesthesiology, 104(2):261-2.
58. Sergio Vide, C. N. (2020). Hypnotic Effect: Inducing Unconsciousness and Emergence from Anaesthesia. In J. F. Pedro L. Gambús, Personalized Anaesthesia Targeting Physiological Systems for Optimal Effect (p. 105). Cambridge Universoty Press.
59. Butterworth, o. F. (2013). Neuromuscular Blocking Agents. McGraw Hill.
60. Patrick L. Purdon, A. S. (2015). Clinical Electroencephalography for Anesthesiologists: Part I: Background and Basic Signatures. Anesthesiology.
61. Marco-Ariño N, V. S.-A.-M. (n.d.). Semimechanistic models to relate noxious stimulation, movement, and pupillary dilation responses in the presence of opioids.
62. Marco-Ariño N., V. S.-A.-M. (2021 Oct). Semimechanistic models to relate noxious stimulation, movement, and pupillary dilation responses in the presence of opioids. CPT Pharmacometrics Syst Pharmacol.
63. Jeanne, M. (2009). Heart rate variability during total intravenous anesthesia: effects of nociception and analgesia. Autonomic Neuroscience.
64. Gbenga Ogedegbe, M. a. (2010). Principles and techniques of blood pressure measurement. Cardiology clinics, vol 28.
65. Patrick L. Purdon, A. S. (2015). Electroencephalography for Anesthesiologists: Part I: Background and Basic Signatures. Anesthesiology.
66. Siddharth Misra, H. L. (2020). Chapter 9 - Noninvasive fracture characterization based on the classification of sonic wave travel tim. Subchapter 4.1.4. Random forest (RF) classifier. In Machine Learning for Subsurface Characterization (pp. 243-287). Gulf Professional Publishing,.
67. Pushpa Singh, N. S. (2021). Chapter 5 - Diagnosing of disease using machine learning. Subchapter 5.6.2. K-nearest neighbors (KNN). In Machine Learning and the Internet of Medical Things in Healthcare. Academic press.
68. Pushpa Singh, N. S. (2021). Chapter 5 - Diagnosing of disease using machine learning. Subchapter 5.5. Confusion matrix. In Machine Learning and the Internet of Medical Things in Healthcare.

69. Fawcett, T. (2006). An introduction to ROC analysis,. In Pattern Recognition Letters.

10 Annex

“Libraries.py” archive contains all the needed libraries in order to compile the code.

libraries.py

```
#Importing common libraries
from numpy import *
import pandas as pd
import matplotlib.pyplot as plt
from os import chdir
from pandas.io.parsers import read_csv
from scipy.interpolate import interp1d
import random
import pickle

# Import datasets, classifiers and performance metrics
from sklearn.metrics import ConfusionMatrixDisplay, roc_auc_score, roc_curve, accuracy_score, auc
from sklearn import datasets, svm, metrics
from sklearn.model_selection import train_test_split
from os import chdir, getcwd

print('libraries imported')
```

In this Train_Test_Matrices.py archive the X_train, y_train, X_test and y_test matrices are generated and saved for later use.

Train_Test_acquisitions.py

```
from libraries import *


path = '/Users/marcpalomercadenas/Desktop/TFG/TFG/archives/VARIABLES.xlsx'
source = pd.read_excel(path)

#Input feature selection
selected_features = ['ECG_HR', 'NIBP_MEAN','PROPO_CE', 'REMI_CE']

#Function to find the index and the time where LoC occurs
def finder(patient_dataframe, word1, word2):

    patient_dataframe.reset_index(drop = True)

    #We initialize the local variables
    idx = []
    a = 0
    b = 0

    #We search for "verb'" inside the event feature
    try:
        idx = list(patient_dataframe['EVENT']).index(word1)
    except ValueError:
        a = 1

    #We search for "verbal" inside the event feature
    try :
        idx = list(patient_dataframe['EVENT']).index(word2)
    except ValueError:
        b = 1


    #If neither of the options has been useful to detecting the verbal response nothing is returned
    if a*b == 1:
        idx = []
        LoC_time = 0
    else:
        LoC_time = patient_dataframe['TIME'].loc[idx]


    return [idx, LoC_time]

#Data preprocessing before entering it to the model
def data_preprocessing(patient_dataframe,idx,LoC_t,selected_features):

    #Busquem el index del primer valor de propo i si no hi és descartem el pacient
    fvi = patient_dataframe['PROPO_CE'].first_valid_index()


    #Patient is only valid it it has propo induction and LoC happend after propo induction
    if fvi != None and idx > fvi:

        #Cast of the df
        a = patient_dataframe.loc[fvi:(2*idx-fvi),:]


        #Interpolatio  Nans
        a.interpolate(method='linear', limit_direction='forward', axis=0, inplace = True)

        #LoC binarisation (creating a vector of 0 until negative verbal response, and the other are 1)
        index_LoC = list(a['TIME']).index(LoC_t)
        a.reset_index(drop = True, inplace = True)
        LoC_index = list(a['TIME']).index(LoC_t)
        a['LoC']=0
        a.loc[LoC_index:,'LoC']=1

        #Vdf of wanted variables
        a = a.loc[:,['ECG_HR', 'NIBP_MEAN','PROPO_CE', 'REMI_CE', 'LoC', 'TIME']]

        #NaN handling
        a.loc[:,'REMI_CE'] = a.loc[:,'REMI_CE'].fillna(0)
        a = a.dropna()
        a.reset_index(drop = True, inplace = True)

        #Variable assignation
        LoC = a['LoC']
        Time = a['TIME']
        a = a.loc[:,selected_features]

    else:
        a = pd.DataFrame(columns= selected_features)
        LoC = pd.DataFrame()
        Time = pd.DataFrame()



    return [a, LoC, Time]

#Variable initialization
general_dataframe = pd.DataFrame()
X_train =[]
X_test = []
y_train = []
y_test = []
available_event = 0
no_event = 0
e = 0
available_index = 0
available_patient = 0

#Extracting patients df
for patient_ID, event_state in zip(source['ID'],source['EVENTS']):
    if event_state == 'AVAILABLE':

        #Counting the patients with record of EVENTS
        available_event = available_event +1

        #Reading the current archive of the patient
        path = '/Users/marcpalomercadenas/Desktop/TFG/TFG/archives'
        chdir(path)
        archive = pd.read_csv(patient_ID+'.csv')

        #Finder returns the index of the verbal response
        index, LoC_t = finder(patient_dataframe = archive, word1 = "verb'", word2 ="verbal")

        #If there is index of verbal response, there is verbal response
        if index != []:
            available_index = available_index+1

            #We cut the patients' dataframe info  +/- 2 minutes from the lose of verbal response
            frame, LoC, Time = data_preprocessing(archive,index,LoC_t,selected_features)
            #We append the 240 seconds patients' dataframes for later concatenation and visualization
            pd.concat([general_dataframe, pd.concat([frame, LoC], ignore_index = True)], ignore_index=
True)

            #If the patient had verb resp on propo+remi, the dataframe shouldn't be empty:
            if len(frame) != 0:
                available_patient = available_patient +1


                #Data scaling. There are a lot more types of scaling
                from sklearn.preprocessing import MaxAbsScaler
                scaler = MaxAbsScaler()
                frame = scaler.fit_transform(frame)

                # Split data into train and test subsets, randomly and putting whole patients in train or test,
to avoid overfitting.
                x = random.uniform(0,1)
                if x < 0.5:

                    X_train.append(frame)
                    y_train.append(LoC)
                else:
                    X_test.append(frame)
                    y_test.append(LoC)


            else:
                e = e+1
    else:
        no_event = no_event +1


print(f'There are {available_index}  patients from {available_index + no_event} with verbal response
recording')
print(f'There are {round((available_patient)*100/available_index,1)}% of patients from the
{available_index} with recorded neg. verbal response which data is handled')

#Concatenation the patients' data for model training
X_train = concatenate((X_train))
X_test = concatenate(X_test)
y_train = concatenate(y_train)
y_test = concatenate(y_test)

#Saving the matrices to csv so that the results don't change constantly
pd.DataFrame(X_train).to_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/X
_train.csv')
pd.DataFrame(X_test).to_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/X_
test.csv')
pd.DataFrame(y_train).to_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/y_
train.csv')
```

```
pd.DataFrame(y_test).to_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/y_test.csv')
```

In the Model.py archive generates and validates the SVC and RFC models, later they are saved into pickle archives for later use.

Model.py

```
#Importing libraries
from libraries import *


#Reading train and test dataframes
X_train = pd.read_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/X_train.csv', index_col = 0)
X_test = pd.read_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/X_test.csv', index_col = 0)
y_train = pd.read_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/y_train.csv', index_col = 0)
y_test = pd.read_csv('/Users/marcpalomercadenas/Desktop/TFG/TFG/TrainTestMatrices/y_test.csv', index_col = 0)

# Create a classifier: a support vector classifier
SVM = svm.SVC(gamma='scale', probability= True) #, probability = True
SVM.fit(X_train, y_train)

# Create a 2nd classifier: a Random forest classifier
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=10, random_state=42)
rfc.fit(X_train, y_train)

#Predictions
predicted_SVM = SVM.predict(X_test)
y_pred_proba_SVM =SVM.predict_proba(X_test)[:,-1]
predicted_rfc = rfc.predict(X_test)
y_pred_proba_rfc =rfc.predict_proba(X_test)[:,-1]

#------- MODEL PERFORMANCE-------

# Roc and Auc
false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_pred_proba_SVM)
roc_curve_plot = metrics.plot_roc_curve(SVM, X_test, y_test, alpha = 0.8)
ax = plt.gca()
rfc_disp = metrics.plot_roc_curve(rfc, X_test, y_test, ax=ax, alpha=0.8)
plt.title('ROC curve')
plt.ylabel('True positive rate')
plt.xlabel('False positive rate (positive label: 1)')


plt.savefig('ROC curve')
plt.show()


# Ploting metrics extracted from Confusion matrix and CM itserlffor SVM
print(
    f"Classification report for classifier {SVM}:\n"
    f"{metrics.classification_report(y_test, predicted_SVM)}\n"
)

# true digit values and the predicted digit values.
disp = ConfusionMatrixDisplay.from_predictions(y_test, predicted_SVM)
disp.figure_.suptitle("Confusion Matrix for SVM")
print(f"Confusion matrix for SVM:\n{disp.confusion_matrix}")

plt.savefig('ConfusionMatrix for SVM')
plt.show()


# Ploting metrics extracted from Confusion matrix and CM itserlffor rfc
print(
    f"Classification report for classifier {rfc}:\n"
    f"{metrics.classification_report(y_test, predicted_rfc)}\n"
)

# true digit values and the predicted digit values.
disp = ConfusionMatrixDisplay.from_predictions(y_test, predicted_rfc)
disp.figure_.suptitle("Confusion Matrix for rfc")
print(f"Confusion matrix for rfc:\n{disp.confusion_matrix}")

plt.savefig('ConfusionMatrix for rfc')
plt.show()


 # %%
 #---------Feature importance--------
feature_names = ['ECG_HR', 'NIBP_MEAN','PROPO_CE', 'REMI_CE']
importances = rfc.feature_importances_
plt.pie(importances, labels = feature_names)

 # %%
 #Saving our model
with open("SVM_model.pickle", "wb") as file:
    pickle.dump(SVM, file)

with open("RandomForest_model.pickle", "wb") as file:
    pickle.dump(rfc, file)
```

```
print('Models Saved')

, probability= True) #, probability = True
SVM.fit(X_train, y_train)

# Create 2nd classifier: a Random forest classifier
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=10, random_state=42)
rfc.fit(X_train, y_train)

#Predictions
predicted_SVM = SVM.predict(X_test)
y_pred_proba_SVM =SVM.predict_proba(X_test)[:,-1]
predicted_rfc = rfc.predict(X_test)
y_pred_proba_rfc =rfc.predict_proba(X_test)[:,-1]

#------- MODEL PERFORMANCE-------

# Roc and Auc
false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_pred_proba_SVM)
roc_curve_plot = metrics.plot_roc_curve(SVM, X_test, y_test, alpha = 0.8)
ax = plt.gca()
rfc_disp = metrics.plot_roc_curve(rfc, X_test, y_test, ax=ax, alpha=0.8)
plt.title('ROC curve')
plt.ylabel('True positive rate')
plt.xlabel('False positive rate (positive label: 1)')

plt.savefig('ROC curve')
plt.show()


# Ploting metrics extracted from Confusion matrix and CM itserlffor SVM
print(
    f"Classification report for classifier {SVM}:\n"
    f"{metrics.classification_report(y_test, predicted_SVM)}\n"
)

# true digit values and the predicted digit values.
disp = ConfusionMatrixDisplay.from_predictions(y_test, predicted_SVM)
disp.figure_.suptitle("Confusion Matrix for SVM")
print(f"Confusion matrix for SVM:\n{disp.confusion_matrix}")

plt.savefig('ConfusionMatrix for SVM')
plt.show()


# Ploting metrics extracted from Confusion matrix and CM itserlffor rfc
print(
    f"Classification report for classifier {rfc}:\n"
    f"{metrics.classification_report(y_test, predicted_rfc)}\n"
)
```

```
# true digit values and the predicted digit values.
disp = ConfusionMatrixDisplay.from_predictions(y_test, predicted_rfc)
disp.figure_.suptitle("Confusion Matrix for rfc")
print(f"Confusion matrix for rfc:\n{disp.confusion_matrix}")

plt.savefig('ConfusionMatrix for rfc')
plt.show()


 # %%
 #---------Feature importance--------
feature_names = ['ECG_HR', 'NIBP_MEAN','PROPO_CE', 'REMI_CE']
importances = rfc.feature_importances_
plt.pie(importances, labels = feature_names)

 # %%
 #Saving our model
with open("SVM_model.pickle", "wb") as file:
    pickle.dump(SVM, file)

with open("RandomForest_model.pickle", "wb") as file:
    pickle.dump(rfc, file)

print('Models Saved')
```

Finally, Example.py implements a plot of an example case at threshold 0.85

```
Example.py
from libraries import *
from Train_Test_adquisition  import  data_preprocessing, finder

#Loading models
chdir("/Users/marcpalomercadenas/Desktop/TFG/TFG")
with open('SVM_model.pickle', "rb") as file:
    SVM = pickle.load(file)

with open('RandomForest_model.pickle', "rb") as file:
    rfc = pickle.load(file)


path = '/Users/marcpalomercadenas/Desktop/TFG/TFG/archives/CMA4_201217_083154.csv'
source = pd.read_csv(path)

#Input feature selection
selected_features = ['ECG_HR', 'NIBP_MEAN','PROPO_CE', 'REMI_CE']

#Processing the data
index, LoC_t  = finder(source, "verb'"", "verbal")
```

```
data_processed, LoC, Time = data_preprocessing(source, index, LoC_t,selected_features)

data_processed.columns = ['0','1','2','3']

#Normalisation
from sklearn.preprocessing import MaxAbsScaler
scaler = MaxAbsScaler()
data = scaler.fit_transform(data_processed)

#Model outcome
predicted_values = rfc.predict_proba(data)
pred_LoC = predicted_values[:,1]

#Plotting of the probability
plt.figure(dpi=1200)
plt.plot(Time, pred_LoC)
plt.plot(Time, LoC)
plt.legend(['SVM Prediction','Loss of verbal response'])
plt.title('LoC Prediction Example')
plt.xlabel('time since init Propofol infusion (s)')
plt.ylabel('Probability of being unconscious')
plt.savefig('LoC Example')
plt.show()



#Plotting with a threshold
plt.figure(dpi=1200)
plt.plot(Time, pred_LoC > 0.85)
plt.plot(Time, LoC)
plt.legend(['SVM Prediction','Loss of verbal response'])
plt.title('LoC Prediction Example')
plt.xlabel('time since init Propofol infusion (s)')
plt.ylabel('Probability of being unconscious. Threshold = 0.85')
plt.savefig('LoC Example with Threshold')
plt.show()
```